chanhvinh.com

User's Manual

32

RZ/A1L Group, RZ/A1LU Group, RZ/A1LC Group

User's Manual: Hardware

Renesas Microprocessor
RZ Family / RZ/A Series

arm

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (http://www.renesas.com).

Renesas Electronics
www.renesas.com

Rev.6.00 Jan 2021

chanhvinh.com

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

© 2020 Renesas Electronics Corporation. All rights reserved.

chanhvinh.com

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on
   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

chanhvinh.com

Contents

1. Overview 1-1
1.1 Features of This LSI 1-1
1.2 Product Lineup 1-9
1.3 Block Diagram 1-9
1.4 Pin Assignment 1-10
1.5 Pin Functions 1-14
1.6 List of Pins 1-20
2. CPU 2-1
2.1 Features 2-1
2.2 Configuration Signals 2-2
3. Boot Mode 3-1
3.1 Features 3-1
3.2 Boot Mode and Pin Function Setting 3-1
3.3 Hardware Used in Each Boot Mode 3-2
3.4 Exception Vector Address at a Reset in Each Boot Mode 3-3
3.5 Operation 3-4
3.5.1 Boot Mode 0 3-4
3.5.2 Boot Mode 1 3-4
3.5.3 Boot Mode 2 3-5
3.5.4 Boot Mode 3 3-7
3.6 Notes 3-9
3.6.1 Boot Related Pins 3-9
3.6.2 Operation when an Exception Occurs with the Exception Vector Set to the High Vector Address 3-9
3.6.3 Notes on Serial Flash Booting (Boot Mode 1) after This LSI is Reset 3-9
4. Secondary Cache 4-1
4.1 Features 4-1
4.2 Configuration Signals 4-1
5. LSI Internal Bus 5-1
5.1 LSI Internal Bus 5-1
5.1.1 Configuration 5-1
5.1.2 Operation 5-1
5.2 North Main Bus 5-2
5.2.1 Configuration 5-2
5.2.2 Features 5-2
5.2.3 Peripheral Buses 5-3
5.3 South Main Bus 5-5
5.3.1 Configuration 5-5
5.3.2 Features 5-5
5.3.3 Connected Buses 5-6
5.4 Address Map 5-7

chanhvinh.com

5.5 Address Remapping 5-10
5.5.1 Overview 5-10
5.5.2 Operation 5-10
5.6 AXI Interconnect 5-11
5.6.1 Configuration 5-11
5.6.2 Operation 5-11
5.7 Bus Bridges 5-11
5.8 AXI Protocol Control Signals 5-12
5.8.1 Bus Masters other than Cortex-A9, CoreSight, and the Direct Memory Access Controller 5-12
5.8.2 Cortex-A9 5-12
5.8.3 CoreSight 5-12
5.8.4 Direct Memory Access Controller 5-13
5.8.5 Slave Area 5-13
5.9 Write Buffers 5-13
5.10 Register Descriptions 5-14
5.10.1 Remap Register (RMPR) 5-15
5.10.2 AXI Bus Control Register 0 (AXIBUSCTL0) 5-16
5.10.3 AXI Bus Control Register 2 (AXIBUSCTL2) 5-17
5.10.4 AXI Bus Control Register 5 (AXIBUSCTL5) 5-17
5.10.5 AXI Bus Control Register 6 (AXIBUSCTL6) 5-18
5.10.6 AXI Bus Control Register 7 (AXIBUSCTL7) 5-19
5.10.7 AXI Bus Response Error Interrupt Control Register 0 (AXIRERRCTL0) 5-20
5.10.8 AXI Bus Response Error Interrupt Control Register 2 (AXIRERRCTL2) 5-21
5.10.9 AXI Bus Response Error Status Register 0 (AXIRERRST0) 5-22
5.10.10 AXI Bus Response Error Status Register 2 (AXIRERRST2) 5-23
5.10.11 AXI Bus Response Error Clear Register 0 (AXIRERRCLR0) 5-24
5.10.12 AXI Bus Response Error Clear Register 2 (AXIRERRCLR2) 5-25
5.11 Interrupt Request 5-26
6. Clock Pulse Generator 6-1
6.1 Features 6-1
6.2 Input/Output Pins 6-4
6.3 Clock Mode 6-5
6.4 Register Descriptions 6-7
6.4.1 Frequency Control Register (FRQCR) 6-7
6.5 Changing the Frequency 6-9
6.5.1 Changing the Division Ratio 6-9
6.6 Usage of the Clock Pins 6-10
6.6.1 In the Case of Inputting an External Clock 6-10
6.6.2 In the Case of Using a Crystal Resonator 6-11
6.6.3 In the Case of Not Using the Clock Pin 6-11
6.7 Oscillation Stabilizing Time 6-12

chanhvinh.com

6.7.1 Oscillation Stabilizing Time of the On-chip Crystal Oscillator 6-12
6.7.2 Oscillation Stabilizing Time of the PLL circuit 6-12
6.8 Notes on Board Design 6-13
6.8.1 Note on Using a PLL Oscillation Circuit 6-13
6.9 Definition of Modulation Rate and Frequency in the SSCG Specification 6-14
6.10 Clock Signals 6-15
6.10.1 Clock Signals for the System and Realtime Clock 6-15
6.10.2 Audio and USB Clock Signals 6-16
6.10.3 Video Image Clock Signals 6-16
6.10.4 Other Clock Signals 6-17
6.10.5 Internal Clock Signals (1) 6-18
6.10.6 Internal Clock Signals (2) 6-19
6.11 Usage Note 6-20
6.11.1 Notes on the SSCG 6-20
7. Interrupt Controller 7-1
7.1 Features 7-1
7.2 Input/Output Pins 7-2
7.3 Register Descriptions 7-3
7.3.1 Interrupt Control Register 0 (ICR0) 7-13
7.3.2 Interrupt Control Register 1 (ICR1) 7-14
7.3.3 IRQ Interrupt Request Register (IRQRR) 7-15
7.4 Interrupt Sources 7-16
7.4.1 NMI Interrupt 7-16
7.4.2 IRQ Interrupts 7-16
7.4.3 On-Chip Peripheral Module Interrupts 7-17
7.4.4 Pin Interrupts 7-18
7.5 Interrupt IDs 7-19
7.6 Operation 7-36
7.6.1 Initial Settings 7-36
7.6.2 Flow of Interrupt Operations 7-38
7.7 Data Transfer with Interrupt Request Signals 7-39
7.7.1 Handling Interrupt Request Signals as Sources for CPU Interrupt but Not Direct Memory Access Controller Activating 7-39
7.7.2 Handling Interrupt Request Signals as Sources for Activating Direct Memory Access Controller but Not CPU Interrupt 7-39
7.8 Usage Note 7-40
7.8.1 Timing to Clear Interrupt Source 7-40
7.8.2 Notes on Selecting IRQ Interrupt Pin Functions 7-40
7.8.3 Notes on Reading Interrupt ID Values from Interrupt Acknowledge Register (ICCIAR) 7-40
7.8.4 Notes on Using IRQ Pins as Triggers for Release from Standby when Software Standby is in Use 7-41

chanhvinh.com

8. Bus State Controller 8-1
8.1 Features 8-1
8.2 Input/Output Pins 8-3
8.3 Area Overview 8-4
8.3.1 Address Map 8-4
8.3.2 Data Bus Width and Related Pin Setting for Each Area Depending on Boot Mode 8-5
8.4 Register Descriptions 8-6
8.4.1 Common Control Register (CMNCR) 8-7
8.4.2 CSn Space Bus Control Register (CSnBCR) (n = 0 to 5) 8-8
8.4.3 CSn Space Wait Control Register (CSnWCR) (n = 0 to 5) 8-10
8.4.4 SDRAM Control Register (SDCR) 8-27
8.4.5 Refresh Timer Control/Status Register (RTCSR) 8-29
8.4.6 Refresh Timer Counter (RTCNT) 8-30
8.4.7 Refresh Time Constant Register (RTCOR) 8-30
8.4.8 Timeout Cycle Constant Register (TOSCORn) (n = 0 to 5) 8-31
8.4.9 Timeout Status Register (TOSTR) 8-32
8.4.10 Timeout Enable Register (TOENR) 8-34
8.5 Operation 8-35
8.5.1 Access Size and Data Alignment 8-35
8.5.2 Normal Space Interface 8-37
8.5.3 Access Wait Control 8-42
8.5.4 $\overline{\text{CSn}}$ Assert Period Expansion 8-44
8.5.5 MPX-I/O Interface 8-45
8.5.6 SDRAM Interface 8-48
8.5.7 Burst ROM (Clocked Asynchronous) Interface 8-77
8.5.8 SRAM Interface with Byte Selection 8-78
8.5.9 Burst ROM (Clocked Synchronous) Interface 8-83
8.5.10 Wait between Access Cycles 8-84
8.5.11 Others 8-87
9. Direct Memory Access Controller 9-1
9.1 Features 9-1
9.2 Input/Output Pins 9-2
9.3 Register Configuration 9-2
9.4 Register Descriptions 9-4
9.4.1 Next Source Address Register n (N0SA_n, N1SA_n) 9-12
9.4.2 Next Destination Address Register n (N0DA_n, N1DA_n) 9-12
9.4.3 Next Transaction Byte Register n (N0TB_n, N1TB_n) 9-13
9.4.4 Current Source Address Register (CRSA_n) 9-13
9.4.5 Current Destination Address Register (CRDA_n) 9-14
9.4.6 Current Transaction Byte Register (CRTB_n) 9-14
9.4.7 Channel Status Register n (CHSTAT_n) 9-15

chanhvinh.com

9.4.8 Channel Control Register n (CHCTRL_n) 9-18
9.4.9 Channel Configuration Register n (CHCFG_n) 9-20
9.4.10 Channel Interval Register n (CHITVL_n) 9-22
9.4.11 Channel Extension Register n (CHEXT_n) 9-23
9.4.12 Next Link Address Register n (NXLA_n) 9-24
9.4.13 Current Link Address Register n (CRLA_n) 9-24
9.4.14 DMA Control Register (DCTRL_0_7, DCTRL_8_15) 9-25
9.4.15 DMA Status EN Register (DSTAT_EN_0_7) 9-26
9.4.16 DMA Status EN Register (DSTAT_EN_8_15) 9-26
9.4.17 DMA Status ER Register (DSTAT_ER_0_7) 9-27
9.4.18 DMA Status ER Register (DSTAT_ER_8_15) 9-27
9.4.19 DMA Status END Register (DSTAT_END_0_7) 9-28
9.4.20 DMA Status END Register (DSTAT_END_8_15) 9-28
9.4.21 DMA Status TC Register (DSTAT_TC_0_7) 9-29
9.4.22 DMA Status TC Register (DSTAT_TC_8_15) 9-29
9.4.23 DMA Status SUS Register (DSTAT_SUS_0_7) 9-30
9.4.24 DMA Status SUS Register (DSTAT_SUS_8_15) 9-30
9.4.25 DMA Extension Resource Selectors 0 to 7 (DMARS0 to DMARS7) 9-31
9.5 Operation 9-34
9.5.1 Transfer Flow 9-34
9.5.2 DMA Transfer Requests 9-34
9.6 DMA Mode 9-39
9.6.1 Mode Setting 9-39
9.6.2 Register Mode 9-39
9.6.3 Link Mode 9-45
9.7 DMA Transfer 9-52
9.7.1 Transfer Mode 9-52
9.7.2 Priority Control for DMA Channels 9-53
9.7.3 Number of States of an External Bus Cycle 9-55
9.7.4 DMA Transfer Request 9-55
9.7.5 DMA Acknowledge Output Function 9-57
9.7.6 DMA Transfer End Output Function 9-59
9.7.7 DMA Transfer End Interrupt 9-59
9.7.8 DMA Error Interrupt 9-60
9.7.9 Interval Count Function 9-60
9.7.10 Difference in Operation Due to the Transfer Size 9-61
9.7.11 Transfer Status 9-62
9.8 DMA Setting Examples 9-66
9.8.1 Setting Example 1 (Register Mode/Hardware Request) 9-66
9.8.2 Setting Example 2 (Register Mode/Software Request) 9-68
9.8.3 Setting Example 3 (Register Mode/Continuous Execution) 9-70

chanhvinh.com

9.8.4 Setting Example 4 (Link Mode) 9-71
9.8.5 Next Register Set Continuous Execution Setting 9-74
9.9 Note 9-76
9.9.1 Divided Output of DACK0 and TEND0 9-76
9.9.2 TEND0 Not Output 9-77
9.9.3 Atomic Access (ARLOCK[1:0] and AWLOCK[1:0]) 9-77
10. Multi-Function Timer Pulse Unit 2 10-1
10.1 Features 10-1
10.2 Input/Output Pins 10-5
10.3 Register Descriptions 10-6
10.3.1 Timer Control Register (TCR) 10-8
10.3.2 Timer Mode Register (TMDR) 10-11
10.3.3 Timer I/O Control Register (TIOR) 10-13
10.3.4 Timer Interrupt Enable Register (TIER) 10-31
10.3.5 Timer Status Register (TSR) 10-33
10.3.6 Timer Buffer Operation Transfer Mode Register (TBTM) 10-36
10.3.7 Timer Input Capture Control Register (TICCR) 10-37
10.3.8 Timer A/D Converter Start Request Control Register (TADCR) 10-38
10.3.9 Timer A/D Converter Start Request Cycle Set Registers (TADCORA_4 and TADCORB_4) 10-40
10.3.10 Timer A/D Converter Start Request Cycle Set Buffer Registers (TADCOBRA_4 and TADCOBRB_4) 10-40
10.3.11 Timer Counter (TCNT) 10-40
10.3.12 Timer General Register (TGR) 10-41
10.3.13 Timer Start Register (TSTR) 10-42
10.3.14 Timer Synchronous Register (TSYR) 10-43
10.3.15 Timer Read/Write Enable Register (TRWER) 10-44
10.3.16 Timer Output Master Enable Register (TOER) 10-45
10.3.17 Timer Output Control Register 1 (TOCR1) 10-46
10.3.18 Timer Output Control Register 2 (TOCR2) 10-48
10.3.19 Timer Output Level Buffer Register (TOLBR) 10-51
10.3.20 Timer Gate Control Register (TGCR) 10-52
10.3.21 Timer Subcounter (TCNTS) 10-53
10.3.22 Timer Dead Time Data Register (TDDR) 10-53
10.3.23 Timer Cycle Data Register (TCDR) 10-54
10.3.24 Timer Cycle Buffer Register (TCBR) 10-54
10.3.25 Timer Interrupt Skipping Set Register (TITCR) 10-55
10.3.26 Timer Interrupt Skipping Counter (TITCNT) 10-57
10.3.27 Timer Buffer Transfer Set Register (TBTER) 10-58
10.3.28 Timer Dead Time Enable Register (TDER) 10-59
10.3.29 Timer Waveform Control Register (TWCR) 10-60

chanhvinh.com

10.3.30 Bus Master Interface 10-60
10.4 Operation 10-61
10.4.1 Basic Functions 10-61
10.4.2 Synchronous Operation 10-67
10.4.3 Buffer Operation 10-69
10.4.4 Cascaded Operation 10-73
10.4.5 PWM Modes 10-78
10.4.6 Phase Counting Mode 10-82
10.4.7 Reset-Synchronized PWM Mode 10-88
10.4.8 Complementary PWM Mode 10-91
10.4.9 A/D Converter Start Request Delaying Function 10-126
10.4.10 TCNT Capture at Crest and/or Trough in Complementary PWM Operation 10-129
10.5 Interrupt Sources 10-130
10.5.1 Interrupt Sources and Priorities 10-130
10.5.2 Activation of Direct Memory Access Controller 10-131
10.5.3 A/D Converter Activation 10-131
10.6 Operation Timing 10-133
10.6.1 Input/Output Timing 10-133
10.6.2 Interrupt Signal Timing 10-138
10.7 Usage Notes 10-141
10.7.1 Module Standby Mode Setting 10-141
10.7.2 Input Clock Restrictions 10-141
10.7.3 Caution on Period Setting 10-142
10.7.4 Contention between TCNT Write and Clear Operations 10-142
10.7.5 Contention between TCNT Write and Increment Operations 10-143
10.7.6 Contention between TGR Write and Compare Match 10-143
10.7.7 Contention between Buffer Register Write and Compare Match 10-144
10.7.8 Contention between Buffer Register Write and TCNT Clear 10-145
10.7.9 Contention between TGR Read and Input Capture 10-145
10.7.10 Contention between TGR Write and Input Capture 10-146
10.7.11 Contention between Buffer Register Write and Input Capture 10-146
10.7.12 TCNT_2 Write and Overflow/Underflow Contention in Cascade Connection 10-147
10.7.13 Counter Value during Complementary PWM Mode Stop 10-148
10.7.14 Buffer Operation Setting in Complementary PWM Mode 10-148
10.7.15 Reset Sync PWM Mode Buffer Operation and Compare Match Flag 10-149
10.7.16 Overflow Flags in Reset Synchronous PWM Mode 10-150
10.7.17 Contention between Overflow/Underflow and Counter Clearing 10-150
10.7.18 Contention between TCNT Write and Overflow/Underflow 10-151
10.7.19 Cautions on Transition from Normal Operation or PWM Mode 1 to Reset-Synchronized PWM Mode 10-151
10.7.20 Output Level in Complementary PWM Mode and Reset-Synchronized PWM Mode 10-151

chanhvinh.com

10.7.21 Interrupts in Module Standby Mode 10-151
10.7.22 Simultaneous Capture of TCNT_1 and TCNT_2 in Cascade Connection 10-152
10.7.23 Notes on Output Waveform Control During Synchronous Counter Clearing in Complementary PWM Mode 10-152
10.8 Output Pin Initialization for Multi-Function Timer Pulse Unit 2 10-154
10.8.1 Operating Modes 10-154
10.8.2 Reset Start Operation 10-154
10.8.3 Operation in Case of Re-Setting Due to Error during Operation, etc. 10-155
10.8.4 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation, etc. 10-156
11. OS Timer 11-1
11.1 Functional Overview 11-1
11.1.1 Features of OSTM 11-1
11.2 Registers 11-2
11.2.1 Registers Overview 11-2
11.2.2 Details of OSTM Registers 11-3
11.3 Functional Description 11-7
11.3.1 Block Diagram 11-7
11.3.2 Count Clock 11-8
11.3.3 Generation of Interrupt Request 11-8
11.3.4 Starting and Stopping the Timer 11-9
11.3.5 Interval Timer Mode 11-9
11.3.6 Free-Running Comparison Mode 11-13
12. Watchdog Timer 12-1
12.1 Features 12-1
12.2 Input/Output Pin 12-2
12.3 Register Descriptions 12-2
12.3.1 Watchdog Timer Counter (WTCNT) 12-3
12.3.2 Watchdog Timer Control/Status Register (WTCSR) 12-3
12.3.3 Watchdog Reset Control/Status Register (WRCSR) 12-5
12.3.4 Notes on Register Access 12-6
12.4 Usage 12-8
12.4.1 Canceling Software Standby Mode 12-8
12.4.2 Using Watchdog Timer Mode 12-8
12.4.3 Using Interval Timer Mode 12-9
12.5 Usage Notes 12-10
12.5.1 Timer Variation 12-10
12.5.2 Prohibition against Setting H'FF to WTCNT 12-10
12.5.3 Interval Timer Overflow Flag 12-10
12.5.4 System Reset by $\overline{\text{WDTOVF}}$ Signal 12-10
12.5.5 Internal Reset in Watchdog Timer Mode 12-10

chanhvinh.com

13. Realtime Clock 13-1
13.1 Features 13-1
13.2 Input/Output Pin 13-3
13.3 Register Descriptions 13-3
13.3.1 64-Hz Counter (R64CNT) 13-4
13.3.2 Second Counter (RSECCNT) 13-5
13.3.3 Minute Counter (RMINCNT) 13-5
13.3.4 Hour Counter (RHRCNT) 13-6
13.3.5 Day of Week Counter (RWKCNT) 13-6
13.3.6 Day Counter (RDAYCNT) 13-7
13.3.7 Month Counter (RMONCNT) 13-7
13.3.8 Year Counter (RYRCNT) 13-8
13.3.9 Second Alarm Register (RSECAR) 13-8
13.3.10 Minute Alarm Register (RMINAR) 13-9
13.3.11 Hour Alarm Register (RHRAR) 13-9
13.3.12 Day of Week Alarm Register (RWKAR) 13-10
13.3.13 Day Alarm Register (RDAYAR) 13-10
13.3.14 Month Alarm Register (RMONAR) 13-11
13.3.15 Year Alarm Register (RYRAR) 13-11
13.3.16 Control Register 1 (RCR1) 13-12
13.3.17 Control Register 2 (RCR2) 13-13
13.3.18 Control Register 3 (RCR3) 13-14
13.3.19 Control Register 5 (RCR5) 13-14
13.3.20 Frequency Register H/L (RFRH/L) 13-15
13.4 Operation 13-16
13.4.1 Initial Settings of Registers after Power-On and Oscillation Stabilization Time 13-16
13.4.2 Setting Time 13-16
13.4.3 Reading Time 13-17
13.4.4 Alarm Function 13-18
13.5 Usage Notes 13-19
13.5.1 Register Writing during Count Operation 13-19
13.5.2 Use of Realtime Clock Periodic Interrupts 13-19
13.5.3 Transition to Standby Mode after Setting Register 13-19
13.5.4 Usage Notes when Writing to and Reading the Register 13-19
14. Serial Communication Interface with FIFO 14-1
14.1 Features 14-1
14.2 Input/Output Pins 14-3
14.3 Register Descriptions 14-4
14.3.1 Receive Shift Register (SCRSR) 14-5
14.3.2 Receive FIFO Data Register (SCFRDR) 14-5
14.3.3 Transmit Shift Register (SCTSR) 14-6

chanhvinh.com

14.3.4 Transmit FIFO Data Register (SCFTDR) 14-6
14.3.5 Serial Mode Register (SCSMR) 14-7
14.3.6 Serial Control Register (SCSCR) 14-9
14.3.7 Serial Status Register (SCFSR) 14-11
14.3.8 Bit Rate Register (SCBRR) 14-15
14.3.9 FIFO Control Register (SCFCR) 14-19
14.3.10 FIFO Data Count Set Register (SCFDR) 14-21
14.3.11 Serial Port Register (SCSPTR) 14-22
14.3.12 Line Status Register (SCLSR) 14-24
14.3.13 Serial Extension Mode Register (SCEMR) 14-25
14.4 Operation 14-26
14.4.1 Overview 14-26
14.4.2 Operation in Asynchronous Mode 14-28
14.4.3 Operation in Clock Synchronous Mode 14-39
14.5 Interrupts 14-47
14.6 Usage Notes 14-48
14.6.1 SCFTDR Writing and TDFE Flag 14-48
14.6.2 SCFRDR Reading and RDF Flag 14-48
14.6.3 Restriction on Direct Memory Controller Usage 14-48
14.6.4 Break Detection and Processing 14-48
14.6.5 Sending a Break Signal 14-49
14.6.6 Receive Data Sampling Timing and Receive Margin (Asynchronous Mode) 14-49
14.6.7 Selection of Base Clock in Asynchronous Mode 14-50
15. Serial Communications Interface 15-1
15.1 Overview 15-1
15.2 Register Descriptions 15-3
15.2.1 Receive Shift Register (RSR) 15-4
15.2.2 Receive Data Register (RDR) 15-4
15.2.3 Transmit Data Register (TDR) 15-4
15.2.4 Transmit Shift Register (TSR) 15-4
15.2.5 Serial Mode Register (SMR) 15-5
15.2.6 Serial Control Register (SCR) 15-8
15.2.7 Serial Status Register (SSR) 15-12
15.2.8 Smart Card Mode Register (SCMR) 15-16
15.2.9 Bit Rate Register (BRR) 15-17
15.2.10 Serial Extended Mode Register (SEMR) 15-21
15.2.11 Noise Filter Setting Register (SNFR) 15-21
15.2.12 Extended Function Control Register (SECR) 15-22
15.3 Operation in Asynchronous Mode 15-23
15.3.1 Serial Data Transfer Format 15-23
15.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode 15-25

chanhvinh.com

15.3.3 Clock 15-26
15.3.4 CTS and RTS Functions 15-26
15.3.5 SCI Initialization (Asynchronous Mode) 15-27
15.3.6 Serial Data Transmission (Asynchronous Mode) 15-28
15.3.7 Serial Data Reception (Asynchronous Mode) 15-30
15.4 Multi-Processor Communications Function 15-34
15.4.1 Multi-Processor Serial Data Transmission 15-35
15.4.2 Multi-Processor Serial Data Reception 15-36
15.5 Operation in Clock Synchronous Mode 15-39
15.5.1 Clock 15-39
15.5.2 CTS and RTS Functions 15-39
15.5.3 Initialization (Clock Synchronous Mode) 15-40
15.5.4 Serial Data Transmission (Clock Synchronous Mode) 15-41
15.5.5 Serial Data Reception (Clock Synchronous Mode) 15-43
15.5.6 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode) 15-46
15.6 Operation in Smart Card Interface Mode 15-47
15.6.1 Sample Connection 15-47
15.6.2 Data Format (Except in Block Transfer Mode) 15-48
15.6.3 Block Transfer Mode 15-50
15.6.4 Receive Data Sampling Timing and Reception Margin 15-50
15.6.5 Initialization (Smart Card Interface Mode) 15-51
15.6.6 Serial Data Transmission (Except in Block Transfer Mode) 15-52
15.6.7 Serial Data Reception (Except in Block Transfer Mode) 15-55
15.6.8 Clock Output Control 15-56
15.7 Noise Cancellation Function 15-58
15.8 Interrupt Sources 15-59
15.8.1 Interrupts in Serial Communications Interface Mode 15-59
15.8.2 Interrupts in Smart Card Interface Mode 15-60
15.9 Usage Notes 15-61
15.9.1 Setting the Module Standby Function 15-61
15.9.2 Break Detection and Processing 15-61
15.9.3 The Mark State and Production of Breaks 15-61
15.9.4 Receive Error Flags and Transmit Operations (Clock Synchronous Mode Only) 15-61
15.9.5 Writing Data to TDR 15-61
15.9.6 Restrictions on Clock Synchronous Transmission 15-61
15.9.7 Restrictions on Using DMAC 15-61
15.9.8 Points to Note on Starting Transfer 15-61
15.9.9 SCI Operations during Low Power Consumption State 15-62
15.9.10 External Clock Input in Clock Synchronous Mode 15-64
15.10 IrDA Communications 15-65
15.11 IrDA Register Description 15-66

chanhvinh.com

15.11.1 IrDA Control Register (IRCR) 15-66
15.12 IrDA Operation 15-67
15.12.1 Flow of IrDA Setting 15-67
15.12.2 Transmission 15-67
15.12.3 Reception 15-68
15.12.4 Selection of High-Level Pulse Width 15-68
15.13 Notes on Using the IrDA Module 15-69
15.13.1 Shortest Pulse Width in Reception 15-69
15.13.2 Asynchronous Basic Clock for Serial Communication Interface 15-69
16. Renesas Serial Peripheral Interface 16-1
16.1 Features 16-1
16.2 Input/Output Pins 16-3
16.3 Register Descriptions 16-4
16.3.1 Control Register (SPCR) 16-6
16.3.2 Slave Select Polarity Register (SSLP) 16-7
16.3.3 Pin Control Register (SPPCR) 16-7
16.3.4 Status Register (SPSR) 16-8
16.3.5 Data Register (SPDR) 16-10
16.3.6 Sequence Control Register (SPSCR) 16-10
16.3.7 Sequence Status Register (SPSSR) 16-11
16.3.8 Bit Rate Register (SPBR) 16-11
16.3.9 Data Control Register (SPDCR) 16-12
16.3.10 Clock Delay Register (SPCKD) 16-13
16.3.11 Slave Select Negation Delay Register (SSLND) 16-14
16.3.12 Next-Access Delay Register (SPND) 16-15
16.3.13 Command Register (SPCMD) 16-16
16.3.14 Buffer Control Register (SPBFCR) 16-18
16.3.15 Buffer Data Count Setting Register (SPBFDR) 16-19
16.4 Operation 16-20
16.4.1 Overview of Operations 16-20
16.4.2 Pin Control 16-21
16.4.3 System Configuration Example 16-21
16.4.4 Transfer Format 16-24
16.4.5 Data Format 16-26
16.4.6 Error Detection 16-32
16.4.7 Initialization 16-34
16.4.8 SPI Operation 16-35
16.4.9 Error Handling 16-45
16.4.10 Loopback Mode 16-46
16.4.11 Interrupt Sources 16-46

chanhvinh.com

17. SPI Multi I/O Bus Controller 17-1
17.1 Features 17-1
17.2 Block Diagram 17-2
17.3 Input/Output Pins 17-3
17.4 Register Descriptions 17-4
17.4.1 Common Control Register (CMNCR) 17-5
17.4.2 SSL Delay Register (SSLDR) 17-8
17.4.3 Bit Rate Register (SPBCR) 17-9
17.4.4 Data Read Control Register (DRCR) 17-11
17.4.5 Data Read Command Setting Register (DRCMR) 17-12
17.4.6 Data Read Extended Address Setting Register (DREAR) 17-13
17.4.7 Data Read Option Setting Register (DROPR) 17-14
17.4.8 Data Read Enable Setting Register (DRENR) 17-15
17.4.9 SPI Mode Control Register (SMCR) 17-17
17.4.10 SPI Mode Command Setting Register (SMCMR) 17-18
17.4.11 SPI Mode Address Setting Register (SMADR) 17-18
17.4.12 SPI Mode Option Setting Register (SMOPR) 17-19
17.4.13 SPI Mode Enable Setting Register (SMENR) 17-20
17.4.14 SPI Mode Read Data Register 0 (SMRDR0) 17-22
17.4.15 SPI Mode Read Data Register 1 (SMRDR1) 17-22
17.4.16 SPI Mode Write Data Register 0 (SMWDR0) 17-23
17.4.17 SPI Mode Write Data Register 1 (SMWDR1) 17-23
17.4.18 Common Status Register (CMNSR) 17-24
17.4.19 SPI AC Input Characteristics Adjustment Register (CKDLY) (RZ/A1LU only) 17-25
17.4.20 Data Read Dummy Cycle Setting Register (DRDMCR) 17-26
17.4.21 Data Read DDR Enable Register (DRDRENR) (RZ/A1LU only) 17-27
17.4.22 SPI Mode Dummy Cycle Setting Register (SMDMCR) 17-28
17.4.23 SPI Mode DDR Enable Register (SMDRENR) (RZ/A1LU only) 17-29
17.4.24 SPI AC Output Characteristics Adjustment Register (SPODLY) (RZ/A1LU only) 17-30
17.5 Operation 17-31
17.5.1 System Configuration 17-31
17.5.2 Address Map 17-32
17.5.3 32-bit Serial Flash Addresses 17-32
17.5.4 Data Alignment 17-33
17.5.5 Operating Modes 17-34
17.5.6 External Address Space Read Mode 17-34
17.5.7 Read Cache 17-39
17.5.8 SPI Operating Mode 17-40
17.5.9 Transfer Format 17-43
17.5.10 Data Format 17-45
17.5.11 Data Pin Control 17-50

chanhvinh.com

17.5.12 SPBSSL Pin Control 17-52
17.5.13 Flags 17-52
17.6 Usage Notes 17-53
17.6.1 Notes on Transfer to Read Data in SPI Operating Mode 17-53
17.6.2 Notes on Starting Transfer from the SPBSSL Retained State in SPI Operating Mode 17-53
18. I²C Bus Interface 18-1
18.1 Features 18-1
18.1.1 Channels 18-1
18.1.2 Register Base Addresses 18-1
18.1.3 External I/O Signals 18-2
18.2 Overview 18-3
18.2.1 Functional Overview 18-3
18.2.2 Block Diagram 18-5
18.3 Registers 18-7
18.3.1 RIICnCR1 — I²C Bus Control Register 1 18-7
18.3.2 RIICnCR2 — I²C Bus Control Register 2 18-10
18.3.3 RIICnMR1 — I²C Bus Mode Register 1 18-14
18.3.4 RIICnMR2 — I²C Bus Mode Register 2 18-16
18.3.5 RIICnMR3 — I²C Bus Mode Register 3 18-18
18.3.6 RIICnFER — I²C Bus Function Enable Register 18-21
18.3.7 RIICnSER — I²C Bus Status Enable Register 18-23
18.3.8 RIICnIER — I²C Bus Interrupt Enable Register 18-25
18.3.9 RIICnSR1 — I²C Bus Status Register 1 18-27
18.3.10 RIICnSR2 — I²C Bus Status Register 2 18-30
18.3.11 RIICnSARy — I²C Slave Address Register y (y = 0 to 2) 18-35
18.3.12 RIICnBRL — I²C Bus Bit Rate Low-Level Register 18-37
18.3.13 RIICnBRH — I²C Bus Bit Rate High-Level Register 18-38
18.3.14 RIICnDRT — I²C Bus Transmit Data Register 18-41
18.3.15 RIICnDRR — I²C Bus Receive Data Register 18-42
18.3.16 RIICnDRS — I²C Bus Shift Register 18-43
18.4 Interrupt Sources 18-44
18.5 Operation 18-45
18.5.1 Communication Data Format 18-45
18.5.2 Initial Settings 18-46
18.5.3 Master Transmit Operation 18-47
18.5.4 Master Receive Operation 18-52
18.5.5 Slave Transmit Operation 18-58
18.5.6 Slave Receive Operation 18-61
18.6 SCL Synchronization Circuit 18-64
18.7 Facility for Delaying SDA Output 18-65
18.8 Digital Noise-Filter Circuits 18-66

chanhvinh.com

18.9 Address Match Detection 18-67
18.9.1 Slave-Address Match Detection 18-67
18.9.2 Detection of the General Call Address 18-69
18.9.3 Device-ID Address Detection 18-70
18.9.4 Host Address Detection 18-72
18.10 Automatically Low-Hold Function for SCL 18-73
18.10.1 Function to Prevent Wrong Transmission of Transmit Data 18-73
18.10.2 NACK Reception Transfer Suspension Function 18-74
18.10.3 Function to Prevent Failure to Receive Data 18-75
18.11 Arbitration-Lost Detection Functions 18-77
18.11.1 Master Arbitration-Lost Detection (MALE Bit) 18-77
18.11.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit) 18-79
18.11.3 Slave Arbitration-Lost Detection (SALE Bit) 18-80
18.12 Start Condition/Restart Condition/Stop Condition Issuing Function 18-81
18.12.1 Issuing a Start Condition 18-81
18.12.2 Issuing a Restart Condition 18-81
18.12.3 Issuing a Stop Condition 18-83
18.13 Bus Hanging 18-84
18.13.1 Timeout Function 18-84
18.13.2 Extra SCL Clock Cycle Output Function 18-86
18.13.3 RIIC Reset and Internal Reset 18-87
18.14 SMBus Operation 18-88
18.14.1 SMBus Timeout Measurement 18-88
18.14.2 SMBus Host Notification Protocol/Notify ARP Master 18-89
18.15 Reset Function of RIIC 18-90
19. Serial Sound Interface 19-1
19.1 Features 19-1
19.2 Input/Output Pins 19-3
19.3 Register Description 19-4
19.3.1 Control Register (SSICR) 19-6
19.3.2 Status Register (SSISR) 19-10
19.3.3 Transmit Data Register (SSITDR) 19-12
19.3.4 Receive Data Register (SSIRDR) 19-12
19.3.5 FIFO Control Register (SSIFCR) 19-13
19.3.6 FIFO Status Register (SSIFSR) 19-15
19.3.7 Transmit FIFO Data Register (SSIFTDR) 19-17
19.3.8 Receive FIFO Data Register (SSIFRDR) 19-17
19.3.9 TDM Mode Register (SSITDMR) 19-18
19.3.10 FC Control Register (SSIFCCR) 19-19
19.3.11 FC Mode Register (SSIFCMR) 19-20
19.3.12 FC Status Register (SSIFCSR) 19-21

chanhvinh.com

19.4 Operation Description 19-22
19.4.1 Bus Format 19-22
19.4.2 Non-Compressed Modes 19-23
19.4.3 TDM Mode 19-31
19.4.4 WS Continue Mode 19-32
19.4.5 Operation Modes 19-32
19.4.6 Transmit Operation 19-33
19.4.7 Receive Operation 19-36
19.4.8 Serial Bit Clock Control 19-38
19.5 Usage Notes 19-39
19.5.1 Limitations from Underflow or Overflow during DMA Operation 19-39
19.5.2 Note on Changing Mode from Master Transceiver to Master Receiver 19-39
19.5.3 Limits on TDM mode and WS Continue Mode 19-39
20. Media Local Bus 20-1
20.1 Features 20-1
20.2 Input/Output Pins 20-2
20.3 Register Description 20-2
21. CAN Interface 21-1
21.1 Overview 21-1
21.1.1 Units 21-1
21.1.2 Register addresses 21-2
21.1.3 Clock supply 21-2
21.1.4 Interrupts 21-3
21.1.5 I/O signals 21-3
21.2 Function 21-4
21.2.1 Block Diagram 21-6
21.3 Registers 21-7
21.3.1 RSCAN0CmCFG — Channel Configuration Register (m = 0 or 1) 21-20
21.3.2 RSCAN0CmCTR — Channel Control Register (m = 0 or 1) 21-22
21.3.3 RSCAN0CmSTS — Channel Status Register (m = 0 or 1) 21-26
21.3.4 RSCAN0CmERFL — Channel Error Flag Register (m = 0 or 1) 21-28
21.3.5 RSCAN0GCFG — Global Configuration Register 21-32
21.3.6 RSCAN0GCTR — Global Control Register 21-35
21.3.7 RSCAN0GSTS — Global Status Register 21-37
21.3.8 RSCAN0GERFL — Global Error Flag Register 21-39
21.3.9 RSCAN0GTINTSTS0 — Global TX Interrupt Status Register 0 21-40
21.3.10 RSCAN0GTSC — Global Timestamp Counter Register 21-42
21.3.11 RSCAN0GAFLECTR — Receive Rule Entry Control Register 21-43
21.3.12 RSCAN0GAFLCFG0 — Receive Rule Configuration Register 0 21-44
21.3.13 RSCAN0GAFLIDj — Receive Rule ID Register (j = 0 to 15) 21-45
21.3.14 RSCAN0GAFLMj — Receive Rule Mask Register (j = 0 to 15) 21-47

chanhvinh.com

21.3.15 RSCAN0GAFLP0j — Receive Rule Pointer 0 Register (j = 0 to 15) 21-48
21.3.16 RSCAN0GAFLP1j — Receive Rule Pointer 1 Register (j = 0 to 15) 21-50
21.3.17 RSCAN0RMNB — Receive Buffer Number Register 21-51
21.3.18 RSCAN0RMNDy — Receive Buffer New Data Register y (y = 0) 21-52
21.3.19 RSCAN0RMIDq — Receive Buffer ID Register (q = 0 to 31) 21-53
21.3.20 RSCAN0RMPTRq — Receive Buffer Pointer Register (q = 0 to 31) 21-54
21.3.21 RSCAN0RMDF0q — Receive Buffer Data Field 0 Register (q = 0 to 31) 21-55
21.3.22 RSCAN0RMDF1q — Receive Buffer Data Field 1 Register (q = 0 to 31) 21-56
21.3.23 RSCAN0RFCCx — Receive FIFO Buffer Configuration and Control Register (x = 0 to 7) 21-57
21.3.24 RSCAN0RFSTSx — Receive FIFO Buffer Status Register (x = 0 to 7) 21-59
21.3.25 RSCAN0RFPCTRx — Receive FIFO Buffer Pointer Control Register (x = 0 to 7) 21-61
21.3.26 RSCAN0RFIDx — Receive FIFO Buffer Access ID Register (x = 0 to 7) 21-62
21.3.27 RSCAN0RFPTRx — Receive FIFO Buffer Access Pointer Register (x = 0 to 7) 21-63
21.3.28 RSCAN0RFDF0x — Receive FIFO Buffer Access Data Field 0 Register (x = 0 to 7) 21-64
21.3.29 RSCAN0RFDF1x — Receive FIFO Buffer Access Data Field 1 Register (x = 0 to 7) 21-65
21.3.30 RSCAN0CFCCk — Transmit/receive FIFO buffer Configuration and Control Register k (k = 0 to 5) 21-66
21.3.31 RSCAN0CFSTSk — Transmit/receive FIFO buffer Status Register (k = 0 to 5) 21-70
21.3.32 RSCAN0CFPCTRk — Transmit/receive FIFO buffer Pointer Control Register (k = 0 to 5) 21-73
21.3.33 RSCAN0CFIDk — Transmit/receive FIFO buffer Access ID Register (k = 0 to 5) 21-75
21.3.34 RSCAN0CFPTRk — Transmit/receive FIFO buffer Access Pointer Register (k = 0 to 5) 21-77
21.3.35 RSCAN0CFDF0k — Transmit/receive FIFO buffer Access Data Field 0 Register (k = 0 to 5) 21-79
21.3.36 RSCAN0CFDF1k — Transmit/receive FIFO buffer Access Data Field 1 Register (k = 0 to 5) 21-80
21.3.37 RSCAN0FESTS — FIFO Empty Status Register 21-81
21.3.38 RSCAN0FFSTS — FIFO Full Status Register 21-82
21.3.39 RSCAN0FMSTS — FIFO Message Lost Status Register 21-83
21.3.40 RSCAN0RFISTS — Receive FIFO Buffer Interrupt Flag Status Register 21-84
21.3.41 RSCAN0CFRISTS — Transmit/receive FIFO buffer Receive Interrupt Flag Status Register 21-85
21.3.42 RSCAN0CFTISTS — Transmit/receive FIFO buffer Transmit Interrupt Flag Status Register 21-86
21.3.43 RSCAN0TMCp — Transmit Buffer Control Register (p = 0 to 31) 21-87
21.3.44 RSCAN0TMSTSp — Transmit Buffer Status Register (p = 0 to 31) 21-89
21.3.45 RSCAN0TMTRSTSy — Transmit Buffer Transmit Request Status Register y (y = 0) 21-91

chanhvinh.com

21.3.46 RSCAN0TMTARSTSy — Transmit Buffer Transmit Abort Request Status Register y (y = 0) 21-92
21.3.47 RSCAN0TMTCSTSy — Transmit Buffer Transmit Complete Status Register y (y = 0) 21-93
21.3.48 RSCAN0TMTASTSy — Transmit Buffer Transmit Abort Status Register y (y = 0) 21-94
21.3.49 RSCAN0TMIECy — Transmit Buffer Interrupt Enable Configuration Register y (y = 0) 21-95
21.3.50 RSCAN0TMIDp — Transmit Buffer ID Register (p = 0 to 31) 21-96
21.3.51 RSCAN0TMPTRp — Transmit Buffer Pointer Register (p= 0 to 31) 21-97
21.3.52 RSCAN0TMDF0p — Transmit Buffer Data Field 0 Register (p = 0 to 31) 21-98
21.3.53 RSCAN0TMDF1p — Transmit Buffer Data Field 1 Register (p = 0 to 31) 21-99
21.3.54 RSCAN0TXQCCm — Transmit Queue Configuration and Control Register (m = 0 or 1) 21-100
21.3.55 RSCAN0TXQSTSm — Transmit Queue Status Register (m = 0 or 1) 21-102
21.3.56 RSCAN0TXQPCTRm — Transmit Queue Pointer Control Register (m = 0 or 1) 21-104
21.3.57 RSCAN0THLCCm — Transmit History Configuration and Control Register (m = 0 or 1) 21-105
21.3.58 RSCAN0THLSTSm — Transmit History Status Register (m = 0 or 1) 21-107
21.3.59 RSCAN0THLACCm — Transmit History Access Register (m = 0 or 1) 21-109
21.3.60 RSCAN0THLPCTRm — Transmit History Pointer Control Register (m = 0 or 1) 21-110
21.3.61 RSCAN0GTSTCFG — Global Test Configuration Register 21-111
21.3.62 RSCAN0GTSTCTR — Global Test Control Register 21-112
21.3.63 RSCAN0GLOCKK — Global Lock Key Register 21-113
21.4 Interrupt Sources 21-114
21.5 RSCAN Modes 21-118
21.5.1 Global Modes 21-118
21.5.2 Channel Modes 21-121
21.6 Reception Function 21-126
21.6.1 Data Processing Using the Receive Rule Table 21-126
21.7 Transmission Functions 21-130
21.7.1 Transmit Priority Determination 21-131
21.7.2 Transmission Using Transmit Buffers 21-131
21.7.3 Transmission Using FIFO Buffers 21-132
21.7.4 Transmission Using Transmit Queues 21-135
21.7.5 Transmit History Function 21-135
21.8 Gateway Function 21-137
21.9 Test Function 21-138
21.9.1 Standard Test Mode 21-138
21.9.2 Listen-Only Mode 21-138
21.9.3 Self-Test Mode (Loopback Mode) 21-139
21.9.4 Inter-Channel Communication Test 21-141
21.10 RS-CAN Setting Procedure 21-142

chanhvinh.com

21.10.1 Initial Settings 21-142
21.10.2 Reception Procedure 21-148
21.10.3 Transmission Procedure 21-153
21.10.4 Test Settings 21-162
21.11 Notes on the RS-CAN Module 21-164
22. IEBus Controller 22-1
22.1 IEBB Features 22-1
22.2 Configuration 22-3
22.2.1 Function overview 22-3
22.2.2 Block diagram 22-4
22.3 Registers 22-5
22.3.1 IEBBn register overview 22-5
22.3.2 IEBBn control register details 22-6
22.4 Interrupt Operations 22-72
22.4.1 Interrupt request signals 22-72
22.4.2 Interrupt judgment examples 22-77
22.5 Operation 22-79
22.5.1 FIFO 22-79
22.5.2 Initial settings 22-81
22.5.3 Master transmission (single mode) 22-82
22.5.4 Master transmission (FIFO mode) 22-84
22.5.5 Master reception (single mode) 22-86
22.5.6 Master reception (FIFO mode) 22-88
22.5.7 Slave transmission (single mode) 22-90
22.5.8 Slave transmission (FIFO mode) 22-93
22.5.9 Slave reception (single mode) 22-96
22.5.10 Slave reception (FIFO mode) 22-98
22.6 Setup Procedures 22-99
22.6.1 Master transmission (single mode) 22-99
22.6.2 Master transmission (FIFO mode) 22-100
22.6.3 Master reception (single mode) 22-101
22.6.4 Master reception (FIFO mode) 22-102
22.6.5 Slave transmission (single mode) 22-103
22.6.6 Slave transmission (FIFO mode) 22-106
22.6.7 Slave reception (single mode) 22-109
22.6.8 Slave reception (FIFO mode) 22-110
22.7 Functions 22-111
22.7.1 IEBus communication protocol 22-111
22.7.2 Determination of bus mastership (arbitration) 22-112
22.7.3 Communication mode 22-112
22.7.4 Communication address 22-113

chanhvinh.com

22.7.5 Broadcast communication 22-113
22.7.6 IEBus transfer format 22-114
22.7.7 Transfer data 22-124
22.7.8 Bit format 22-128
23. Renesas SPDIF Interface 23-1
23.1 Overview 23-1
23.2 Features 23-1
23.3 Functional Block Diagram 23-2
23.4 Input/Output Pins 23-3
23.5 Renesas SPDIF (IEC60958) Frame Format 23-3
23.6 Register 23-5
23.7 Register Descriptions 23-6
23.7.1 Control Register (CTRL) 23-6
23.7.2 Status Register (STAT) 23-9
23.7.3 Transmitter Channel 1 Audio Register (TLCA) 23-11
23.7.4 Transmitter Channel 2 Audio Register (TRCA) 23-12
23.7.5 Transmitter DMA Audio Data Register (TDAD) 23-12
23.7.6 Transmitter User Data Register (TUI) 23-13
23.7.7 Transmitter Channel 1 Status Register (TLCS) 23-14
23.7.8 Transmitter Channel 2 Status Register (TRCS) 23-15
23.7.9 Receiver Channel 1 Audio Register (RLCA) 23-16
23.7.10 Receiver Channel 2 Audio Register (RRCA) 23-16
23.7.11 Receiver DMA Audio Data (RDAD) 23-17
23.7.12 Receiver User Data Register (RUI) 23-18
23.7.13 Receiver Channel 1 Status Register (RLCS) 23-19
23.7.14 Receiver Channel 2 Status Register (RRCS) 23-20
23.8 Functional Description—Transmitter 23-21
23.8.1 Transmitter Module 23-21
23.8.2 Transmitter Module Initialization 23-21
23.8.3 Initial Settings for Transmitter Module 23-22
23.8.4 Transmitter Module Data Transfer 23-22
23.9 Functional Description—Receiver 23-24
23.9.1 Receiver Module 23-24
23.9.2 Receiver Module Initialization 23-24
23.9.3 Receiver Module Data Transfer 23-25
23.10 Disabling the Module 23-27
23.10.1 Transmitter and Receiver Idle 23-27
23.11 Compressed Mode Data 23-27
23.12 References 23-27
23.13 Usage Notes 23-28
23.13.1 Clearing TUIR 23-28

chanhvinh.com

23.13.2 Frequency of Clock Input for Audio 23-28
24. CD-ROM Decoder 24-1
24.1 Features 24-1
24.1.1 Formats Supported by CD-ROM Decoder 24-2
24.2 Block Diagrams 24-2
24.3 Register Descriptions 24-5
24.3.1 Enable Control Register (CROMEN) 24-7
24.3.2 Sync Code-Based Synchronization Control Register (CROMSY0) 24-8
24.3.3 Decoding Mode Control Register (CROMCTL0) 24-9
24.3.4 EDC/ECC Check Control Register (CROMCTL1) 24-10
24.3.5 Automatic Decoding Stop Control Register (CROMCTL3) 24-11
24.3.6 Decoding Option Setting Control Register (CROMCTL4) 24-12
24.3.7 HEAD20 to HEAD22 Representation Control Register (CROMCTL5) 24-13
24.3.8 Sync Code Status Register (CROMST0) 24-13
24.3.9 Post-ECC Header Error Status Register (CROMST1) 24-14
24.3.10 Post-ECC Subheader Error Status Register (CROMST3) 24-14
24.3.11 Header/Subheader Validity Check Status Register (CROMST4) 24-15
24.3.12 Mode Determination and Link Sector Detection Status Register (CROMST5) 24-16
24.3.13 ECC/EDC Error Status Register (CROMST6) 24-17
24.3.14 Buffer Status Register (CBUFST0) 24-17
24.3.15 Decoding Stoppage Source Status Register (CBUFST1) 24-18
24.3.16 Buffer Overflow Status Register (CBUFST2) 24-18
24.3.17 Pre-ECC Correction Header: Minutes Data Register (HEAD00) 24-19
24.3.18 Pre-ECC Correction Header: Seconds Data Register (HEAD01) 24-19
24.3.19 Pre-ECC Correction Header: Frames (1/75 Second) Data Register (HEAD02) 24-19
24.3.20 Pre-ECC Correction Header: Mode Data Register (HEAD03) 24-19
24.3.21 Pre-ECC Correction Subheader: File Number (Byte 16) Data Register (SHEAD00) 24-20
24.3.22 Pre-ECC Correction Subheader: Channel Number (Byte 17) Data Register (SHEAD01) 24-20
24.3.23 Pre-ECC Correction Subheader: Sub-Mode (Byte 18) Data Register (SHEAD02) 24-20
24.3.24 Pre-ECC Correction Subheader: Data Type (Byte 19) Data Register (SHEAD03) 24-21
24.3.25 Pre-ECC Correction Subheader: File Number (Byte 20) Data Register (SHEAD04) 24-21
24.3.26 Pre-ECC Correction Subheader: Channel Number (Byte 21) Data Register (SHEAD05) 24-21
24.3.27 Pre-ECC Correction Subheader: Sub-Mode (Byte 22) Data Register (SHEAD06) 24-22
24.3.28 Pre-ECC Correction Subheader: Data Type (Byte 23) Data Register (SHEAD07) 24-22
24.3.29 Post-ECC Correction Header: Minutes Data Register (HEAD20) 24-22
24.3.30 Post-ECC Correction Header: Seconds Data Register (HEAD21) 24-23
24.3.31 Post-ECC Correction Header: Frames (1/75 Second) Data Register (HEAD22) 24-23
24.3.32 Post-ECC Correction Header: Mode Data Register (HEAD23) 24-23
24.3.33 Post-ECC Correction Subheader: File Number (Byte 16) Data Register (SHEAD20) 24-24
24.3.34 Post-ECC Correction Subheader: Channel Number (Byte 17) Data Register (SHEAD21) 24-24
24.3.35 Post-ECC Correction Subheader: Sub-Mode (Byte 18) Data Register (SHEAD22) 24-24

chanhvinh.com

24.3.36 Post-ECC Correction Subheader: Data Type (Byte 19) Data Register (SHEAD23) 24-25
24.3.37 Post-ECC Correction Subheader: File Number (Byte 20) Data Register (SHEAD24) 24-25
24.3.38 Post-ECC Correction Subheader: Channel Number (Byte 21) Data Register (SHEAD25) ... 24-25
24.3.39 Post-ECC Correction Subheader: Sub-Mode (Byte 22) Data Register (SHEAD26) 24-26
24.3.40 Post-ECC Correction Subheader: Data Type (Byte 23) Data Register (SHEAD27) 24-26
24.3.41 Automatic Buffering Setting Control Register (CBUFCTL0) 24-27
24.3.42 Automatic Buffering Start Sector Setting: Minutes Control Register (CBUFCTL1) 24-28
24.3.43 Automatic Buffering Start Sector Setting: Seconds Control Register (CBUFCTL2) 24-28
24.3.44 Automatic Buffering Start Sector Setting: Frames Control Register (CBUFCTL3) 24-28
24.3.45 ISY Interrupt Source Mask Control Register (CROMST0M) 24-29
24.3.46 CD-ROM Decoder Reset Control Register (ROMDECRST) 24-29
24.3.47 CD-ROM Decoder Reset Status Register (RSTSTAT) 24-30
24.3.48 Serial Sound Interface Data Control Register (SSI) 24-31
24.3.49 Interrupt Flag Register (INTHOLD) 24-32
24.3.50 Interrupt Source Mask Control Register (INHINT) 24-33
24.3.51 CD-ROM Decoder Stream Data Input Register (STRMDIN0) 24-33
24.3.52 CD-ROM Decoder Stream Data Input Register (STRMDIN2) 24-34
24.3.53 CD-ROM Decoder Stream Data Output Register (STRMDOUT0) 24-34
24.4 Operation 24-35
24.4.1 Endian Conversion for Data in the Input Stream 24-35
24.4.2 Sync Code Maintenance Function 24-36
24.4.3 Error Correction 24-38
24.4.4 Automatic Decoding Stop Function 24-39
24.4.5 Buffering Format 24-40
24.4.6 Target-Sector Buffering Function 24-41
24.5 Interrupt Sources 24-43
24.5.1 Interrupt and DMA Transfer Request Signals 24-43
24.5.2 Timing of Status Registers Updates 24-44
24.6 Usage Notes 24-45
24.6.1 Stopping and Resuming Buffering Alone during Decoding 24-45
24.6.2 Setting Sync Code Status Register (CROMST0) 24-45
24.6.3 Link Blocks 24-45
24.6.4 Stopping and Resuming CD-DSP Operation 24-45
24.6.5 Note on Clearing the IREADY Flag 24-46
24.6.6 Note on Stream Data Transfer (1) 24-46
24.6.7 Note on Stream Data Transfer (2) 24-46
24.6.8 Note on Software Reset 24-46
25. LIN Interface 25-1
25.1 Features 25-1
25.1.1 Channels 25-1
25.1.2 Register Addresses 25-1

chanhvinh.com

25.1.3 Clock Supply 25-1
25.1.4 Interrupts and DMA 25-2
25.1.5 I/O Signals 25-2
25.2 Function 25-3
25.2.1 Block Diagram 25-5
25.2.2 Description of Blocks 25-5
25.3 Registers 25-6
25.3.1 LIN Master Related Registers 25-7
25.4 Interrupt Sources 25-33
25.5 Modes 25-34
25.6 LIN Reset Mode 25-36
25.7 LIN Mode 25-37
25.7.1 LIN Master Mode 25-39
25.7.2 Data Transmission/Reception 25-42
25.7.3 Transmission/Reception Data Buffering 25-44
25.7.4 Wake-up Transmission/Reception 25-47
25.7.5 Status 25-49
25.7.6 Error Status 25-50
25.8 LIN Self-Test Mode 25-52
25.8.1 Change to LIN Self-Test Mode 25-54
25.8.2 Transmission in LIN Master Self-Test Mode 25-55
25.8.3 Reception in LIN Master Self-Test Mode 25-56
25.8.4 Terminating LIN Self-Test Mode 25-57
25.9 Baud Rate Generator 25-58
25.9.1 LIN Master Mode 25-58
25.9.2 Noise Filter 25-59
26. Ethernet Controller 26-1
26.1 Features 26-1
26.2 Input/Output Pins 26-3
26.3 Register Descriptions 26-4
26.3.1 Software Reset Register (ARSTR) 26-7
26.3.2 E-MAC Mode Register (ECMR) 26-8
26.3.3 E-MAC Status Register (ECSR) 26-10
26.3.4 E-MAC Interrupt Permission Register (ECSIPR) 26-11
26.3.5 PHY Interface Register (PIR) 26-12
26.3.6 MAC Address High Register (MAHR) 26-12
26.3.7 MAC Address Low Register (MALR) 26-13
26.3.8 Receive Frame Length Register (RFLR) 26-14
26.3.9 CRC Error Frame Receive Counter Register (CEFCR) 26-15
26.3.10 Frame Receive Error Counter Register (FRECR) 26-15
26.3.11 Too-Short Frame Receive Counter Register (TSFRCR) 26-16

chanhvinh.com

26.3.12 Too-Long Frame Receive Counter Register (TLFRCR) 26-16
26.3.13 Residual-Bit Frame Receive Counter Register (RFCR) 26-17
26.3.14 Multicast Address Frame Receive Counter Register (MAFCR) 26-17
26.3.15 Automatic PAUSE Frame Register (APR) 26-18
26.3.16 Manual PAUSE Frame Register (MPR) 26-19
26.3.17 Automatic PAUSE Frame Retransmit Count Register (TPAUSER) 26-19
26.3.18 PAUSE Frame Transmit Counter Register (PFTCR) 26-20
26.3.19 PAUSE Frame Receive Counter Register (PFRCR) 26-20
26.3.20 TSU Counter Reset Register (TSU_CTRST) 26-21
26.3.21 CAM Entry Table Specification Enable Register (Common) (TSU_FWSLC) 26-22
26.3.22 VLANtag Set Register (TSU_VTAG0) 26-23
26.3.23 CAM Entry Table Busy Register (TSU_ADSBSY) 26-24
26.3.24 CAM Entry Table Enable Register (TSU_TEN) 26-25
26.3.25 CAM Entry Table POST 1 Register (TSU_POST1) 26-27
26.3.26 CAM Entry Table POST 2 Register (TSU_POST2) 26-28
26.3.27 CAM Entry Table POST 3 Register (TSU_POST3) 26-29
26.3.28 CAM Entry Table POST 4 Register (TSU_POST4) 26-30
26.3.29 CAM Entry Table 0H to 31H Registers (TSU_ADRH0 to TSU_ADRH31) 26-32
26.3.30 CAM Entry Table 0L to 31L Registers (TSU_ADRL0 to TSU_ADRL31) 26-32
26.3.31 Transmit Frame Counter Register (Normal Transmission Only) (TXNLCR0) 26-33
26.3.32 Transmit Frame Counter Register (Normal and Erroneous Transmission) (TXALCR0) 26-33
26.3.33 Receive Frame Counter Register (Normal Reception Only) (RXNLCR0) 26-34
26.3.34 Receive Frame Counter Register (Normal and Erroneous Reception) (RXALCR0) 26-34
26.3.35 E-DMAC Start Register (EDSR) 26-35
26.3.36 E-DMAC Mode Register (EDMR) 26-36
26.3.37 E-DMAC Transmit Request Register (EDTRR) 26-37
26.3.38 E-DMAC Receive Request Register (EDRRR) 26-38
26.3.39 Transmit Descriptor List Start Address Register (TDLAR) 26-39
26.3.40 Receive Descriptor List Start Address Register (RDLAR) 26-39
26.3.41 E-MAC/E-DMAC Status Register (EESR) 26-40
26.3.42 E-MAC/E-DMAC Status Interrupt Permission Register (EESIPR) 26-43
26.3.43 Transmit/Receive Status Copy Enable Register (TRSCER) 26-45
26.3.44 Receive Missed-Frame Counter Register (RMFCR) 26-46
26.3.45 Transmit FIFO Threshold Register (TFTR) 26-47
26.3.46 FIFO Depth Register (FDR) 26-48
26.3.47 Receiving Method Control Register (RMCR) 26-49
26.3.48 Receive Descriptor Fetch Address Register (RDFAR) 26-49
26.3.49 Receive Descriptor Finished Address Register (RDFXR) 26-50
26.3.50 Receive Descriptor Final Flag Register (RDFFR) 26-50
26.3.51 Transmit Descriptor Fetch Address Register (TDFAR) 26-51
26.3.52 Transmit Descriptor Finished Address Register (TDFXR) 26-51

chanhvinh.com

26.3.53 Transmit Descriptor Final Flag Register (TDFFR) 26-52
26.3.54 Overflow Alert FIFO Threshold Register (FCFTR) 26-53
26.3.55 Receive Data Padding Insert Register (RPADIR) 26-54
26.3.56 Intelligent Checksum Mode Register (CSMR) 26-55
26.3.57 Intelligent Checksum Skipped Bytes Monitor Register (CSSBM) 26-56
26.3.58 Intelligent Checksum Monitor Register (CSSMR) 26-56
26.4 Operation 26-57
26.4.1 Descriptors and Descriptor List 26-59
26.4.2 Transmission 26-72
26.4.3 Reception 26-76
26.4.4 CAM Function 26-80
26.4.5 Transmit Processing of Multi-Buffer Frame (Single-Frame/Multi-Descriptor) 26-81
26.4.6 Padding Insertion in Receive Data 26-82
26.4.7 Interrupt Processing 26-83
26.4.8 Activation Procedure 26-84
26.4.9 Flow Control 26-86
26.4.10 Intelligent Checksum Calculation Function 26-87
26.5 Connection to PHY-LSI 26-89
26.5.1 MII Frame Transmission/Reception Timing 26-89
26.5.2 Accessing MII Registers 26-90
26.6 Usage Notes 26-93
26.6.1 Checksum Calculation of Ethernet Frames 26-93
26.6.2 Notes on Using the Intelligent Checksum Function 26-94
26.6.3 Software Reset 26-94
27. A/D Converter 27-1
27.1 Features 27-1
27.2 Input/Output Pins 27-3
27.3 Register Descriptions 27-4
27.3.1 A/D Data Registers A to H (ADDRA to ADDRH) 27-5
27.3.2 A/D Comparison Upper Limit Value Registers A to H (ADCMPHA to ADCMPHH) 27-6
27.3.3 A/D Comparison Lower Limit Value Registers A to H (ADCMPLA to ADCMPLH) 27-7
27.3.4 A/D Control/Status Register (ADCSR) 27-8
27.3.5 A/D Comparison Interrupt Enable Register (ADCMPER) 27-10
27.3.6 A/D Comparison Status Register (ADCMPSR) 27-11
27.4 Operation 27-12
27.4.1 Single Mode 27-12
27.4.2 Multi Mode 27-14
27.4.3 Scan Mode 27-16
27.4.4 A/D Converter Activation by External Trigger or Multi-Function Timer Pulse Unit 2 27-18
27.4.5 Input Sampling and A/D Conversion Time 27-18

chanhvinh.com

27.4.6 External Trigger Input Timing 27-20
27.5 Interrupt Sources and DMA Transfer Request 27-21
27.6 Definitions of A/D Conversion Accuracy 27-22
27.7 Usage Notes 27-23
27.7.1 Module Standby Mode Setting 27-23
27.7.2 Setting Analog Input Voltage 27-23
27.7.3 Notes on Board Design 27-23
27.7.4 Processing of Analog Input Pins 27-24
27.7.5 Permissible Signal Source Impedance 27-25
27.7.6 Influences on Absolute Precision 27-25
27.7.7 Usage Note on Port Pins 27-25
28. USB2.0 Host/Function Module 28-1
28.1 Features 28-1
28.2 Input/Output Pins 28-3
28.3 Register Descriptions 28-4
28.3.1 System Configuration Control Register (SYSCFG0) 28-10
28.3.2 CPU Bus Wait Setting Register (BUSWAIT) 28-12
28.3.3 System Configuration Status Register (SYSSTS0) 28-13
28.3.4 Device State Control Register 0 (DVSTCTR0) 28-14
28.3.5 Test Mode Register (TESTMODE) 28-17
28.3.6 DMAn-FIFO Bus Configuration Registers (D0FBCFG, D1FBCFG) 28-18
28.3.7 FIFO Port Registers (CFIFO, D0FIFO, D1FIFO) 28-20
28.3.8 FIFO Port Select Registers (CFIFOSEL, D0FIFOSEL, D1FIFOSEL) 28-22
28.3.9 FIFO Port Control Registers (CFIFOCTR, D0FIFOCTR, D1FIFOCTR) 28-26
28.3.10 Interrupt Enable Register 0 (INTENB0) 28-28
28.3.11 Interrupt Enable Register 1 (INTENB1) 28-29
28.3.12 BRDY Interrupt Enable Register (BRDYENB) 28-30
28.3.13 NRDY Interrupt Enable Register (NRDYENB) 28-31
28.3.14 BEMP Interrupt Enable Register (BEMPENB) 28-31
28.3.15 SOF Output Configuration Register (SOFCFG) 28-32
28.3.16 Interrupt Status Register 0 (INTSTS0) 28-33
28.3.17 Interrupt Status Register 1 (INTSTS1) 28-36
28.3.18 BRDY Interrupt Status Register (BRDYSTS) 28-38
28.3.19 NRDY Interrupt Status Register (NRDYSTS) 28-39
28.3.20 BEMP Interrupt Status Register (BEMPSTS) 28-39
28.3.21 Frame Number Register (FRMNUM) 28-40
28.3.22 µFrame Number Register (UFRMNUM) 28-41
28.3.23 USB Address Register (USBADDR) 28-41
28.3.24 USB Request Type Register (USBREQ) 28-42
28.3.25 USB Request Value Register (USBVAL) 28-43
28.3.26 USB Request Index Register (USBINDX) 28-43

chanhvinh.com

28.3.27 USB Request Length Register (USBLENG) 28-44
28.3.28 DCP Configuration Register (DCPCFG) 28-44
28.3.29 DCP Maximum Packet Size Register (DCPMAXP) 28-47
28.3.30 DCP Control Register (DCPCTR) 28-48
28.3.31 Pipe Window Select Register (PIPESEL) 28-52
28.3.32 Pipe Configuration Register (PIPECFG) 28-53
28.3.33 Pipe Buffer Setting Register (PIPEBUF) 28-57
28.3.34 Pipe Maximum Packet Size Register (PIPEMAXP) 28-59
28.3.35 Pipe Timing Control Register (PIPEPERI) 28-60
28.3.36 PIPEn Control Registers (PIPEnCTR) (n = 1 to F) 28-61
28.3.37 PIPEn Transaction Counter Enable Registers (PIPEnTRE) (n = 1 to 5, 9, and A to F) 28-70
28.3.38 PIPEn Transaction Counter Registers (PIPEnTRN) (n = 1 to 5, 9, and A to F) 28-71
28.3.39 Device Address n Configuration Registers (DEVADDn) (n = 0 to 9, and A) 28-72
28.3.40 UTMI Suspend Mode Register (SUSPMODE) 28-73
28.3.41 D0FIFO Continuous Transfer Port Register n (D0FIFOBn)
D1FIFO Continuous Transfer Port Register n (D1FIFOBn) (n = 0 to 7) 28-74
28.4 Operation 28-75
28.4.1 System Control and Oscillation Control 28-75
28.4.2 Interrupt Functions 28-77
28.4.3 Pipe Control 28-89
28.4.4 FIFO Buffer Memory 28-95
28.4.5 FIFO Port Functions 28-97
28.4.6 Control Transfers (DCP) 28-100
28.4.7 Bulk Transfers (PIPE1 to PIPE5, PIPE9 to PIPE15) 28-102
28.4.8 Interrupt Transfers (PIPE6 to PIPE9, PIPE10) 28-104
28.4.9 Isochronous Transfers (PIPE1 and PIPE2) 28-105
28.4.10 SOF Interpolation Function 28-113
29. Video Display Controller 5 (1): Overview 29-1
29.1 Features 29-1
29.2 Block Diagram 29-3
29.3 Input/Output Pins 29-4
29.4 Clocks 29-4
29.5 Hsync and Vsync Signals 29-5
29.5.1 External Input Vsync 29-5
29.5.2 Free-Running Vsync 29-6
29.5.3 Usage Note on Changing Vsync Signal Selections 29-8
30. Video Display Controller 5 (2): Input Controller 30-1
30.1 Input Controller Functions 30-1
30.1.1 Overview of Functions 30-1
30.1.2 Updating Registers of External Signal Input Block and Sync Signal Adjustment Block 30-2
30.1.3 Controlling Input 30-2

chanhvinh.com

30.1.4 Controlling Externally Input Video Signals 30-3
30.1.5 Selecting Clock Edge for Externally Input Signals 30-4
30.1.6 Externally Input Sync Signal Inversion Control 30-4
30.1.7 Bit Allocation of Externally Input Video Image Signals 30-5
30.1.8 Typical Signal Timing of BT601 Format 30-9
30.1.9 Typical Signal Timing of BT656 Format 30-12
30.1.10 SAV/EAV Code in BT656 Format 30-14
30.1.11 BT656 Progressive Format 30-17
30.1.12 BT656/BT601/YCbCr422 Format Setting 30-21
30.1.13 YCbCr444/RBG888/666/565 Input Timing 30-24
30.1.14 Field Differentiation and Vsync Signal Phase Adjustment 30-26
30.1.15 Vsync Signal Delay Adjustment in Line Units 30-27
30.1.16 Sync Signal Delay Adjustment 30-27
30.1.17 Horizontal Noise Reduction 30-28
30.1.18 Color Matrix 30-30
30.2 Register Descriptions 30-33
30.2.1 External Input Block Register Update Control Register (INP_UPDATE) 30-34
30.2.2 Input Select Control Register (INP_SEL_CNT) 30-35
30.2.3 External Input Sync Signal Control Register (INP_EXT_SYNC_CNT) 30-36
30.2.4 Vsync Signal Phase Adjustment Register (INP_VSYNC_PH_ADJ) 30-37
30.2.5 Sync Signal Delay Adjustment Register (INP_DLY_ADJ) 30-37
30.2.6 Image Quality Adjustment Block Register Update Control Register (IMGCNT_UPDATE) 30-38
30.2.7 NR Control Register 0 (IMGCNT_NR_CNT0) 30-39
30.2.8 NR Control Register 1 (IMGCNT_NR_CNT1) 30-40
30.2.9 Image Quality Adjustment Block Matrix Mode Register (IMGCNT_MTX_MODE) 30-41
30.2.10 Image Quality Adjustment Block Matrix YG Adjustment Register 0 (IMGCNT_MTX_YG_ADJ0) 30-41
30.2.11 Image Quality Adjustment Block Matrix YG Adjustment Register 1 (IMGCNT_MTX_YG_ADJ1) 30-42
30.2.12 Image Quality Adjustment Block Matrix CBB Adjustment Register 0 (IMGCNT_MTX_CBB_ADJ0) 30-42
30.2.13 Image Quality Adjustment Block Matrix CBB Adjustment Register 1 (IMGCNT_MTX_CBB_ADJ1) 30-43
30.2.14 Image Quality Adjustment Block Matrix CRR Adjustment Register 0 (IMGCNT_MTX_CRR_ADJ0) 30-44
30.2.15 Image Quality Adjustment Block Matrix CRR Adjustment Register 1 (IMGCNT_MTX_CRR_ADJ1) 30-45
30.3 Usage Methods 30-46
30.3.1 Input Format Adjustment Method 30-46
30.3.2 Usage Method of Conversion Color Matrix 30-48
31. Video Display Controller 5 (3): Scaler 31-1
31.1 Scaler 31-1

chanhvinh.com

31.1.1 Overview of Functions 31-1
31.1.2 Register Control 31-2
31.1.3 Synchronization Control 31-3
31.1.4 Setting Angle of View 31-8
31.1.5 Scaling Settings 31-12
31.1.6 Horizontal Prefilter 31-14
31.1.7 Horizontal Scale-Down 31-14
31.1.8 Vertical Scale-Down 31-16
31.1.9 Horizontal Scale Up 31-18
31.1.10 Vertical Scale-Up 31-19
31.1.11 IP Conversion 31-21
31.1.12 Control of Interrupt on Specified Image Line before Scaling-down, and Reading of Current Image Line before Scaling-down 31-23
31.1.13 Trimming 31-24
31.1.14 Screen Synthesis 31-25
31.1.15 Selecting Format for Writing Video Image Signals to Frame Buffer 31-26
31.1.16 Horizontal Mirroring and Rotation 31-27
31.1.17 Writing to Frame Buffer 31-27
31.1.18 Selecting a Scaling-up Process or Graphics 0 Process 31-33
31.1.19 Selecting Field for Frame Buffer Reading 31-34
31.1.20 Pointer Buffer and Frame Buffer Reading Processing 31-35
31.2 Register Descriptions 31-37
31.2.1 SCL0 Register Update Control Register (SC0_SCL0_UPDATE) 31-39
31.2.2 Mask Control Register (SC0_SCL0_FRC1) 31-40
31.2.3 Missing Vsync Compensation Control Register (SC0_SCL0_FRC2) 31-40
31.2.4 Output Sync Select Register (SC0_SCL0_FRC3) 31-41
31.2.5 Free-Running Period Control Register (SC0_SCL0_FRC4) 31-42
31.2.6 Output Delay Control Register (SC0_SCL0_FRC5) 31-42
31.2.7 Full-Screen Vertical Size Register (SC0_SCL0_FRC6) 31-43
31.2.8 Full-Screen Horizontal Size Register (SC0_SCL0_FRC7) 31-44
31.2.9 Vsync Detection Register (SC0_SCL0_FRC9) 31-45
31.2.10 Status Monitor 0 Register (SC0_SCL0_MON0) 31-45
31.2.11 Interrupt Control Register (SC0_SCL0_INT) 31-46
31.2.12 Scaling-Down Control Register (SC0_SCL0_DS1) 31-46
31.2.13 Vertical Capture Size Register (SC0_SCL0_DS2) 31-47
31.2.14 Horizontal Capture Size Register (SC0_SCL0_DS3) 31-48
31.2.15 Horizontal Scale Down Register (SC0_SCL0_DS4) 31-49
31.2.16 Initial Vertical Phase Register (SC0_SCL0_DS5) 31-50
31.2.17 Vertical Scaling Register (SC0_SCL0_DS6) 31-51
31.2.18 Scaling-Down Control Block Output Size Register (SC0_SCL0_DS7) 31-52
31.2.19 Scaling-Up Control Register (SC0_SCL0_US1) 31-53

chanhvinh.com

31.2.20 Output Image Vertical Size Register (SC0_SCL0_US2) 31-54
31.2.21 Output Image Horizontal Size Register (SC0_SCL0_US3) 31-55
31.2.22 Scaling-Up Control Block Input Size Register (SC0_SCL0_US4) 31-56
31.2.23 Horizontal Scale Up Register (SC0_SCL0_US5) 31-56
31.2.24 Horizontal Scale Up Initial Phase Register (SC0_SCL0_US6) 31-57
31.2.25 Trimming Register (SC0_SCL0_US7) 31-57
31.2.26 Frame Buffer Read Select Register (SC0_SCL0_US8) 31-58
31.2.27 Background Color Register (SC0_SCL0_OVR1) 31-58
31.2.28 SCL1 Register Update Control Register (SC0_SCL1_UPDATE) 31-59
31.2.29 Writing Mode Register (SC0_SCL1_WR1) 31-60
31.2.30 Write Address Register 1T (SC0_SCL1_WR2) 31-61
31.2.31 Write Address Register 2T (SC0_SCL1_WR3) 31-62
31.2.32 Write Address Register 3T (SC0_SCL1_WR4) 31-63
31.2.33 Frame Sub-Sampling Register (SC0_SCL1_WR5) 31-64
31.2.34 Bit Reduction Register (SC0_SCL1_WR6) 31-65
31.2.35 Write Detection Register (SC0_SCL1_WR7) 31-65
31.2.36 Write Address Register 1B (SC0_SCL1_WR8) 31-66
31.2.37 Write Address Register 2B (SC0_SCL1_WR9) 31-67
31.2.38 Write Address Register 3B (SC0_SCL1_WR10) 31-68
31.2.39 Write Detection Register B (SC0_SCL1_WR11) 31-68
31.2.40 Status Monitor 1 Register (SC0_SCL1_MON1) 31-69
31.2.41 Pointer Buffer 0 Register (SC0_SCL1_PBUF0) 31-69
31.2.42 Pointer Buffer 1 Register (SC0_SCL1_PBUF1) 31-70
31.2.43 Pointer Buffer 2 Register (SC0_SCL1_PBUF2) 31-70
31.2.44 Pointer Buffer 3 Register (SC0_SCL1_PBUF3) 31-71
31.2.45 Pointer Buffer and Field Information Register (SC0_SCL1_PBUF_FLD) 31-71
31.2.46 Pointer Buffer Control Register (SC0_SCL1_PBUF_CNT) 31-72
31.2.47 Graphics 0 Register Update Control Register (GR0_UPDATE) 31-73
31.2.48 Frame Buffer Read Control Register (Graphics 0) (GR0_FLM_RD) 31-73
31.2.49 Frame Buffer Control Register 1 (Graphics 0) (GR0_FLM1) 31-74
31.2.50 Frame Buffer Control Register 2 (Graphics 0) (GR0_FLM2) 31-75
31.2.51 Frame Buffer Control Register 3 (Graphics 0) (GR0_FLM3) 31-76
31.2.52 Frame Buffer Control Register 4 (Graphics 0) (GR0_FLM4) 31-77
31.2.53 Frame Buffer Control Register 5 (Graphics 0) (GR0_FLM5) 31-78
31.2.54 Frame Buffer Control Register 6 (Graphics 0) (GR0_FLM6) 31-79
31.2.55 Alpha Blending Control Register 1 (Graphics 0) (GR0_AB1) 31-81
31.2.56 Alpha Blending Control Register 2 (Graphics 0) (GR0_AB2) 31-82
31.2.57 Alpha Blending Control Register 3 (Graphics 0) (GR0_AB3) 31-82
31.2.58 Alpha Blending Control Register 7 (Graphics 0) (GR0_AB7) 31-83
31.2.59 Alpha Blending Control Register 8 (Graphics 0) (GR0_AB8) 31-84
31.2.60 Alpha Blending Control Register 9 (Graphics 0) (GR0_AB9) 31-84

chanhvinh.com

31.2.61 Alpha Blending Control Register 10 (Graphics 0) (GR0_AB10) 31-85
31.2.62 Alpha Blending Control Register 11 (Graphics 0) (GR0_AB11) 31-86
31.2.63 Background Color Control Register (Graphics 0) (GR0_BASE) 31-86
31.2.64 CLUT Table Control Register (Graphics 0) (GR0_CLUT) 31-87
31.3 Usage Method 31-88
31.3.1 Scaling Setting Example for 525i Video Input and VGA-Size (640 x 480) Video Output 31-88
31.3.2 Scaling Setting Example for Graphics Display 31-91
31.3.3 Scaling Setting Example for Scaled-up Graphics Display 31-93
32. Video Display Controller 5 (4): Image Quality Improver 32-1
32.1 Image Quality Improver 32-1
32.1.1 Overview of Functions 32-1
32.1.2 Register Update Control 32-1
32.1.3 Black Stretch 32-2
32.1.4 Enhancer 32-3
32.1.5 Color Matrix 32-9
32.2 Register Description 32-11
32.2.1 Register Update Control Register in Image Quality Improver (ADJ0_UPDATE) 32-12
32.2.2 Black Stretch Register (ADJ0_BKSTR_SET) 32-13
32.2.3 Enhancer Timing Adjustment Register 1 (ADJ0_ENH_TIM1) 32-14
32.2.4 Enhancer Timing Adjustment Register 2 (ADJ0_ENH_TIM2) 32-14
32.2.5 Enhancer Timing Adjustment Register 3 (ADJ0_ENH_TIM3) 32-15
32.2.6 Enhancer Sharpness Register 1 (ADJ0_ENH_SHP1) 32-15
32.2.7 Enhancer Sharpness Register 2 (ADJ0_ENH_SHP2) 32-16
32.2.8 Enhancer Sharpness Register 3 (ADJ0_ENH_SHP3) 32-17
32.2.9 Enhancer Sharpness Register 4 (ADJ0_ENH_SHP4) 32-18
32.2.10 Enhancer Sharpness Register 5 (ADJ0_ENH_SHP5) 32-18
32.2.11 Enhancer Sharpness Register 6 (ADJ0_ENH_SHP6) 32-19
32.2.12 Enhancer LTI Register 1 (ADJ0_ENH_LTI1) 32-20
32.2.13 Enhancer LTI Register 2 (ADJ0_ENH_LTI2) 32-21
32.2.14 Matrix Mode Register in Image Quality Improver (ADJ0_MTX_MODE) 32-22
32.2.15 Matrix YG Control Register 0 in Image Quality Improver (ADJ0_MTX_YG_ADJ0) 32-22
32.2.16 Matrix YG Control Register 1 in Image Quality Improver (ADJ0_MTX_YG_ADJ1) 32-23
32.2.17 Matrix CBB Control Register 0 in Image Quality Improver (ADJ0_MTX_CBB_ADJ0) 32-24
32.2.18 Matrix CBB Control Register 1 in Image Quality Improver (ADJ0_MTX_CBB_ADJ1) 32-25
32.2.19 Matrix CRR Control Register 0 in Image Quality Improver (ADJ0_MTX_CRR_ADJ0) 32-26
32.2.20 Matrix CRR Control Register 1 in Image Quality Improver (ADJ0_MTX_CRR_ADJ1) 32-27
32.3 Usage Method 32-28
32.3.1 Black Stretch Usage Method 32-28
32.3.2 LTI Processing of Enhancer 32-28
32.3.3 Sharpness Processing of Enhancer 32-29
32.3.4 Setting Method for Color Matrix Data Conversion 32-30

chanhvinh.com

33. Video Display Controller 5 (5): Image Synthesizer 33-1
33.1 Image Synthesizer 33-1
33.1.1 Overview of Functions 33-1
33.1.2 Graphics Data Read Control 33-2
33.1.3 Setting Graphics Display Area 33-12
33.1.4 Interrupt Generation at Specified Line 33-13
33.1.5 Formats of Frame Buffer Read Signals and Corresponding Alpha Blending Types 33-13
33.1.6 Display Selection 33-14
33.1.7 Background Color Display Processing 33-16
33.1.8 Lower-Layer Graphics Display Processing 33-16
33.1.9 Current Graphics Display Processing 33-16
33.1.10 Display with Alpha Blending in a Rectangular Area 33-17
33.1.11 RGB-Index Chroma-Key Processing 33-19
33.1.12 CLUT-Index Chroma-Key Processing 33-20
33.1.13 Display with Alpha Blending in One-Pixel Units 33-21
33.1.14 Alpha Blending Calculation 33-22
33.1.15 CLUT Table 33-22
33.1.16 Multiplication Processing with Current Alpha at Alpha Blending in Rectangular Area 33-23
33.1.17 Selection of Lower-Layer Graphics in VIN Synthesizer 33-23
33.2 Register Descriptions 33-24
33.2.1 Graphics 2 Register Update Control Register (GR2_UPDATE) 33-28
33.2.2 Frame Buffer Read Control Register (Graphics 2) (GR2_FLM_RD) 33-28
33.2.3 Frame Buffer Control Register 1 (Graphics 2) (GR2_FLM1) 33-29
33.2.4 Frame Buffer Control Register 2 (Graphics 2) (GR2_FLM2) 33-30
33.2.5 Frame Buffer Control Register 3 (Graphics 2) (GR2_FLM3) 33-30
33.2.6 Frame Buffer Control Register 4 (Graphics 2) (GR2_FLM4) 33-31
33.2.7 Frame Buffer Control Register 5 (Graphics 2) (GR2_FLM5) 33-31
33.2.8 Frame Buffer Control Register 6 (Graphics 2) (GR2_FLM6) 33-32
33.2.9 Alpha Blending Control Register 1 (Graphics 2) (GR2_AB1) 33-33
33.2.10 Alpha Blending Control Register 2 (Graphics 2) (GR2_AB2) 33-34
33.2.11 Alpha Blending Control Register 3 (Graphics 2) (GR2_AB3) 33-35
33.2.12 Alpha Blending Control Register 4 (Graphics 2) (GR2_AB4) 33-35
33.2.13 Alpha Blending Control Register 5 (Graphics 2) (GR2_AB5) 33-36
33.2.14 Alpha Blending Control Register 6 (Graphics 2) (GR2_AB6) 33-36
33.2.15 Alpha Blending Control Register 7 (Graphics 2) (GR2_AB7) 33-37
33.2.16 Alpha Blending Control Register 8 (Graphics 2) (GR2_AB8) 33-37
33.2.17 Alpha Blending Control Register 9 (Graphics 2) (GR2_AB9) 33-38
33.2.18 Alpha Blending Control Register 10 (Graphics 2) (GR2_AB10) 33-38
33.2.19 Alpha Blending Control Register 11 (Graphics 2) (GR2_AB11) 33-39
33.2.20 Background Color Control Register (Graphics 2) (GR2_BASE) 33-39
33.2.21 CLUT Table Control Register (Graphics 2) (GR2_CLUT) 33-40

chanhvinh.com

33.2.22 Status Monitor Register (Graphics 2) (GR2_MON) 33-40
33.2.23 Graphics 3 Register Update Control Register (GR3_UPDATE) 33-41
33.2.24 Frame Buffer Read Control Register (Graphics 3) (GR3_FLM_RD) 33-41
33.2.25 Frame Buffer Control Register 1 (Graphics 3) (GR3_FLM1) 33-42
33.2.26 Frame Buffer Control Register 2 (Graphics 3) (GR3_FLM2) 33-43
33.2.27 Frame Buffer Control Register 3 (Graphics 3) (GR3_FLM3) 33-43
33.2.28 Frame Buffer Control Register 4 (Graphics 3) (GR3_FLM4) 33-44
33.2.29 Frame Buffer Control Register 5 (Graphics 3) (GR3_FLM5) 33-44
33.2.30 Frame Buffer Control Register 6 (Graphics 3) (GR3_FLM6) 33-45
33.2.31 Alpha Blending Control Register 1 (Graphics 3) (GR3_AB1) 33-46
33.2.32 Alpha Blending Control Register 2 (Graphics 3) (GR3_AB2) 33-47
33.2.33 Alpha Blending Control Register 3 (Graphics 3) (GR3_AB3) 33-47
33.2.34 Alpha Blending Control Register 4 (Graphics 3) (GR3_AB4) 33-48
33.2.35 Alpha Blending Control Register 5 (Graphics 3) (GR3_AB5) 33-48
33.2.36 Alpha Blending Control Register 6 (Graphics 3) (GR3_AB6) 33-49
33.2.37 Alpha Blending Control Register 7 (Graphics 3) (GR3_AB7) 33-49
33.2.38 Alpha Blending Control Register 8 (Graphics 3) (GR3_AB8) 33-50
33.2.39 Alpha Blending Control Register 9 (Graphics 3) (GR3_AB9) 33-50
33.2.40 Alpha Blending Control Register 10 (Graphics 3) (GR3_AB10) 33-51
33.2.41 Alpha Blending Control Register 11 (Graphics 3) (GR3_AB11) 33-51
33.2.42 Background Color Control Register (Graphics 3) (GR3_BASE) 33-52
33.2.43 CLUT Table and Interrupt Control Register (Graphics 3) (GR3_CLUT_INT) 33-52
33.2.44 Status Monitor Register (Graphics 3) (GR3_MON) 33-53
33.2.45 VIN Synthesizer Register Update Control Register (GR_VIN_UPDATE) 33-54
33.2.46 Alpha Blending Control Register 1 (VIN Synthesizer) (GR_VIN_AB1) 33-55
33.3 Usage Method 33-56
33.3.1 Mute Image 33-56
33.3.2 Alpha Blending in Rectangular Area 33-56
34. Video Display Controller 5 (7): Output Controller 34-1
34.1 Output Controller 34-1
34.1.1 Overview of Functions 34-1
34.1.2 Register Update Control 34-2
34.1.3 Route Selection 34-2
34.1.4 Panel Brightness Adjustment 34-3
34.1.5 Contrast Adjustment 34-3
34.1.6 Gamma Correction 34-4
34.1.7 Dither Process 34-8
34.1.8 Output Format Conversion 34-10
34.1.9 LCD TCON 34-16
34.2 Register Descriptions 34-26
34.2.1 Register Update Control Register G in Gamma Correction Block (GAM_G_UPDATE) 34-30

chanhvinh.com

34.2.2 Function Switch Register in Gamma Correction Block (GAM_SW) 34-31
34.2.3 Table Setting Register G1 to G16 in Gamma Correction Block (GAM_G_LUT1 to GAM_G_LUT16) 34-32
34.2.4 Area Setting Register G1 in Gamma Correction Block (GAM_G_AREA1) 34-34
34.2.5 Area Setting Register G2 in Gamma Correction Block (GAM_G_AREA2) 34-35
34.2.6 Area Setting Register G3 in Gamma Correction Block (GAM_G_AREA3) 34-36
34.2.7 Area Setting Register G4 in Gamma Correction Block (GAM_G_AREA4) 34-37
34.2.8 Area Setting Register G5 in Gamma Correction Block (GAM_G_AREA5) 34-38
34.2.9 Area Setting Register G6 in Gamma Correction Block (GAM_G_AREA6) 34-39
34.2.10 Area Setting Register G7 in Gamma Correction Block (GAM_G_AREA7) 34-40
34.2.11 Area Setting Register G8 in Gamma Correction Block (GAM_G_AREA8) 34-41
34.2.12 Register Update Control Register B in Gamma Correction Block (GAM_B_UPDATE) 34-42
34.2.13 Table Setting Register B1 to B16 in Gamma Correction Block (GAM_B_LUT1 to GAM_B_LUT16) 34-43
34.2.14 Area Setting Register B1 in Gamma Correction Block (GAM_B_AREA1) 34-45
34.2.15 Area Setting Register B2 in Gamma Correction Block (GAM_B_AREA2) 34-46
34.2.16 Area Setting Register B3 in Gamma Correction Block (GAM_B_AREA3) 34-47
34.2.17 Area Setting Register B4 in Gamma Correction Block (GAM_B_AREA4) 34-48
34.2.18 Area Setting Register B5 in Gamma Correction Block (GAM_B_AREA5) 34-49
34.2.19 Area Setting Register B6 in Gamma Correction Block (GAM_B_AREA6) 34-50
34.2.20 Area Setting Register B7 in Gamma Correction Block (GAM_B_AREA7) 34-51
34.2.21 Area Setting Register B8 in Gamma Correction Block (GAM_B_AREA8) 34-52
34.2.22 Register Update Control Register R in Gamma Correction Block (GAM_R_UPDATE) 34-53
34.2.23 Table Setting Register R1 to R16 in Gamma Correction Block (GAM_R_LUT1 to GAM_R_LUT16) 34-54
34.2.24 Area Setting Register R1 in Gamma Correction Block (GAM_R_AREA1) 34-56
34.2.25 Area Setting Register R2 in Gamma Correction Block (GAM_R_AREA2) 34-57
34.2.26 Area Setting Register R3 in Gamma Correction Block (GAM_R_AREA3) 34-58
34.2.27 Area Setting Register R4 in Gamma Correction Block (GAM_R_AREA4) 34-59
34.2.28 Area Setting Register R5 in Gamma Correction Block (GAM_R_AREA5) 34-60
34.2.29 Area Setting Register R6 in Gamma Correction Block (GAM_R_AREA6) 34-61
34.2.30 Area Setting Register R7 in Gamma Correction Block (GAM_R_AREA7) 34-62
34.2.31 Area Setting Register R8 in Gamma Correction Block (GAM_R_AREA8) 34-63
34.2.32 TCON Register Update Control Register (TCON_UPDATE) 34-64
34.2.33 TCON Reference Timing Setting Register (TCON_TIM) 34-64
34.2.34 TCON Vertical Timing Setting Register A1 (TCON_TIM_STVA1) 34-65
34.2.35 TCON Vertical Timing Setting Register A2 (TCON_TIM_STVA2) 34-66
34.2.36 TCON Vertical Timing Setting Register B1 (TCON_TIM_STVB1) 34-67
34.2.37 TCON Vertical Timing Setting Register B2 (TCON_TIM_STVB2) 34-68
34.2.38 TCON Horizontal Timing Setting Register STH1 (TCON_TIM_STH1) 34-69
34.2.39 TCON Horizontal Timing Setting Register STH2 (TCON_TIM_STH2) 34-70
34.2.40 TCON Horizontal Timing Setting Register STB1 (TCON_TIM_STB1) 34-71

chanhvinh.com

34.2.41 TCON Horizontal Timing Setting Register STB2 (TCON_TIM_STB2) 34-72
34.2.42 TCON Horizontal Timing Setting Register CPV1 (TCON_TIM_CPV1) 34-73
34.2.43 TCON Horizontal Timing Setting Register CPV2 (TCON_TIM_CPV2) 34-74
34.2.44 TCON Horizontal Timing Setting Register POLA1 (TCON_TIM_POLA1) 34-75
34.2.45 TCON Horizontal Timing Setting Register POLA2 (TCON_TIM_POLA2) 34-76
34.2.46 TCON Horizontal Timing Setting Register POLB1 (TCON_TIM_POLB1) 34-77
34.2.47 TCON Horizontal Timing Setting Register POLB2 (TCON_TIM_POLB2) 34-78
34.2.48 TCON Data Enable Polarity Setting Register (TCON_TIM_DE) 34-79
34.2.49 Register Update Control Register in Output Controller (OUT_UPDATE) 34-79
34.2.50 Output Interface Register (OUT_SET) 34-80
34.2.51 Brightness (DC) Correction Register 1 (OUT_BRIGHT1) 34-81
34.2.52 Brightness (DC) Correction Register 2 (OUT_BRIGHT2) 34-82
34.2.53 Contrast (Gain) Correction Register (OUT_CONTRAST) 34-82
34.2.54 Panel Dither Register (OUT_PDTHA) 34-83
34.2.55 Output Phase Control Register (OUT_CLK_PHASE) 34-84
34.3 Usage Methods 34-85
34.3.1 Gamma Correction Adjustment Method 34-85
34.3.2 Dither Usage Method 34-85
34.3.3 Output Format Adjustment Method 34-86
35. Video Display Controller 5 (8):System Controller 35-1
35.1 System Controller 35-1
35.1.1 Overview of Functions 35-1
35.1.2 Interrupt Control 35-1
35.1.3 Panel Clock Control 35-4
35.1.4 CLUT Table Read Select Signal Status Flag 35-5
35.2 Register Descriptions 35-6
35.2.1 Interrupt Control Register 1 (SYSCNT_INT1) 35-7
35.2.2 Interrupt Control Register 2 (SYSCNT_INT2) 35-9
35.2.3 Interrupt Control Register 4 (SYSCNT_INT4) 35-10
35.2.4 Interrupt Control Register 5 (SYSCNT_INT5) 35-11
35.2.5 Panel Clock Control Register (SYSCNT_PANEL_CLK) 35-12
35.2.6 CLUT Table Read Select Signal Status Register (SYSCNT_CLUT) 35-13
36. Capture Engine Unit 36-1
36.1 Features of CEU 36-1
36.2 Functional Overview of CEU 36-2
36.3 Pin Configuration of CEU 36-3
36.4 Register Descriptions of CEU 36-4
36.4.1 Capture Start Register (CAPSR) 36-5
36.4.2 Capture Control Register (CAPCR) 36-9
36.4.3 Capture Interface Control Register (CAMCR) 36-12
36.4.4 Capture Interface Cycle Register (CMCYR) 36-16

chanhvinh.com

36.4.5 Capture Interface Offset Register (CAMOR) 36-17
36.4.6 Capture Interface Width Register (CAPWR) 36-19
36.4.7 Capture Interface Input Format Register (CAIFR) 36-21
36.4.8 CEU Register Control Register (CRCNTR) 36-25
36.4.9 CEU Register Forcible Control Register (CRCMPR) 36-26
36.4.10 Capture Filter Control Register (CFLCR) 36-27
36.4.11 Capture Filter Size Clip Register (CFSZR) 36-29
36.4.12 Capture Destination Width Register (CDWDR) 36-31
36.4.13 Capture Data Address Y Register (CDAYR) 36-32
36.4.14 Capture Data Address C Register (CDACR) 36-33
36.4.15 Capture Data Bottom-Field Address Y Register (CDBYR) 36-35
36.4.16 Capture Data Bottom-Field Address C Register (CDBCR) 36-36
36.4.17 Capture Bundle Destination Size Register (CBDSR) 36-37
36.4.18 Capture Low-Pass Filter Control Register (CLFCR) 36-38
36.4.19 Firewall Operation Control Register (CFWCR) 36-39
36.4.20 Capture Data Output Control Register (CDOCR) 36-39
36.4.21 Capture Event Interrupt Enable Register (CEIER) 36-43
36.4.22 Capture Event Flag Clear Register (CETCR) 36-45
36.4.23 Capture Status Register (CSTSR) 36-49
36.4.24 Capture Data Size Register (CDSSR) 36-51
36.4.25 Capture Data Address Y Register 2 (CDAYR2) 36-52
36.4.26 Capture Data Address C Register 2 (CDACR2) 36-54
36.4.27 Capture Data Bottom-Field Address Y Register 2 (CDBYR2) 36-56
36.4.28 Capture Data Bottom-Field Address C Register 2 (CDBCR2) 36-57
36.5 Usage Notes for CEU 36-59
36.5.1 Conditions for Connection to an External Module 36-59
36.5.2 Restrictions on Input/Output Functions 36-60
36.5.3 Cooperation with Video Display Controller 5 36-60
36.5.4 Software Reset 36-60
37. SCUX 37-1
37.1 Features 37-1
37.2 Input/Output Pins 37-3
37.3 Register Descriptions 37-4
37.3.1 IPC0_n Initialization Register (IPCIR_IPC0_n) (n = 0, 1, 2, 3) 37-11
37.3.2 IPC0_n Pass Select Register (IPSLR_IPC0_n) (n = 0, 1, 2, 3) 37-11
37.3.3 OPC0_n Initialization Register (OPCIR_OPC0_n) (n = 0, 1, 2, 3) 37-12
37.3.4 OPC0_n Pass Select Register (OPSLR_OPC0_n) (n = 0, 1, 2, 3) 37-12
37.3.5 FFD0_n FIFO Download Initialization Register (FFDIR_FFD0_n) (n = 0, 1, 2, 3) 37-13
37.3.6 FFD0_n FIFO Download Audio Information Register (FDAIR_FFD0_n) (n = 0, 1, 2, 3) 37-13
37.3.7 FFD0_n FIFO Download Request Size Register (DRQSR_FFD0_n) (n = 0, 1, 2, 3) 37-14
37.3.8 FFD0_n FIFO Download Pass Register (FFDPR_FFD0_n) (n = 0, 1, 2, 3) 37-15

chanhvinh.com

37.3.9 FFD0_n FIFO Download Boot Register (FFDBR_FFD0_n) (n = 0, 1, 2, 3) 37-15
37.3.10 FFD0_n FIFO Download Event Mask Register (DEVMR_FFD0_n) (n = 0, 1, 2, 3) 37-16
37.3.11 FFD0_n FIFO Download Event Clear Register (DEVCR_FFD0_n) (n = 0, 1, 2, 3) 37-17
37.3.12 FFU0_n FIFO Upload Initialization Register (FFUIR_FFU0_n) (n = 0, 1, 2, 3) 37-18
37.3.13 FFU0_n FIFO Upload Audio Information Register (FUAIR_FFU0_n) (n = 0, 1, 2, 3) 37-18
37.3.14 FFU0_n FIFO Upload Request Size Register (URQSR_FFU0_n) (n = 0, 1, 2, 3) 37-19
37.3.15 FFU0_n FIFO Upload Pass Register (FFUPR_FFU0_n) (n = 0, 1, 2, 3) 37-20
37.3.16 FFU0_n FIFO Upload Event Mask Register (UEVMR_FFU0_n) (n = 0, 1, 2, 3) 37-21
37.3.17 FFU0_n FIFO Upload Event Clear Register (UEVCR_FFU0_n) (n = 0, 1, 2, 3) 37-22
37.3.18 2SRC0_m Initialization Register p (SRCIRp_2SRC0_m) (m = 0, 1; p = 0, 1) 37-23
37.3.19 2SRC0_m Audio Information Register p (SADIRp_2SRC0_m) (m = 0, 1; p = 0, 1) 37-24
37.3.20 2SRC0_m Bypass Register p (SRCBRp_2SRC0_m) (m = 0, 1; p = 0, 1) 37-25
37.3.21 2SRC0_m IFS Control Register p (IFSCRp_2SRC0_m) (m = 0, 1; p = 0, 1) 37-25
37.3.22 2SRC0_m IFS Value Setting Register p (IFSVRp_2SRC0_m) (m = 0, 1; p = 0, 1) 37-26
37.3.23 2SRC0_m Control Register p (SRCCRp_2SRC0_m) (m = 0, 1; p = 0, 1) 37-31
37.3.24 2SRC0_m Minimum FS Setting Register p (MNFSRp_2SRC0_m) (m = 0, 1; p = 0, 1) 37-32
37.3.25 2SRC0_m Buffer Size Setting Register p (BFSSRp_2SRC0_m) (m = 0, 1; p = 0, 1) 37-32
37.3.26 2SRC0_m SCU2 Status Register p (SC2SRp_2SRC0_m) (m = 0, 1; p = 0, 1) 37-33
37.3.27 2SRC0_m Wait Time Setting Register p (WATSRp_2SRC0_m) (m = 0, 1; p = 0, 1) 37-33
37.3.28 2SRC0_m Event Mask Register p (SEVMRp_2SRC0_m) (m = 0, 1; p = 0, 1) 37-34
37.3.29 2SRC0_m Event Clear Register p (SEVCRp_2SRC0_m) (m = 0, 1; p = 0, 1) 37-35
37.3.30 2SRC0_m Initialization Register RIF (SRCIRR_2SRC0_m) (m = 0, 1) 37-36
37.3.31 DVU0_n Initialization Register (DVUIR_DVU0_n) (n = 0, 1, 2, 3) 37-36
37.3.32 DVU0_n Audio Information Register (VADIR_DVU0_n) (n = 0, 1, 2, 3) 37-37
37.3.33 DVU0_n Bypass Register (DVUBR_DVU0_n) (n = 0, 1, 2, 3) 37-38
37.3.34 DVU0_n Control Register (DVUCR_DVU0_n) (n = 0, 1, 2, 3) 37-38
37.3.35 DVU0_n Zero Cross Mute Control Register (ZCMCR_DVU0_n) (n = 0, 1, 2, 3) 37-39
37.3.36 DVU0_n Volume Ramp Control Register (VRCTR_DVU0_n) (n = 0, 1, 2, 3) 37-40
37.3.37 DVU0_n Volume Ramp Period Register (VRPDR_DVU0_n) (n = 0, 1, 2, 3) 37-41
37.3.38 DVU0_n Volume Ramp Decibel Register (VRDBR_DVU0_n) (n = 0, 1, 2, 3) 37-42
37.3.39 DVU0_n Volume Ramp Wait Time Register (VRWTR_DVU0_n) (n = 0, 1, 2, 3) 37-43
37.3.40 DVU0_n Volume Value Setting 0 Register (VOL0R_DVU0_n) (n = 0, 1, 2, 3) 37-43
37.3.41 DVU0_n Volume Value Setting 1 Register (VOL1R_DVU0_n) (n = 0, 1, 2, 3) 37-44
37.3.42 DVU0_n Volume Value Setting 2 Register (VOL2R_DVU0_n) (n = 0, 1, 2, 3) 37-45
37.3.43 DVU0_n Volume Value Setting 3 Register (VOL3R_DVU0_n) (n = 0, 1, 2, 3) 37-46
37.3.44 DVU0_n Volume Value Setting 4 Register (VOL4R_DVU0_n) (n = 0, 1, 2, 3) 37-47
37.3.45 DVU0_n Volume Value Setting 5 Register (VOL5R_DVU0_n) (n = 0, 1, 2, 3) 37-48
37.3.46 DVU0_n Volume Value Setting 6 Register (VOL6R_DVU0_n) (n = 0, 1, 2, 3) 37-49
37.3.47 DVU0_n Volume Value Setting 7 Register (VOL7R_DVU0_n) (n = 0, 1, 2, 3) 37-50
37.3.48 DVU0_n Enable Register (DVUER_DVU0_n) (n = 0, 1, 2, 3) 37-50
37.3.49 DVU0_n Status Register (DVUSR_DVU0_n) (n = 0, 1, 2, 3) 37-51

chanhvinh.com

37.3.50 DVU0_n Event Mask Register (VEVMR_DVU0_n) (n = 0, 1, 2, 3) 37-52
37.3.51 DVU0_n Event Clear Register (VEVCR_DVU0_n) (n = 0, 1, 2, 3) 37-54
37.3.52 MIX0_0 Initialization Register (MIXIR_MIX0_0) 37-55
37.3.53 MIX0_0 Audio Information Register (MADIR_MIX0_0) 37-56
37.3.54 MIX0_0 Bypass Register (MIXBR_MIX0_0) 37-57
37.3.55 MIX0_0 Mode Register (MIXMR_MIX0_0) 37-57
37.3.56 MIX0_0 Volume Period Register (MVPDR_MIX0_0) 37-58
37.3.57 MIX0_0 Decibel A Register (MDBAR_MIX0_0) 37-59
37.3.58 MIX0_0 Decibel B Register (MDBBR_MIX0_0) 37-60
37.3.59 MIX0_0 Decibel C Register (MDBCR_MIX0_0) 37-61
37.3.60 MIX0_0 Decibel D Register (MDBDR_MIX0_0) 37-62
37.3.61 MIX0_0 Decibel Enable Register (MDBER_MIX0_0) 37-63
37.3.62 MIX0_0 Status Register (MIXSR_MIX0_0) 37-63
37.3.63 Software Reset Register (SWRSR_CIM) 37-64
37.3.64 DMA Control Register (DMACR_CIM) 37-65
37.3.65 DMA Transfer Register for FFD0_n (DMATDn_CIM) (n = 0, 1, 2, 3) 37-66
37.3.66 DMA Transfer Register for FFU0_n (DMATUn_CIM) (n = 0, 1, 2, 3) 37-67
37.3.67 SSI Route Select Register (SSIRSEL_CIM) 37-68
37.3.68 FFD0_n Timing Select Register (FDTSELn_CIM) (n = 0, 1, 2, 3) 37-70
37.3.69 FFU0_n Timing Select Register (FDTSELn_CIM) (n = 0, 1, 2, 3) 37-72
37.3.70 SSI Pin Mode Register (SSIPMD_CIM) 37-74
37.3.71 SSI Control Register (SSICTRL_CIM) 37-76
37.3.72 SRCn Route Select Register (SRCRSELn_CIM) (n = 0, 1, 2, 3) 37-78
37.3.73 MIX Route Select Register (MIXRSEL_CIM) 37-82
37.4 Operation 37-86
37.4.1 Initial Setting Procedure 37-86
37.4.2 Transfer Start Procedure and Stop Procedure 37-86
37.4.3 Data Rearrangement for Each Channel 37-91
37.4.4 Pin Connection Specifications of SSIF 37-94
37.4.5 Data Transfer Route 37-95
37.4.6 Input Timing Signal and Output Timing Signal 37-101
37.4.7 2SRC (SRC) Block 37-104
37.4.8 DVU Block 37-106
37.4.9 MIX Block 37-110
37.5 Usage Note 37-114
37.5.1 Software Reset 37-114
38. SD Host Interface 38-1
38.1 Overview 38-1
38.1.1 Features 38-1
38.1.2 Block Diagram 38-1
38.1.3 Input/Output Pins 38-2

chanhvinh.com

38.2 Register Descriptions 38-3
38.2.1 Command Type Register (SD_CMD) 38-5
38.2.2 SD Command Argument Registers (SD_ARG) 38-6
38.2.3 Data STOP Register (SD_STOP) 38-7
38.2.4 Block Count Register (SD_SECCNT) 38-8
38.2.5 SD Card Response Registers (SD_RSP) 38-8
38.2.6 SD Card Interrupt Flag Register 1 (SD_INFO1) 38-10
38.2.7 SD Card Interrupt Flag Register 2 (SD_INFO2) 38-12
38.2.8 SD_INFO1 Interrupt Mask Register (SD_INFO1_MASK) 38-14
38.2.9 SD_INFO2 Interrupt Mask Register (SD_INFO2_MASK) 38-15
38.2.10 SD Clock Control Register (SD_CLK_CTRL) 38-16
38.2.11 Transfer Data Length Register (SD_SIZE) 38-16
38.2.12 SD Card Access Control Option Register (SD_OPTION) 38-17
38.2.13 SD Error Status Register 1 (SD_ERR_STS1) 38-18
38.2.14 SD Error Status Register 2 (SD_ERR_STS2) 38-19
38.2.15 SD Buffer Read/Write Register (SD_BUF0) 38-19
38.2.16 SDIO Mode Control Register (SDIO_MODE) 38-20
38.2.17 SDIO Interrupt Flag Register (SDIO_INFO1) 38-22
38.2.18 SDIO_INFO1 Interrupt Mask Register (SDIO_INFO1_MASK) 38-23
38.2.19 DMA Mode Enable Register (CC_EXT_MODE) 38-23
38.2.20 Software Reset Register (SOFT_RST) 38-24
38.2.21 Version Register (VERSION) 38-24
38.2.22 Swap Control Register (EXT_SWAP) 38-25
38.3 Operation 38-26
38.3.1 SD I/F 38-26
38.3.2 Card Detect/Write Protect 38-28
38.3.3 Interrupt Request and DMA Transfer Request 38-30
38.3.4 Communications Errors and Timeouts 38-32
38.4 Usage Example 38-34
38.4.1 Command without Data Transfer 38-34
38.4.2 Single Block Read 38-36
38.4.3 Single Block Write 38-38
38.4.4 Multiple Block Read 38-40
38.4.5 Multiple Block Write 38-42
38.4.6 IO_RW_DIRECT Command (CMD52) 38-44
38.4.7 IO_RW_EXTENDED Command (CMD53/Multiple Block Read) 38-45
38.4.8 IO_RW_EXTENDED Command (CMD53/Multiple Block Write) 38-47
38.4.9 DMA Transfer 38-49
38.4.10 Example of SD_CMD Register Setting 38-51
38.5 Usage Note 38-52

chanhvinh.com

39. MMC Host Interface 39-1
39.1 Features 39-1
39.2 Input/Output Pins 39-1
39.3 Register Descriptions 39-2
39.3.1 Command Setting Register (CE_CMD_SET) 39-3
39.3.2 Argument Register (CE_ARG) 39-5
39.3.3 Argument Register for Automatically-Issued CMD12 (CE_ARG_CMD12) 39-5
39.3.4 Command Control Register (CE_CMD_CTRL) 39-6
39.3.5 Transfer Block Setting Register (CE_BLOCK_SET) 39-6
39.3.6 Clock Control Register (CE_CLK_CTRL) 39-7
39.3.7 Buffer Access Configuration Register (CE_BUF_ACC) 39-8
39.3.8 Response Registers 3 to 0 (CE_RESP3 to CE_RESP0) 39-9
39.3.9 Response Register for Automatically-Issued CMD12 (CE_RESP_CMD12) 39-10
39.3.10 Data Register (CE_DATA) 39-10
39.3.11 Interrupt Flag Register (CE_INT) 39-11
39.3.12 Interrupt Enable Register (CE_INT_EN) 39-15
39.3.13 Status Register 1 (CE_HOST _STS1) 39-17
39.3.14 Status Register 2 (CE_HOST _STS2) 39-18
39.3.15 DMA Mode Setting Register (CE_DMA_MODE) 39-19
39.3.16 Card Detection/Port Control Register (CE_DETECT) 39-20
39.3.17 Special Mode Setting Register (CE_ADD_MODE) 39-21
39.3.18 Version Register (CE_VERSION) 39-21
39.4 Interrupt Requests 39-22
39.5 DMA Specifications 39-23
39.5.1 DMA for Buffer Writing 39-23
39.5.2 DMA for Buffer Reading 39-23
39.6 Operation 39-24
39.6.1 Command/Response Formats 39-24
39.6.2 Data Block Format 39-25
39.6.3 Buffer Structure and Buffer Accesses 39-26
39.6.4 Automatic CMD12 Issuance 39-27
39.6.5 High Priority Interrupt (HPI) 39-28
39.6.6 Background Operation 39-28
39.6.7 Operation in the Case of Error/Timeout 39-28
39.7 Examples of Setting 39-29
39.7.1 Legends 39-29
39.7.2 Command Transmission 39-29
39.7.3 Command Transmission → Response Reception 39-30
39.7.4 Command Transmission → Response Reception (with Response Busy) 39-31
39.7.5 Single-Block Read 39-33
39.7.6 Multi-Block Read 39-34

chanhvinh.com

39.7.7 Multi-Block Read (with Automatic CMD12 Issuance) 39-35
39.7.8 Single-Block Write 39-36
39.7.9 Multi-Block Write 39-37
39.7.10 Multi-Block Write (with Automatic CMD12 Issuance) 39-38
39.7.11 Forcible Termination 39-39
39.7.12 Setting Values of CE_CMD_SET 39-40
39.8 Usage Note 39-42
39.8.1 Card Detection 39-42
39.8.2 Multi-Block Transfer 39-42
39.8.3 Software Reset 39-42
40. On-Chip RAM 40-1
40.1 Features 40-1
40.2 Usage Notes 40-3
40.2.1 Page Conflict 40-3
40.2.2 Data Retention 40-3
41. Ports 41-1
41.1 Features 41-1
41.1.1 Port group 41-1
41.1.2 Port group index n 41-1
41.1.3 Base address 41-1
41.2 Functional Overview 41-2
41.2.1 Mode of Pin Functions 41-2
41.2.2 Pin Data Input/Output 41-3
41.3 Register Description 41-4
41.3.1 Port Register (Pn) 41-4
41.3.2 Port Set and Reset Register (PSRn) 41-5
41.3.3 Port Pin Read Register (PPRn/JPPR0) 41-5
41.3.4 Port Mode Register (PMn) 41-6
41.3.5 Port Mode Control Register (PMCn/JPMC0) 41-6
41.3.6 Port Function Control Register (PFCn) 41-7
41.3.7 Port Function Control Expansion Register (PFCEn) 41-8
41.3.8 Port NOT Register (PNOTn) 41-8
41.3.9 Port Mode Set and Reset Register (PMSRn) 41-9
41.3.10 Port Mode Control Set and Reset Register (PMCSRn/JPMCSR0) 41-10
41.3.11 Port Function Control Additional Expansion Register (PFCAEn) 41-10
41.3.12 Port Input Buffer Control Register (PIBCn/JPIBC0) 41-11
41.3.13 Port Bidirection Control Register (PBDCn) 41-11
41.3.14 Port IP Control Register (PIPCn) 41-12
41.3.15 Serial Sound Interface Noise Canceler Control Register (SNCR) 41-13
41.4 Port Function 41-14
41.5 JTAG Port 0 (JP0) 41-14

chanhvinh.com

41.6 Port 0 (P0) 41-15
41.7 Port 1 (P1) 41-16
41.8 Port 2 (P2) 41-17
41.9 Port 3 (P3) 41-18
41.10 Port 4 (P4) 41-19
41.11 Port 5 (P5) 41-20
41.12 Port 6 (P6) 41-21
41.13 Port 7 (P7) 41-23
41.14 Port 8 (P8) 41-24
41.15 Port 9 (P9) 41-25
41.16 Port Control Logical Diagram 41-26
41.17 Flowchart Examples of Port Setting 41-27
42. Power-Down Modes 42-1
42.1 Features 42-1
42.1.1 States of Processing and Power-Down Modes 42-1
42.2 Register Descriptions 42-3
42.2.1 Standby Control Register 1 (STBCR1) 42-4
42.2.2 Standby Control Register 2 (STBCR2) 42-5
42.2.3 Standby Control Register 3 (STBCR3) 42-6
42.2.4 Standby Control Register 4 (STBCR4) 42-7
42.2.5 Standby Control Register 5 (STBCR5) 42-8
42.2.6 Standby Control Register 6 (STBCR6) 42-9
42.2.7 Standby Control Register 7 (STBCR7) 42-10
42.2.8 Standby Control Register 8 (STBCR8) 42-11
42.2.9 Standby Control Register 9 (STBCR9) 42-12
42.2.10 Standby Control Register 10 (STBCR10) 42-13
42.2.11 Standby Control Register 11 (STBCR11) 42-14
42.2.12 Standby Control Register 12 (STBCR12) 42-15
42.2.13 Software Reset Control Register 1 (SWRSTCR1) 42-16
42.2.14 Software Reset Control Register 2 (SWRSTCR2) 42-17
42.2.15 System Control Register 1 (SYSCR1) 42-18
42.2.16 System Control Register 2 (SYSCR2) 42-19
42.2.17 System Control Register 3 (SYSCR3) 42-20
42.2.18 CPU Status Register (CPUSTS) 42-21
42.2.19 Standby Request Register 1 (STBREQ1) 42-22
42.2.20 Standby Request Register 2 (STBREQ2) 42-23
42.2.21 Standby Acknowledge Register 1 (STBACK1) 42-24
42.2.22 Standby Acknowledge Register 2 (STBACK2) 42-25
42.2.23 On-Chip Data-Retention RAM Area Setting Register (RRAMKP) 42-26
42.2.24 Deep Standby Control Register (DSCTR) 42-27
42.2.25 Deep Standby Cancel Source Select Register (DSSSR) 42-28

chanhvinh.com

42.2.26 Deep Standby Cancel Edge Select Register (DSESR) 42-30
42.2.27 Deep Standby Cancel Source Flag Register (DSFR) 42-31
42.2.28 XTAL Crystal Oscillator Gain Control Register (XTALCTR) 42-32
42.3 Operation 42-33
42.3.1 Sleep Mode 42-33
42.3.2 Software Standby Mode 42-34
42.3.3 Software Standby Mode Application Example 42-36
42.3.4 Deep Standby Mode 42-36
42.3.5 Module Standby Function 42-41
42.3.6 Software Reset 42-42
42.3.7 Adjustment of XTAL Crystal Oscillator Gain 42-42
42.4 Usage Notes 42-44
42.4.1 Usage Notes on Setting Registers 42-44
42.4.2 Usage Notes when the Realtime Clock is not Used 42-44
42.4.3 Usage Notes Applying when the USB_X1 Pin is not to be Used 42-44
42.4.4 Notes on Using IRQ Pins as Triggers for Release from Standby when Software Standby is in Use 42-44
43. Debugger Interface 43-1
43.1 Features 43-1
43.2 Input/Output Pins 43-5
43.3 Registers for Boundary-Scan TAP Controller 43-6
43.3.1 Bypass Register (BSBPR) 43-6
43.3.2 Instruction Register (BSIR) 43-6
43.3.3 Boundary Scan Register (SDBSR) 43-7
43.3.4 ID Register (BSID) 43-15
43.4 ICE Registers 43-16
43.4.1 Mode Reset Control Register (ICEREGMDRSTCTL) 43-16
43.4.2 JTAG Trace Select Register (ICEREGJTTRCSEL) 43-17
43.4.3 Clock Power Control Register (ICEREGCLKPWRCTRL) 43-18
43.4.4 Lock Access Register (ICEREGLOCKACCESS) 43-18
43.5 Operation 43-19
43.5.1 TAP Controller 43-19
43.5.2 Reset Signal Setting 43-19
43.6 Boundary Scan 43-20
43.6.1 Supported Instructions 43-20
43.6.2 Points for Attention 43-21
43.7 Usage Notes 43-21
44. JPEG Codec Unit 44-1
44.1 Features 44-1
44.2 Register Descriptions 44-3
44.2.1 JPEG Code Mode Register (JCMOD) 44-4

chanhvinh.com

44.2.2 JPEG Code Command Register (JCCMD) 44-5
44.2.3 JPEG Code Quantization Table Number Register (JCQTN) 44-6
44.2.4 JPEG Code Huffman Table Number Register (JCHTN) 44-7
44.2.5 JPEG Code DRI Upper Register (JCDRIU) 44-7
44.2.6 JPEG Code DRI Lower Register (JCDRID) 44-8
44.2.7 JPEG Code Vertical Size Upper Register (JCVSZU) 44-8
44.2.8 JPEG Code Vertical Size Lower Register (JCVSZD) 44-8
44.2.9 JPEG Code Horizontal Size Upper Register (JCHSZU) 44-9
44.2.10 JPEG Coded Horizontal Size Lower Register (JCHSZD) 44-9
44.2.11 JPEG Code Data Count Upper Register (JCDTCU) 44-9
44.2.12 JPEG Code Data Count Middle Register (JCDTCM) 44-10
44.2.13 JPEG Code Data Count Lower Register (JCDTCD) 44-10
44.2.14 JPEG Interrupt Enable Register 0 (JINTE0) 44-11
44.2.15 JPEG Interrupt Status Register 0 (JINTS0) 44-12
44.2.16 JPEG Code Decode Error Register (JCDERR) 44-12
44.2.17 JPEG Code Reset Register (JCRST) 44-13
44.2.18 JPEG Interface Compression Control Register (JIFECNT) 44-14
44.2.19 JPEG Interface Compression Source Address Register (JIFESA) 44-15
44.2.20 JPEG Interface Compression Line Offset Register (JIFESOFST) 44-16
44.2.21 JPEG Interface Compression Destination Address Register (JIFEDA) 44-16
44.2.22 JPEG Interface Compression Source Line Count Register (JIFESLC) 44-17
44.2.23 JPEG Interface Compression Destination Count Register (JIFEDDC) 44-17
44.2.24 JPEG Interface Decompression Control Register (JIFDCNT) 44-18
44.2.25 JPEG Interface Decompression Source Address Register (JIFDSA) 44-20
44.2.26 JPEG Interface Decompression Line Offset Register (JIFDDOFST) 44-20
44.2.27 JPEG Interface Decompression Destination Address Register (JIFDDA) 44-21
44.2.28 JPEG Interface Decompression Source Data Count Register (JIFDSDC) 44-22
44.2.29 JPEG Interface Decompression Destination Line Count Register (JIFDDLC) 44-22
44.2.30 JPEG Interface Decompression α Set Register (JIFDADT) 44-23
44.2.31 JPEG Interrupt Enable Register 1 (JINTE1) 44-24
44.2.32 JPEG Interrupt Status Register 1 (JINTS1) 44-25
44.2.33 JPEG Input Image Data CbCr Range Setting Register (JIFESVSZ) 44-26
44.2.34 JPEG Output Image Data CbCr Range Setting Register (JIFESHSZ) 44-26
44.3 Operation 44-27
44.3.1 Compression 44-27
44.3.2 Decompression 44-33
44.3.3 Output Pixel Format in Decompression 44-39
44.3.4 Storing Image Data 44-43
44.4 Interrupts 44-44
44.4.1 Compression/Decompression Process Interrupt Request (JEDI) 44-44
44.4.2 Data Transfer Interrupt Request (JDTI) 44-44

chanhvinh.com

44.5 Bus Reset Processing 44-46
45. EthernetAVB 45-1
45.1 Overview 45-1
45.1.1 Specifications (Functions) 45-1
45.1.2 Block Diagram 45-2
45.1.3 I/O Pins 45-3
45.2 Register Descriptions 45-4
45.2.1 AVB-DMAC Mode Register (CCC) 45-6
45.2.2 Descriptor Base Address Table Register (DBAT) 45-9
45.2.3 Descriptor Base Address Load Request Register (DLR) 45-10
45.2.4 AVB-DMAC Status Register (CSR) 45-13
45.2.5 Current Descriptor Address Register q (CDARq) (q = 0 to 21) 45-16
45.2.6 Error Status Register (ESR) 45-17
45.2.7 Receive Configuration Register (RCR) 45-19
45.2.8 Receive Queue Configuration Register i (RQCi) (i = 0 to 4) 45-22
45.2.9 Receive Padding Configuration Register (RPC) 45-24
45.2.10 Unread Frame Counter Stop Level Configuration Register (UFCS) 45-26
45.2.11 Unread Frame Counter Register i (UFCVi) (i = 0 to 4) 45-27
45.2.12 Unread Frame Counter Decrement Register i (UFCDi) (i = 0 to 4) 45-29
45.2.13 Separation Filter Offset Register (SFO) 45-30
45.2.14 Separation Filter Pattern Register i (SFPi) (i = 0 to 31) 45-31
45.2.15 Separation Filter Mask Register i (SFMi) (i = 0 or 1) 45-32
45.2.16 Transmit Configuration Register (TGC) 45-33
45.2.17 Transmit Configuration Control Register (TCCR) 45-35
45.2.18 Transmit Status Register (TSR) 45-37
45.2.19 Time Stamp FIFO Access Register 0 (TFA0) 45-39
45.2.20 Time Stamp FIFO Access Register 1 (TFA1) 45-40
45.2.21 Time Stamp FIFO Access Register 2 (TFA2) 45-41
45.2.22 CBS Increment Value Register c (CIVRc) (c = 0 or 1) 45-42
45.2.23 CBS Decrement Value Register c (CDVRc) (c = 0 or 1) 45-43
45.2.24 CBS Upper Limit Register c (CULc) (c = 0 or 1) 45-44
45.2.25 CBS Lower Limit Register c (CLLc) (c = 0 or 1) 45-45
45.2.26 Descriptor Interrupt Control Register (DIC) 45-46
45.2.27 Descriptor Interrupt Status Register (DIS) 45-48
45.2.28 Error Interrupt Control Register (EIC) 45-50
45.2.29 Error Interrupt Status Register (EIS) 45-52
45.2.30 Receive Interrupt Control Register 0 (RIC0) 45-55
45.2.31 Receive Interrupt Status Register 0 (RIS0) 45-57
45.2.32 Receive Interrupt Control Register 1 (RIC1) 45-59
45.2.33 Receive Interrupt Status Register 1 (RIS1) 45-60
45.2.34 Receive Interrupt Control Register 2 (RIC2) 45-61

chanhvinh.com

45.2.35 Receive Interrupt Status Register 2 (RIS2) 45-63
45.2.36 Transmit Interrupt Control Register (TIC) 45-66
45.2.37 Transmit Interrupt Status Register (TIS) 45-67
45.2.38 Interrupt Summary Status Register (ISS) 45-69
45.2.39 gPTP Configuration Control Register (GCCR) 45-72
45.2.40 gPTP Maximum Transit Time Configuration Register (GMTT) 45-75
45.2.41 gPTP Presentation Time Comparison Register (GPTC) 45-76
45.2.42 gPTP Timer Increment Configuration Register (GTI) 45-77
45.2.43 gPTP Timer Offset Configuration Register i (GTOi) (i = 0 to 2) 45-78
45.2.44 gPTP Interrupt Control Register (GIC) 45-79
45.2.45 gPTP Interrupt Status Register (GIS) 45-80
45.2.46 gPTP Presentation Time Capture Register (GCPT) 45-82
45.2.47 gPTP Timer Capture Register i (GCTi) (i = 0 to 2) 45-83
45.2.48 gPTP Capture Event Control Register (GCEC) 45-84
45.2.49 E-MAC Mode Register (ECMR) 45-87
45.2.50 Receive Frame Length Register (RFLR) 45-90
45.2.51 E-MAC Status Register (ECSR) 45-91
45.2.52 E-MAC Interrupt Permission Register (ECSIPR) 45-92
45.2.53 PHY Interface Register (PIR) 45-93
45.2.54 Automatic PAUSE Frame Register (APR) 45-94
45.2.55 Manual PAUSE Frame Register (MPR) 45-95
45.2.56 PAUSE Frame Transmit Counter (PFTCR) 45-96
45.2.57 PAUSE Frame Receive Counter (PFRCR) 45-97
45.2.58 Automatic PAUSE Frame Retransmission Count Register (TPAUSER) 45-98
45.2.59 MAC Address High Register (MAHR) 45-99
45.2.60 MAC Address Low Register (MALR) 45-100
45.2.61 CRC Error Frame Receive Counter Register (CEFCR) 45-101
45.2.62 Frame Receive Error Counter Register (FRECR) 45-102
45.2.63 Too-Short Frame Receive Counter Register (TSFRCR) 45-103
45.2.64 Too-Long Frame Receive Counter Register (TLFRCR) 45-104
45.2.65 Residual-Bit Frame Receive Counter Register (RFCR) 45-105
45.2.66 Multicast Address Frame Receive Counter Register (MAFCR) 45-106
45.3 Operation 45-107
45.3.1 AVB-DMAC Operating Modes 45-108
45.3.2 Common Control for Transmission and Reception 45-113
45.3.3 Descriptors 45-122
45.3.4 Control in Reception 45-136
45.3.5 Transmission Control 45-151
45.3.6 CBS (Credit-Based Shaping) 45-167
45.3.7 IEEE802.1: gPTP 45-176
45.3.8 Support for IEEE 1722 45-179

chanhvinh.com

45.3.9 Flow Control 45-180
45.3.10 Interrupts 45-182
45.3.11 Flows of Operations 45-184
45.3.12 Connection to PHY-LSI 45-195
45.3.13 Usage Notes 45-199
46. List of Registers 46-1
46.1 Register Addresses 46-1
46.2 Register Bits 46-68
46.3 Register States 46-230
47. Electrical Characteristics 47-1
47.1 Absolute Maximum Ratings 47-1
47.2 Power-On/Power-Off Sequence 47-1
47.3 DC Characteristics 47-2
47.4 AC Characteristics 47-7
47.4.1 Clock Timing 47-7
47.4.2 Control Signal Timing 47-11
47.4.3 Bus Timing 47-13
47.4.4 Direct Memory Access Controller Timing 47-39
47.4.5 Multi-Function Timer Pulse Unit 2 Timing 47-40
47.4.6 Watchdog Timer Timing 47-41
47.4.7 Serial Communication Interface with FIFO Timing 47-42
47.4.8 Serial Communication Interface Timing 47-43
47.4.9 Renesas Serial Peripheral Interface Timing 47-44
47.4.10 SPI Multi I/O Bus Controller Timing 47-47
47.4.11 I²C Bus Interface Timing 47-50
47.4.12 Serial Sound Interface Timing 47-51
47.4.13 Media Local Bus Timing 47-54
47.4.14 CAN Interface Timing 47-55
47.4.15 Ethernet Controller and EthernetAVB Timing 47-56
47.4.16 A/D Converter Timing 47-59
47.4.17 USB 2.0 Host/Function Module Timing 47-59
47.4.18 Video Display Controller 5 Timing 47-62
47.4.19 Capture Engine Unit Module Signal Timing 47-64
47.4.20 SD Host Interface Timing 47-66
47.4.21 MMC Host Interface Timing 47-67
47.4.22 General Purpose I/O Ports Timing 47-68
47.4.23 Debugger Interface Timing 47-68
47.4.24 AC Characteristics Measurement Conditions 47-70
47.5 A/D Converter Characteristics 47-71
48. States and Handling of Pins 48-1
48.1 Pin States 48-1

chanhvinh.com

48.2 Treatment of Unused Pins 48-6
48.3 Handling of Pins in Deep Standby Mode 48-7
48.4 Recommended Combination of Bypass Capacitor 48-8
Appendix Appendix-1
A. Package Dimensions Appendix-1
Revision History ..... Revision History-1

chanhvinh.com

1. Overview

1.1 Features of This LSI

This LSI is a single-chip microcontroller that includes an Arm Cortex®-A9 processor along with the integrated peripheral functions required to configure a system.

This LSI includes a 32-Kbyte L1 instruction cache, a 32-Kbyte L1 data cache, and a 128-Kbyte L2 cache. This LSI also includes on-chip peripheral functions necessary for system configuration, such as a 3-Mbyte (RZ/A1L and RZ/A1LU) or 2-Mbyte (RZ/A1LC) large-capacity RAM (128 Kbytes are shared by the data-retention RAM), data-retention RAM, multi-function timer pulse unit 2, OS timer, realtime clock, serial communication interface with FIFO, serial communication interface, I^2C bus interface, serial sound interface, media local bus (RZ/A1L only), SCUX, CAN interface, IEBus™* controller (RZ/A1L only), Renesas SPDIF interface, Renesas serial peripheral interface, SPI multi I/O bus controller, CD-ROM decoder (RZ/A1L only), A/D converter, LIN interface (RZ/A1L only), Ethernet controller, EthernetAVB (RZ/A1LU only), USB 2.0 host/function, video display controller 5, JPEG codec unit (RZ/A1LU only), capture engine unit, SD host interface, MMC host interface, interrupt controller modules, and general I/O ports.

The features of this LSI are listed in Table 1.1.

Note: * IEBus (Inter Equipment Bus) is a trademark of Renesas Electronics Corporation.

chanhvinh.com

Table 1.1 Features of RZ/A1L, RZ/A1LU, and RZ/A1LC

Items	Specification
CPU	• Arm Cortex-A9 processor • Maximum operating frequency: 400 MHz • Instruction cache size: 32 Kbytes • Data cache size: 32 Kbytes (write-back algorithm) • TLB entries: 128 entries • Jazelle® architecture extension: Full implementation • Media processing engine with NEON™ technology
L2 cache memory	• Arm CoreLink™ Level 2 Cache Controller L2C-310 • Operating frequency: 133 MHz • Cache size: 128 Kbytes
Interrupt controller	• Arm PrimeCell® Generic Interrupt Controller (PL390) • External interrupt pins (NMI, IRQ7 to IRQ0, and TINT121 to TINT0) • On-chip peripheral interrupts: Priority level set for each module • 32 priority levels available
Bus state controller	• Address space divided into six areas (0 to 5), each a maximum of 64 Mbytes • The following features settable for each area independently —Bus size (8, 16, or 32 bits): Available sizes depend on the area. —Number of access wait cycles (different wait cycles can be specified for read and write access cycles in some areas) —Idle wait cycle insertion (between the same area access cycles or different area access cycles) —Specifying the memory to be connected to each area enables direct connection to SRAM, SRAM with byte selection, SDRAM, and burst ROM (clocked synchronous or asynchronous). The address/data multiplexed I/O (MPX) interface is also available. • Outputs a chip select signal ($\overline{CS0}$ to $\overline{CS5}$) according to the target area ($\overline{CS}$ assert or negate timing can be selected by software) • SDRAM refresh Auto refresh or self refresh mode selectable • SDRAM burst access
Direct memory access controller	• Sixteen channels; external requests are available for one of them. • Can be activated by on-chip peripheral modules. • A specific DMA transfer interval can be specified to adjust the bus occupancy. • Link mode (DMA transfer under descriptor control) supported • Transfer information can be automatically reloaded.
Clock pulse generator	• Clock mode: Input clock can be selected from external input (EXTAL or USB_X1) or crystal resonator. • Input clock can be multiplied by 32 (max.) by the internal PLL circuit. • Peak values of EMI noise can be reduced by the on-chip SSCG circuit. • Five types of clocks generated: —CPU clock (Iφ): Maximum 400.00 MHz —Internal bus clock (Bφ): Maximum 133.33 MHz —Peripheral clock 1 (P1φ): Maximum 66.67 MHz —Peripheral clock 0 (P0φ): Maximum 33.33 MHz
Watchdog timer	• On-chip one-channel watchdog timer • A counter overflow can reset the LSI.
Power-down modes	• Four power-down modes provided to reduce the power consumption in this LSI —Sleep mode —Software standby mode —Deep standby mode —Module standby mode

chanhvinh.com

Items	Specification
Multi-function timer pulse unit 2	• Maximum 16 lines of pulse inputs/outputs based on five channels of 16-bit timers • 18 output compare and input capture registers • Input capture function • Pulse output modes Toggle, PWM, complementary PWM, and reset-synchronized PWM modes • Synchronization of multiple counters • Complementary PWM output mode —Non-overlapping waveforms output for 3-phase inverter control —Automatic dead time setting —0% to 100% PWM duty value specifiable —A/D converter start request delaying function —Interrupt skipping at crest or trough • Reset-synchronized PWM mode Three-phase PWM waveforms in positive and negative phases can be output with a required duty value. • Phase counting mode Two-phase encoder pulse counting available
OS timer	• Two-channel 32-bit counters • Two operating modes: —Interval timer mode —Free-running comparison mode • DMA transfer request or interrupt request can be issued when a compare match occurs.
Realtime clock	• Internal clock, calendar function, alarm function • Interrupts can be generated at intervals of 1/64 s by the 32.768-kHz on-chip crystal oscillator.
Serial communication interface with FIFO	• Five channels • Clock synchronous mode or asynchronous mode selectable • Simultaneous transmission and reception (full-duplex communication) supported • Dedicated baud rate generator • Separate 16-byte FIFO registers for transmission and reception • Modem control function (channels 0 to 2 in asynchronous mode)
Serial communication interface	• Two channels • Clock synchronous mode, asynchronous mode, or smart card interface mode is selectable. • Simultaneous transmission and reception (full-duplex communication) supported • Dedicated baud rate generator • LSB first/MSB first selectable • Modem control function • Encoding and decoding of IrDA communications waveforms in accord with version 1.0 of the IrDA standard (on channel 0)
Renesas serial peripheral interface	• Three channels • SPI operation • Master mode and slave mode selectable • Programmable bit length, clock polarity, and clock phase can be selected. • Consecutive transfers • MSB first/LSB first selectable • Maximum transfer rate: 33.33 Mbps
SPI multi I/O bus controller	• One channel • Up to two serial flash memories with multiple I/O bus sizes (single/dual/quad) can be connected. • External address space read mode (built-in read cache) • SPI operating mode • Clock polarity and clock phase can be selected. • MSB first/LSB first selectable • Maximum transfer rate: 533.33Mbps (SDR transfer, with two serial flash memories connected) 1066.66Mbps (DDR transfer, with two serial flash memories connected) (RZ/A1LU only)
I^2C bus interface	• Four channels • Master mode and slave mode supported • Support for 7-bit and 10-bit slave address formats • Support for multi-master operation • Timeout detection

chanhvinh.com

Items	Specification
Serial sound interface	• Four-channel bidirectional serial transfer • Duplex communication (channels 0, 1, and 3) • Support of various serial audio formats • Support of master and slave functions • Generation of programmable word clock and bit clock • Multi-channel formats • Support of 8, 16, 18, 20, 22, 24, and 32-bit data formats • Support of eight-stage FIFO for transmission and reception • Support of TDM mode • Support of WS continue mode in which the SSIWS signal is not stopped. • Support of direct transfer to the SCUX module • A change of the sampling frequency can be detected.
Media local bus (RZ/A1L only)	• Conforms with version 2.0 of the MediaLB standard. Data transfer at up to 50 Mbps is possible.
SCUX	• Sampling rate conversion —Asynchronous or synchronous sampling rate conversion is possible. —Sampling rate (synchronous mode) Note: The selectable sampling rates depend on the number of used channels and rate ratio. Input [kHz]: 8, 11.025, 12, 16, 22.05, 24, 32, 44.1, 48, 64, 88.2, or 96 is selectable. Output [kHz]: 8, 16, 24, 44.1, 48, or 96 is selectable. —Sampling rate (asynchronous mode) Note: The selectable sampling rates depend on the number of used channels and rate ratio. Input/output [kHz]: 1 to 96 —Data format: 16 or 24 bits • Digital volume and mute functions —The digital volume can be set within the range from a multiple of 0 to 8 (−120 to 18 dB) —Volume ramping supports soft mute, fade-in, and fade-out. —The zero crossing mute function can apply muting at zero-crossing points. • Mixer —Data of two to four source systems can be mixed (added together) into one system. —The ratio to add the sources can be set. • Direct transfer to the serial sound interface module is supported.
CAN interface	• Two channels • ISO11898-1 compliant • Message buffer: —Up to 64 2-channel receive message buffers: shared among all channels. —16 transmit message buffers per channel
IEBus™ controller (RZ/A1L only)	• Conforms with the IEBus protocol (communication modes 1 and 2). • Transfer rates: approximately 18 kbps (in communication mode 1), approximately 27 kbps (in communication mode 2) • Maximum numbers of bytes for transfer: 32 bytes/frame (in communication mode 1), 128 bytes/frame (in communication mode 2) • Operating clock: 8 MHz Note: Input of peripheral clock 0 (P0φ) running at 32 MHz is required.
Renesas SPDIF interface	• Support of IEC60958 standard (stereo and consumer use modes only) • Sampling frequencies of 32 kHz, 44.1 kHz, and 48 kHz • Audio word sizes of 16 to 24 bits per sample • Biphase mark encoding • Double buffered data • Parity encoded serial data • Simultaneous transmit and receive • Receiver autodetects IEC 61937 compressed mode data.

chanhvinh.com

Items	Specification
CD-ROM decoder (RZ/A1L only)	• Support of five formats: Mode 0, mode 1, mode 2, mode 2 form 1, and mode 2 form 2 • Sync codes detection and protection (Protection: When a sync code is not detected, it is automatically inserted.) • Descrambling • ECC correction —P, Q, PQ, and QP correction —PQ or QP correction can be repeated up to three times. • EDC check Performed before and after ECC • Mode and form are automatically detected. • Link sectors are automatically detected. • Buffering data control Buffering CD-ROM data including Sync code is transferred in specified format, after the data is descrambled, corrected by ECC, and checked by EDC.
LIN interface (RZ/A1L only)	• Conforms with revisions 1.3, 2.0, 2.1, and 2.2 of the LIN protocol and SAEJ 2062. • Master mode selectable
Ethernet controller	• Conforms with the Ethernet or the IEEE802.3 MAC (Media Access Control) layer standard • MAC function Constructs/deconstructs data frames (frame format conforming to IEEE802.3, 2000 Edition) Supports transfer at 10 and 100 Mbps Supports full-duplex mode Flow control conforming to IEEE802.3x Supports an MII (Media Independent Interface) for connection to a PHY interface in conformance with IEEE 802.3 Upward protocol support (checksum) function • E-DMAC (Direct Memory Access Controller for Ethernet controller) function
EthernetAVB (RZ/A1LU only)	• Conforms with the Ethernet or the IEEE802.3 MAC (Media Access Control) layer standard • MAC function Constructs/deconstructs data frames (frame format conforming to IEEE802.3, 2000 Edition) Supports transfer at 100 Mbps Supports full-duplex mode Flow control conforming to IEEE802.3x Supports an MII (Media Independent Interface) for connection to a PHY interface in conformance with IEEE 802.3 Upward protocol support (checksum) function • AVB-DMAC (DMAC dedicated to EthernetAVB) function AVB-DMAC conforms with the following 3 standards; IEEE802.1AS (Clock Synchronization Protocol), IEEE802.1Qav (Realtime Transfer Protocol) and IEEE802.1Qat (Bandwidth Reservation Protocol)
USB 2.0 host/function module	• Two channels • Conforms to the Universal Serial Bus Specification Revision 2.0 • 480-Mbps, 12-Mbps, and 1.5-Mbps transfer rates provided (host mode) • 480-Mbps and 12-Mbps transfer rates provided (function mode) • On-chip 8-Kbyte RAM as communication buffers
Video display controller 5	• Video input interface BT601, BT656 format (NTSC/PAL) input: Input clock: 27 MHz/54 MHz Digital pin input (channel 0): YCbCr422, YCbCr444, RGB888, RGB666, RGB565 Digital pin input size: Maximum input video image size to be set*: 1440 pixels × 1024 lines (horizontal × vertical) Note:*Depends on the AC characteristics of the connected device. Examples of input video image size : XGA (1024 × 768), SVGA (800 × 600), WVGA (800 × 480), VGA (640 × 480), WQVGA (480 × 240), QVGA (320 × 240, 240 × 320) • Input video control Horizontal noise reduction (NR) and brightness and gain adjustment using matrix operation

chanhvinh.com

Items	Specification
Video display controller 5	• Scaling control Vertical and horizontal scaling up or down of input video possible at a desired ratio (scaling up of graphics also possible) Scaling up ratio: 1 to 8; scaling down ratio: 1/8 to 1 Interpolation: Hold or linear selectable 2D IP conversion: 2D IP conversion through separately setting the initial phases for the top and bottom fields • Video recording Output pixel format: YCbCr444, YCbCr422, RGB888, RGB565 Output field rate: 1/1, 1/2, 1/4, 1/8 Rotation: Horizontal mirroring and 90/180/270 degree rotation for YCbCr422 and RGB565 Maximum video image size to be stored: ×1 size of input video image • Output video control Black stretch: Black area stretched according to Y signal state Enhancer capability: LTI (transient improvement) and sharpness (contour emphasis) for Y signal • Three graphics layers (one of them also for input video) Available input pixel formats 1 bit/pixel: CLUT1 4 bits/pixel: CLUT4 8 bits/pixel: CLUT8 16 bits/pixel: YCbCr422 (graphics layers 0 and 1), RGB565, ARGB1555, RGBA5551, ARGB4444 32 bits/pixel: ARGB8888, RGBA8888, RGB888, YCbCr444 (graphics layer 0) • Superimposition Alpha blending in a rectangular area: Input video, layer 1, and layer 2 blended according to the transparency percentage α (fade-in and fade-out function available) Chroma key function: Mixing based on transparency percentage α using the specified RGB and CLUT value Pixel-base alpha blending: Alpha blending for each pixel based on transparency percentage α • Panel output control Panel output correction: Brightness adjustment and contrast adjustment, gamma correction, panel dithering TCON: Various timing output for LCD panel driving provided by a total of seven vertical and horizontal panel driver signals Panel output pixel format: RGB888, RGB666, RGB565, serial RGB Output video image size: Maximum output video image size to be set*: 1999 pixels × 2035 lines (horizontal × vertical) Note:*Depends on the AC characteristics of the display panel. Examples of output video image size: XGA (1024 × 768), SVGA (800 × 600), WVGA (800 × 480), VGA (640 × 480), WQVGA (480 × 240), QVGA (320 × 240, 240 × 320)
JPEG codec unit (RZ/A1LU only)	• Compression and decompression method conforming to the JPEG baseline standard within the range described in this document. • Operational precision: Conforming to JPEG Part 2, ISO-IEC10918-2 • Pixel format: Compression: YCbCr422 Decompression: YCbCr444, YCbCr422, YCbCr411, YCbCr420 Output pixel format to the buffer: YCbCr422, ARGB8888, RGB565 • Four quantization tables provided • Four Huffman tables provided (two tables for AC coefficients and two tables for DC coefficients) • Markers supported: SOI, SOF0, SOS, DQT, DHT, DRI, RSTm, and EOI • Image data rate: Max. 133.33 Mbytes/s (at 66.67-MHz operation)

chanhvinh.com

Items	Specification
Capture engine unit	• Examples of input video image size : 5 megapixels (2,560 × 1,920), 3 megapixels (2,048 × 1,536), 2 megapixels (1,632 × 1,224), UXGA (1,600 × 1,200), SXGA (1) (1,280 × 1,024), SXGA (2) (1,280 × 960), WXGA (1,280 × 768), XGA (1,024 × 768), SVGA (800 × 600), WVGA (800 × 480), VGA (640 × 480), WQVGA (480 × 240), QVGA (320 × 240, 240 × 320) Note: Depends on the AC characteristics of the connected device, frame rate of the connected device, and transfer speed to the destination RAM. • Input format: 8-bit YCbCr422 binary data • Memory output format: YCbCr422, YCbCr420 Note: The captured data cannot be displayed via the video display controller 5 because the Y data and CbCr data are split when written to memory.
SD host interface	• Two channels • SD memory I/O card interface (1-/4-bit SD bus) • Error check function: CRC7 (command), CRC16 (data) • Interrupt requests —Card access interrupt —SDIO access interrupt —Card detect interrupt • DMA transfer requests —SD_BUF write —SD_BUF read • Card detection function, write protect supported
MMC host interface	• Interface to multi-media card (MMC) • Data bus: 1-/4-/8-bit MMC mode • Interrupt requests: card detection, error/time-out, and normal operation • DMA transfer requests: CE_DATA write and CE_DATA read • Card detection function
General I/O ports	• 176-pin QFP or BGA: 78 I/Os, 8 inputs with open-drain outputs, and 14 inputs (input only) • 208-pin QFP: 100 I/O pins, 8 input pins with open-drain outputs, and 14 inputs (input only) • Input or output can be selected for each bit.
A/D converter	• 12-bit resolution • Eight input channels • Minimum conversion time: 5.0 µs • A/D conversion request by the external trigger or timer trigger
Debugging interface	• Arm CoreSight™ architecture • JTAG-standard pin assignment
On-chip RAM	• 3-Mbyte (RZ/A1L and RZ/A1LU) or 2-Mbyte (RZ/A1LC) large capacity memory for video display/ recording and work (128 Kbytes are used for data retention) • 128-Kbyte memory for data retention (16 Kbytes × 2, 32 Kbytes × 1, 64 Kbytes × 1)
Boot modes	• Four boot modes Boot mode 0: Booting from memory (bus width: 16 bits) connected to CS0 area Boot mode 1: Booting from a serial flash memory Boot mode 2: Booting from a NAND flash memory with SD controller Boot mode 3: Booting from a NAND flash memory with MMC controller
Power supply voltage	• Vcc: 1.10 to 1.26 V • PVcc: 3.0 to 3.6 V

chanhvinh.com

Items	Specification
Package	• PLBG0176KA-A 176-pin BGA, 8-mm square, 0.5-mm pitch JEITA package code: P-LFBGA176-8×8-0.50 RENESAS code: PLBG0176KA-A • PLQP0176KB-A 176-pin QFP, 24-mm square, 0.5-mm pitch JEITA package code: P-LFQFP176-24×24-0.50 RENESAS code: PLQP0176KB-A • PLQP0208KB-A 208-pin QFP, 28-mm square, 0.5-mm pitch JEITA package code: P-LFQFP208-28×28-0.50 RENESAS Code : PLQP0208KB-A • PRBG0233GA-A 233-pin BGA, 15-mm square, 0.8-mm pitch JEITA package code: P-FBGA233-15x15-0.80 RENESAS Code : PRBG0233GA-A

chanhvinh.com

1.2 Product Lineup

Table 1.2 Product Lineup

Group	Part Number	Temperature Range	Quality Level	Package
RZ/A1L	R7S721020VCBG	-40 to +85°C	Industry usage etc.	PLBG0176KA-A
	R7S721020VCFP		Industry usage etc.	PLQP0176KB-A
	R7S721020VLFP		Car Accessories	
	R7S721021VCFP		Industry usage etc.	PLQP0208KB-A
	R7S721021VLFP		Car Accessories	
RZ/A1LU	R7S721030VCBG	-40 to +85°C	Industry usage etc.	PLBG0176KA-A
	R7S721030VCFP		Industry usage etc.	PLQP0176KB-A
	R7S721030VLFP		Car Accessories	
	R7S721031VCFP		Industry usage etc.	PLQP0208KB-A
	R7S721031VLFP		Car Accessories	
	R7S721031VCBG		Industry usage etc.	PRBG0233GA-A
	R7S721031VLBG		Car Accessories	
RZ/A1LC	R7S721034VCBG	-40 to +85°C	Industry usage etc.	PLBG0176KA-A

1.3 Block Diagram

See section 5, LSI Internal Bus.

chanhvinh.com

1.4 Pin Assignment

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
A	Vss	Vcc	P5_2	P5_0	P4_5	P4_3	Vss	Vss	DM0	DP1	Vss	USB_X1	P2_8	P3_15	Vss	A
B	P5_3	Vss	Vcc	P5_1	P4_6	P4_4	P4_1	REFRIN	DP0	DM1	USB_X2	P2_9	P2_6	Vss	P1_7	B
C	P5_6	P5_4	Vss	Vcc	P4_7	P4_2	P4_0	USBAVcc	VBUS0	VBUS1	Vss	P2_7	Vss	P1_6	P1_5	C
D	P5_7	P5_5	P5_8	PVcc	Vcc	Vcc	Vcc	Vss	Vss	USBAPVcc	PVcc	Vss	PVcc	P1_4	P1_3	D
E	P5_10	P5_11	P5_9	PVcc								PVcc	P1_2	P1_0	P1_1	E
F	P5_15	P5_12	P5_14	P5_13								PVcc	P3_14	P3_13	P3_12	F
G	P6_2	P6_1	P6_0	Vss								P3_8	P3_9	P3_11	P3_10	G
H	P6_5	P6_4	P6_3	Vss								Vss	P3_6	P3_7	P3_5	H
J	P6_7	P6_9	P6_8	P6_6								Vss	P3_2	P3_4	P3_3	J
K	P6_10	P6_11	P6_12	PVcc								P3_1	P3_0	TCK	Vss	K
L	Vss	P6_13	P6_15	PVcc								TMS	$\overline{\text{TRST}}$	JP0_1	JP0_0	L
M	CKIO	P6_14	P7_1	Vss	PVcc	P0_3	P0_0	P0_2	PLLVcc	PVcc	PVcc	Vcc	P2_5	P2_4	Vss	M
N	P7_2	P7_0	Vss	P7_5	P7_8	P7_11	P0_1	Vss	NMI	P1_9	P1_11	P1_14	Vcc	BSCANP	AUDIO_X2	N
P	PVcc	Vss	P7_3	P7_6	P7_10	P2_0	P2_2	RTC_X1	XTAL	P1_8	P1_13	P1_12	AVcc	Vcc	AUDIO_X1	P
R	Vss	P7_4	Vss	P7_7	P7_9	P2_1	P2_3	RTC_X2	EXTAL	$\overline{\text{RES}}$	P1_10	P1_15	AVss	AVref	Vcc	R
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	

Figure 1.1 Pin Assignment of the 176-pin BGA (Upper Perspective View)

chanhvinh.com



Figure 1.2 Pin Assignment of the 176-pin QFP (Top View)

chanhvinh.com

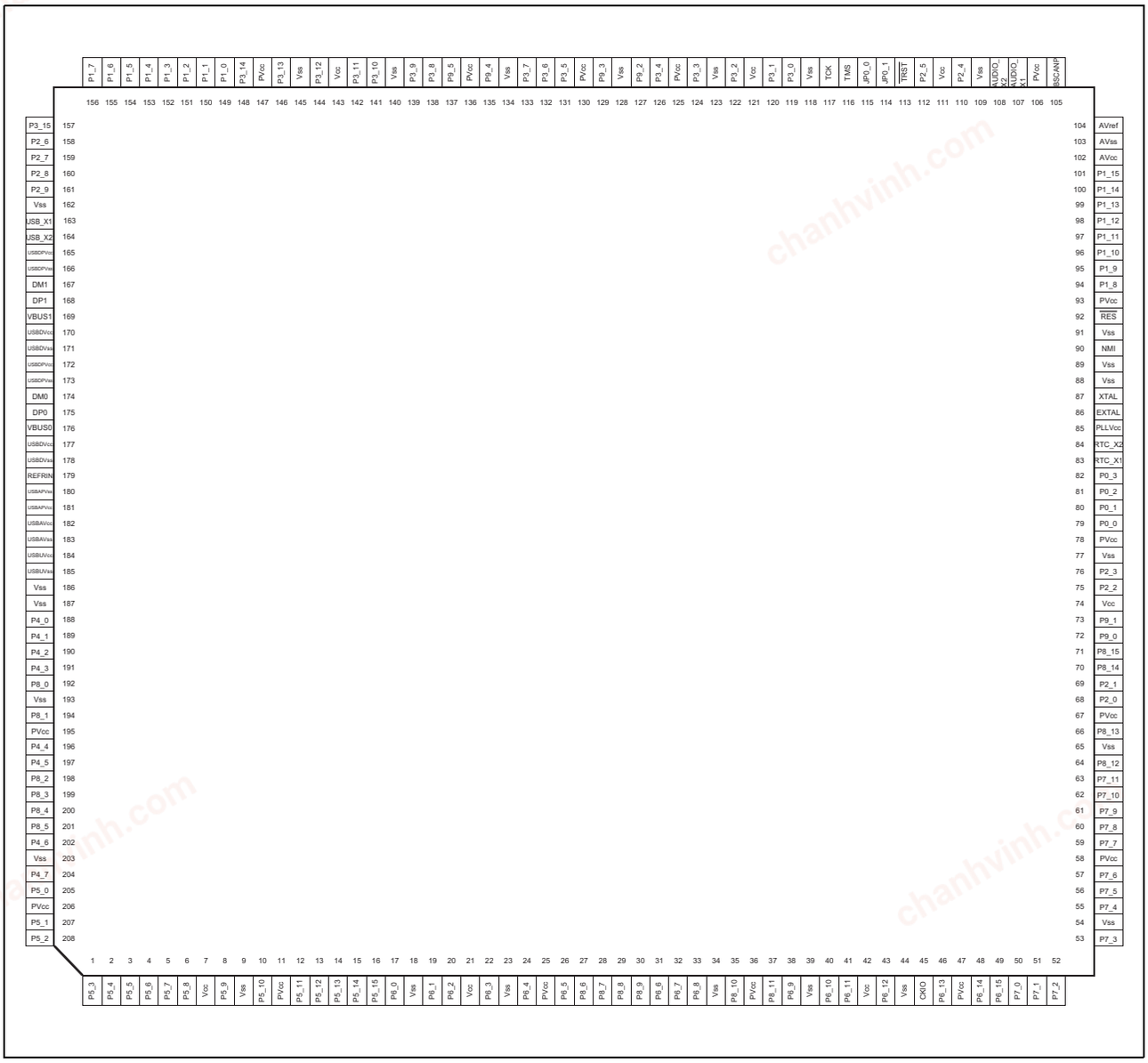


Figure 1.3 Pin Assignment of the 208-pin QFP (Top View)

chanhvinh.com

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
A	Vss	Vcc	P5_2	P5_0	P4_6	P4_5	Vss	P4_2	P4_0	Vss	DM0	DP1	Vss	USB_X1	P2_6	P3_15	Vss	A
B	P5_3	Vss	Vcc	P5_1	P8_5	P8_2	P8_1	P4_3	VSS	REFRIN	DP0	DM1	USB_X2	P2_8	P2_7	Vss	PVcc	B
C	P5_7	P5_4	Vss	Vcc	P4_7	P8_4	P4_4	P8_0	P4_1	USBAVcc	VBUS0	VBUS1	PVcc	P2_9	Vss	PVcc	P1_7	C
D	P5_10	P5_8	P5_6	Vss	Vcc	Vcc	P8_3	PVcc	PVcc	Vss	Vss	USBAPVcc	Vss	Vss	PVcc	P1_6	P1_5	D
E	P5_12	P5_11	P5_9	P5_5										PVcc	P1_4	P1_3	P1_2	E
F	P5_15	P5_14	P5_13	PVcc										PVcc	P1_1	P1_0	P3_14	F
G	P6_1	P6_2	P6_0	PVcc			VSS	VSS	VSS	VSS	VSS			P3_10	P3_13	P3_12	P3_11	G
H	P6_5	P6_4	P6_3	Vcc			VSS	VSS	VSS	VSS	VSS			P3_9	P3_8	P9_5	P9_4	H
J	P8_8	P8_7	P8_6	Vcc			VSS	VSS	VSS	VSS	VSS			Vcc	P3_6	P3_7	Vss	J
K	P8_9	P6_6	P6_7	P6_8			VSS	VSS	VSS	VSS	VSS			Vcc	P9_3	P3_5	P9_2	K
L	P8_10	P8_11	P6_9	PVcc			VSS	VSS	VSS	VSS	VSS			Vcc	P3_2	P3_4	Vss	L
M	Vss	P6_11	P6_10	PVcc										$\overline{\text{TRST}}$	TMS	P3_1	P3_3	M
N	CKIO	P6_12	P6_13	PVcc										BSCANP	JP0_1	TCK	P3_0	N
P	PVcc	P6_14	P7_1	Vss	P7_3	P7_9	P7_11	PVcc	PVcc	PLLVcc	Vss	PVcc	PVcc	Vcc	P2_4	JP0_0	Vss	P
R	P6_15	P7_0	Vss	P7_5	P7_7	P8_13	P8_15	P2_2	P0_0	P0_3	NMI	P1_10	P1_14	P1_15	Vcc	P2_5	AUDIO_X2	R
T	P7_2	Vss	P7_6	P7_8	P8_12	P2_1	P9_0	P2_3	RTC_X1	P0_2	XTAL	P1_8	P1_11	P1_13	AVcc	Vcc	AUDIO_X1	T
U	Vss	P7_4	Vss	P7_10	P2_0	P8_14	P9_1	P0_1	RTC_X2	Vss	EXTAL	$\overline{\text{RES}}$	P1_9	P1_12	AVss	AVref	Vcc	U
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	

Figure 1.4 Pin Assignment of the 233-pin BGA (Upper Perspective View)

chanhvinh.com

1.5 Pin Functions

Table 1.3 Pin Functions

Classification	Symbol	I/O	Name	Function
Power supply	Vcc	I	Power supply	Power supply pins. All the Vcc pins must be connected to the system power supply. This LSI does not operate correctly if there is a pin left open.
	Vss	I	Ground	Ground pins. All the Vss pins must be connected to the system power supply (0 V). This LSI does not operate correctly if there is a pin left open.
	PVcc	I	Power supply for I/O circuits	Power supply for I/O pins. All the PVcc pins must be connected to the system power supply. This LSI does not operate correctly if there is a pin left open.
	PLLVcc	I	Power supply for PLL	Power supply for the on-chip PLL oscillator.
Clock	EXTAL	I	External clock	Connected to a crystal resonator. An external clock signal may also be input to the EXTAL pin.
	XTAL	O	Crystal	Connected to a crystal resonator.
	CKIO	O	System clock output	Supplies the system clock to external devices.
	AUDIO_CLK	I	External clock for audio	Input pin of external clock for audio. A clock input to the divider is selected from an oscillation clock input on this pin or pins AUDIO_X1 and AUDIO_X2.
	AUDIO_X1	I	Crystal resonator/ external clock for audio	Pins connected to a crystal resonator for audio. An external clock can be input on pin AUDIO_X1. A clock input to the divider is selected from an oscillation clock input on these pins or the AUDIO_CLK pin.
	AUDIO_X2	O		
	AUDIO_XOUT	O	AUDIO_X1 clock output	Output for the on-chip crystal oscillator on AUDIO_X1 or the external clock signal.
	AUDIO_XOUT2	O	AUDIO_X1 divided-by-two clock output	Output for the on-chip crystal oscillator on AUDIO_X1 or the external clock signal after frequency division of the selected signal by two.
	AUDIO_XOUT3	O	AUDIO_X1 divided-by-three clock output	Output for the on-chip crystal oscillator on AUDIO_X1 or the external clock signal after frequency division of the selected signal by three.
Operating mode control	MD_BOOT1, MD_BOOT0	I	Mode set	Sets the operating mode. Do not change the signal levels on these pins while the $\overline{\text{RES}}$ pin is asserted or until the mode is fixed, after the negation.
	MD_CLK	I	Clock mode set	Sets the clock operating mode. Do not change the signal levels on this pin while the $\overline{\text{RES}}$ pin is asserted or until the mode is fixed, after the negation.
	MD_CLKS	I	SSCG clock mode set	Switches the SSCG circuit on or off. Do not change the signal levels on this pin while the $\overline{\text{RES}}$ pin is asserted or until the mode is fixed, after the negation.
	BSCANP	I	Boundary scan set	Boundary scan setting pin. This pin is set to the high level for a boundary scan and to the low level for normal operation.

chanhvinh.com

Classification	Symbol	I/O	Name	Function
System control	$\overline{\text{RES}}$	I	Power-on reset	This LSI enters the power-on reset state when this signal goes low.
	$\overline{\text{WDTOVF}}$	O	Watchdog timer overflow	Outputs an overflow signal from the watchdog timer.
Interrupts	NMI	I	Non-maskable interrupt	Non-maskable interrupt request pin. It is handled as an FIQ exception. Fix it high when not in use.
	IRQ7 to IRQ0	I	Interrupt requests 7 to 0	Maskable interrupt request pins. Level-input or edge-input detection can be selected. When the edge-input detection is selected, the rising edge, falling edge, or both edges can also be selected.
	TINT121 to TINT0	I	Interrupt requests 121 to 0	Maskable interrupt request pins. High-level-input or rising edge-input detection can be selected.
Address bus	A25 to A0	O	Address bus	Outputs addresses.
Data bus	D31 to D0	I/O	Data bus	Bidirectional data bus.
Bus control	$\overline{\text{CS5}}$ to $\overline{\text{CS0}}$	O	Chip select 5 to 0	Chip-select signals for external memory or devices.
	$\overline{\text{RD}}$	O	Read	Indicates that data is read from an external device.
	RD/$\overline{\text{WR}}$	O	Read/write	Read/write signal.
	$\overline{\text{BS}}$	O	Bus start	Bus-cycle start signal.
	$\overline{\text{AH}}$	O	Address hold	Address hold timing signal for the device that uses the address/data-multiplexed bus.
	$\overline{\text{WAIT}}$	I	Wait	Inserts a wait cycle into the bus cycles during access to the external space.
	$\overline{\text{WE0}}$	O	Byte select	Indicates a write access to bits 7 to 0 of data of external memory or device.
	$\overline{\text{WE1}}$	O	Byte select	Indicates a write access to bits 15 to 8 of data of external memory or device.
	$\overline{\text{WE2}}$	O	Byte select	Indicates a write access to bits 23 to 16 of data of external memory or device.
	$\overline{\text{WE3}}$	O	Byte select	Indicates a write access to bits 31 to 24 of data of external memory or device.
	DQMLL	O	Byte select	Selects bits D7 to D0 when SDRAM is connected.
	DQMLU	O	Byte select	Selects bits D15 to D8 when SDRAM is connected.
	DQMUL	O	Byte select	Selects bits D23 to D16 when SDRAM is connected.
	DQMUU	O	Byte select	Selects bits D31 to D24 when SDRAM is connected.
	$\overline{\text{RAS}}$	O	RAS	Connected to the $\overline{\text{RAS}}$ pin when SDRAM is connected.
	$\overline{\text{CAS}}$	O	CAS	Connected to the $\overline{\text{CAS}}$ pin when SDRAM is connected.
	CKE	O	CK enable	Connected to the CKE pin when SDRAM is connected.

chanhvinh.com

Classification	Symbol	I/O	Name	Function
Direct memory access controller	DREQ0	I	DMA-transfer request	Input pin to receive external requests for DMA transfer.
	DACK0	O	DMA-transfer request accept	Output pin for signals indicating acceptance of external requests from external devices.
	TEND0	O	DMA-transfer end output	Output pin for DMA transfer end.
Multi-function timer pulse unit 2	TCLKA, TCLKB, TCLKC, TCLKD	I	Timer clock input	External clock input pins for the timer.
	TIOC0A, TIOC0B, TIOC0C, TIOC0D Note: TIOC0B pin is not present on products in the 176-pin package.	I/O	Input capture/ output compare (channel 0)	The TGRA_0 to TGRD_0 input capture input/output compare output/PWM output pins.
	TIOC1A, TIOC1B	I/O	Input capture/ output compare (channel 1)	The TGRA_1 and TGRB_1 input capture input/output compare output/PWM output pins.
	TIOC2A, TIOC2B	I/O	Input capture/ output compare (channel 2)	The TGRA_2 and TGRB_2 input capture input/output compare output/PWM output pins.
	TIOC3A, TIOC3B, TIOC3C, TIOC3D	I/O	Input capture/ output compare (channel 3)	The TGRA_3 to TGRD_3 input capture input/output compare output/PWM output pins.
	TIOC4A, TIOC4B, TIOC4C, TIOC4D	I/O	Input capture/ output compare (channel 4)	The TGRA_4 to TGRD_4 input capture input/output compare output/PWM output pins.
Realtime clock	RTC_X1	I	Crystal resonator for realtime clock/ external clock	Connected to 32.768-kHz crystal resonator. The RTC_X1 pin can also be used to input an external clock.
	RTC_X2	O		
Serial communication interface with FIFO	TxD4 to TxD0	O	Transmit data	Data output pins.
	RxD4 to RxD0	I	Receive data	Data input pins.
	SCK4 to SCK0	I/O	Serial clock	Clock input/output pins.
	$\overline{\text{RTS2}}$ to $\overline{\text{RTS0}}$	I/O	Transmit request	Modem control pins.
	$\overline{\text{CTS2}}$ to $\overline{\text{CTS0}}$	I/O	Transmit enable	Modem control pins.
Serial communication interface	SCI_SCK1, SCI_SCK0	I/O	Serial clock	Clock input/output pins.
	SCI_TXD1, SCI_TXD0	O	Transmit data	Data output pins.
	SCI_RXD1, SCI_RXD0	I	Receive data	Data input pins.
	$\overline{\text{SCI_CTS1}}$/$\overline{\text{RTS1}}$, $\overline{\text{SCI_CTS0}}$/$\overline{\text{RTS0}}$	I/O	Transmit and receive start control	I/O pins for controlling the start of transmission and reception.
Renesas serial peripheral interface	MOSI2 to MOSI0	I/O	Data	Data I/O pins.
	MISO2 to MISO0	I/O	Data	Data I/O pins.
	RSPCK2 to RSPCK0	I/O	Clock	Clock I/O pins.
	SSL20, SSL10, SSL00	I/O	Slave select	Slave select I/O pins.

chanhvinh.com

Classification	Symbol	I/O	Name	Function
SPI multi I/O bus controller	SPBCLK_0	O	Clock	Clock output pins.
	SPBSSL_0	O	Slave select	Slave select output pins.
	SPBMO0_0/SPBIO00_0, SPBMI0_0/SPBIO10_0, SPBIO20_0, SPBIO30_0, SPBMO1_0/SPBIO01_0, SPBMI1_0/SPBIO11_0, SPBIO21_0, SPBIO31_0	I/O	Data	Data I/O pins for channel.
I^2C bus interface	RIIC3SCL to RIIC0SCL	I/O	Serial clock pin	Serial clock I/O pins.
	RIIC3SDA to RIIC0SDA	I/O	Serial data pin	Serial data I/O pins.
Serial sound interface	SSITxD3, SSITxD1, SSITxD0	O	Data output	Serial data output pin.
	SSIRxD3, SSIRxD1, SSIRxD0	I	Data input	Serial data input pin.
	SSIDATA2	I/O	Data I/O	Serial data I/O pins.
	SSISCK3 to SSISCK0	I/O	SSI clock I/O	I/O pins for serial clocks.
	SSIWS3 to SSIWS0	I/O	SSI clock LR I/O	I/O pins for word selection.
Media local bus (RZ/A1L only)	MLB_CLK	I	Clock input	MediaLB clock input pin.
	MLB_SIG	I/O	Signal information I/O	MediaLB signal information I/O pin.
	MLB_DAT	I/O	Data I/O	MediaLB data I/O pin.
CAN interface	CAN_CLK	I	Clock source for CAN communication	Clock source for CAN communication.
	CAN1TX, CAN0TX	O	CAN bus transmit data	Output pins for transmit data on the CAN bus.
	CAN1RX, CAN0RX	I	CAN bus receive data	Input pins for receive data on the CAN bus.
IEBus™ controller (RZ/A1L only)	IETxD	O	IEBus™ controller transmit data	Output pin for transmit data on IEBus™ controller.
	IERxD	I	IEBus™ controller receive data	Input pin for receive data on IEBus™ controller.
Renesas SPDIF interface	SPDIF_OUT	O	Output data	Transmit data output pin.
	SPDIF_IN	I	Input data	Receive data input pin.
LIN interface (RZ/A1L only)	RLIN30TX	O	Output data	Transmit data output pins.
	RLIN30RX	I	Input data	Receive data input pins.
Ethernet controller, EthernetAVB (RZ/A1LU only) Note: Regarding the switching of pin functions between Ethernet controller and EthernetAVB, refer to section 41, Ports.	ET_TXCLK	I	Transmit clock	Clock pin for transmission.
	ET_TXEN	O	Transmit enable	Transmit data enable pin
	ET_TXD3 to ET_TXD0	O	Transmit data	MII transmit data pins.
	ET_COL	I	Collision detection	Collision detection pin.
	ET_TXER	O	Transmit error	Transmit error output pin.
	ET_RXCLK	I	Receive clock	Receive clock pin
	ET_RXDV	I	Receive enable	Receive data enable pin
	ET_RXD3 to ET_RXD0	I	Receive data	MII receive data pins.
	ET_RXER	I	Receive error	Receive error input pin.
	ET_CRS	I	Carrier detection	Carrier detection pin.
	ET_MDC	O	Management data clock	Clock pin for information transfer via MDIO.
	ET_MDIO	I/O	Management data I/O	Bidirectional pin for exchange of management data
EthernetAVB (RZ/A1LU only)	AVB_CAPTURE	I	Timer capture	Capturing input pin for AVTP presentation timer
	AVB_GPTP_EXTERN	I	gPTP timer external clock	External clock pin for gPTP timer

chanhvinh.com

Classification	Symbol	I/O	Name	Function
USB 2.0 host/ function module	DP1, DP0	I/O	USB 2.0 host/function module D+ data	D+ data pins for USB 2.0 host/function module bus.
	DM1, DM0	I/O	USB 2.0 host/function module D– data	D– data pins for USB 2.0 host/function module bus.
	VBUS1, VBUS0	I	VBUS input	Connected to Vbus on USB 2.0 host/ function module bus.
	REFRIN	I	Reference input	Connected to USBAPVss via 5.6-kΩ ± 1% resistance. (QFP package) Connected to Vss via 5.6-kΩ ± 1% resistance. (BGA package)
	USB_X1	I	Crystal resonator/ external clock for USB 2.0 host/function module	Connected to a crystal resonator for USB 2.0 host/function module. An external clock signal may also be input to the USB_X1 pin.
	USB_X2	O		
	USBAPVcc	I	Power supply for transceiver analog pins	Power supply for pins.
	USBAPVss Note: This pin is not present on products in the BGA package.	I	Ground for transceiver analog pins	Ground for pins.
	USBDPVcc Note: This pin is not present on products in the BGA package.	I	Power supply for transceiver digital pins	Power supply for pins.
	USBDPVss Note: This pin is not present on products in the BGA package.	I	Ground for transceiver digital pins	Ground for pins.
	USBAVcc	I	Power supply for transceiver analog core	Power supply for core.
	USBAVss Note: This pin is not present on products in the BGA package.	I	Ground for transceiver analog core	Ground for core.
	USBDVcc Note: This pin is not present on products in the BGA package.	I	Power supply for transceiver digital core	Power supply for core.
	USBDVss Note: This pin is not present on products in the BGA package.	I	Ground for transceiver digital core	Ground for core.
	USBUVcc Note: This pin is not present on products in the BGA package.	I	480-MHz power supply for USB 2.0 host/function module	Power supply for 480-MHz sections
	USBUVss Note: This pin is not present on products in the BGA package.	I	480-MHz ground for USB 2.0 host/function module	Ground for 480-MHz sections
Video display controller 5	LCD0_DATA23 to LCD0_DATA0	O	Output data	Data output pins for panel.
	LCD0_TCON6 to LCD0_TCON0	O	Panel timing adjustment output	Output pins for panel timing adjustment
	LCD0_CLK	O	Panel clock	Panel clock output pins.
	LCD0_EXTCLK	I	Panel clock source	Panel clock source input pins.
	DV0_DATA23 to DV0_DATA0	I	Input data	Data input pins for graphics data.
	DV0_VSYNC	I	VSYNC input	VSYNC input pins.
	DV0_HSYNC	I	HSYNC input	HSYNC input pins.
	DV0_CLK	I	Input clock	Clock input signal pins for graphics data.

chanhvinh.com

Classification	Symbol	I/O	Name	Function
Capture engine unit	VIO_D7 to VIO_D0	I	Input data	Graphics data input pins.
	VIO_CLK	I	Input clock	Graphics data clock input pin.
	VIO_VD	I	VSYNC input	VSYNC input pin.
	VIO_HD	I	HSYNC input	HSYNC input pin.
	VIO_FLD	I	FIELD input	Input pin for field information
SD host interface	SD_CLK_0, SD_CLK_1	O	SD clock	Output pins for SD clock.
	SD_CMD_0, SD_CMD_1	I/O	SD command	SD command output and response input signals.
	SD_D3_0 to SD_D0_0, SD_D3_1 to SD_D0_1	I/O	SD data	SD data bus signals.
	SD_CD_0, SD_CD_1	I	SD card detection	SD card detection.
	SD_WP_0, SD_WP_1	I	SD write protection	SD write protection signals.
MMC host interface	MMC_CLK	O	MMC clock	Output pin for MMC clock.
	MMC_CMD	I/O	MMC command	MMC command output and response input signal.
	MMC_D7 to MMC_D0	I/O	MMC data	MMC data bus signals.
	MMC_CD	I	MMC card detection	MMC card detection.
A/D converter	AN7 to AN0	I	Analog input pins	Analog input pins.
	$\overline{\text{ADTRG}}$	I	A/D conversion trigger input	External trigger input pin for starting A/D conversion.
	AVcc	I	Analog power supply	Power supply pin for A/D converter.
	AVss	I	Analog ground	Ground pin for A/D converter.
	AVref	I	Analog reference voltage	Reference voltage pin for A/D converter.
General I/O ports	P2_0 to P2_9, P3_0 to P3_15, P4_0 to P4_7, P5_0 to P5_15, P6_0 to P6_15, P7_0 to P7_11, P8_0 to P8_15 (208-pin QFP products only), P9_0 to P9_5 (208-pin QFP products only)	I/O	General port	78 general I/O port pins in 176-pin QFP and 176-pin BGA products. 100 general I/O port pins in 208-pin QFP products.
	P1_0 to P1_7	I/O	General port	8 input port pins with open-drain output.
	JP0_0, JP0_1, P0_0 to P0_3, P1_8 to P1_15	I	General port	14 general input port pins.
Debugging interface	TCK/SWDCLK	I	Test clock	Test-clock input pin. Also used as the input clock pin for serial wire debugging
	TMS/SWDIO	I, I/O	Test mode select	Test-mode select signal input pin. Also used as the I/O data pin for serial wire debugging
	TDI	I	Test data input	Serial input pin for instructions and data.
	TDO	O	Test data output	Serial output pin for instructions and data.
	$\overline{\text{TRST}}$	I	Test reset	Initialization-signal input pin. Note: When the chip is in CoreSight debugging mode, do not negate the $\overline{\text{TRST}}$ signal while the $\overline{\text{RES}}$ signal is at the high-level.
	TRACEDATA3 to TRACEDATA0	O	Data output	Trace data output pins.
	TRACECLK	O	Clock output	Trace clock output pin.
	TRACECTL	O	Enable output	Trace enable output pin.

chanhvinh.com

1.6 List of Pins

Table 1.4 List of Pins

176 QFP	176 BGA	208 QFP	233 BGA	Port function/ dedicated function		Mode function		Function 1		Function 2		Function 3		Function 4		Function 5		Function 6		Function 7		Function 8		Simplified Circuit Diagram
No.	No.	No.	No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Figure 1.5
1	B1	1	B1	P5_3	I(s)/O	—	—	D3	I(s)/O	MMC_D7	I(s)/O	ET_TXD3	O	DV0_DATA19	I(s)	LCD0_TCON3	O	—	—	—	—	—	—	(8)
2	C2	2	C2	P5_4	I(s)/O	—	—	D4	I(s)/O	RSPCK2	I(s)/O	SSISCK1	I(s)/O	DV0_DATA20	I(s)	—	—	—	—	—	—	—	—	(8)
3	D2	3	E4	P5_5	I(s)/O	—	—	D5	I(s)/O	SSL20	I(s)/O	SSIWS1	I(s)/O	DV0_DATA21	I(s)	—	—	—	—	—	—	—	—	(8)
4	C1	4	D3	P5_6	I(s)/O	—	—	D6	I(s)/O	MOSI2	I(s)/O	SSITxD1	O	DV0_DATA22	I(s)	SCK2	I(s)/O	—	—	—	—	—	—	(8)
5	D1	5	C1	P5_7	I(s)/O	—	—	D7	I(s)/O	MISO2	I(s)/O	SSIRxD1	I(s)	DV0_DATA23	I(s)	TxD2	O	—	—	—	—	—	—	(8)
6	D3	6	D2	P5_8	I(s)/O	—	—	D8	I(s)/O	CAN0RX	I(s)	TIOC4A	I(s)/O	IRQ3	I(s)	—	—	—	—	—	—	—	—	(8)
7	—	7	—	Vcc																				
8	E3	8	E3	P5_9	I(s)/O	—	—	D9	I(s)/O	CAN0TX	O	TIOC4B	I(s)/O	IRQ4	I(s)	—	—	—	—	—	—	—	—	(8)
9	—	9	—	Vss																				
10	E1	10	D1	P5_10	I(s)/O	—	—	D10	I(s)/O	IERxD*3	I(s)	TIOC4C	I(s)/O	IRQ5	I(s)	—	—	—	—	—	—	—	—	(8)
11	—	11	—	PVcc																				
12	E2	12	E2	P5_11	I(s)/O	—	—	D11	I(s)/O	IETxD*3	O	TIOC4D	I(s)/O	IRQ6	I(s)	—	—	—	—	—	—	—	—	(8)
13	F2	13	E1	P5_12	I(s)/O	—	—	D12	I(s)/O	SSISCK2	I(s)/O	SCK4	I(s)/O	AUDIO_XOUT2	O	—	—	—	—	—	—	—	—	(8)
14	F4	14	F3	P5_13	I(s)/O	—	—	D13	I(s)/O	SSIWS2	I(s)/O	AUDIO_XOUT	O	AUDIO_XOUT3	O	—	—	—	—	—	—	—	—	(8)
15	F3	15	F2	P5_14	I(s)/O	—	—	D14	I(s)/O	SSIDATA2	I(s)/O	RxD4	I(s)	TIOC2A	I(s)/O	—	—	—	—	—	—	—	—	(8)
16	F1	16	F1	P5_15	I(s)/O	—	—	D15	I(s)/O	SD_WP_1	I(s)	TxD4	O	—	—	—	—	—	—	—	—	—	—	(8)
17	G3	17	G3	P6_0	I(s)/O	—	—	D16	I(s)/O	LCD0_DATA8	O	RSPCK0	I(s)/O	TCLKA	I(s)	$\overline{\text{WDTOVF}}$	O	—	—	—	—	—	—	(8)
18	—	18	—	Vss																				
19	G2	19	G1	P6_1	I(s)/O	—	—	D17	I(s)/O	LCD0_DATA9	O	SSL00	I(s)/O	TCLKB	I(s)	—	—	—	—	—	—	—	—	(8)
20	G1	20	G2	P6_2	I(s)/O	—	—	D18	I(s)/O	LCD0_DATA10	O	MOSI0	I(s)/O	TCLKC	I(s)	—	—	—	—	—	—	—	—	(8)
21	—	21	—	Vcc																				
22	H3	22	H3	P6_3	I(s)/O	—	—	D19	I(s)/O	LCD0_DATA11	O	MISO0	I(s)/O	TCLKD	I(s)	—	—	—	—	—	—	—	—	(8)
23	—	23	—	Vss																				
24	H2	24	H2	P6_4	I(s)/O	—	—	D20	I(s)/O	LCD0_DATA12	O	SSISCK3	I(s)/O	MLB_CLK*1	I(s)	—	—	—	—	—	—	—	—	(8)
														$\overline{\text{AVB_CAPTURE}}$*2	I(s)									
25	—	25	—	PVcc																				
26	H1	26	H1	P6_5	I(s)/O	—	—	D21	I(s)/O	LCD0_DATA13	O	SSIWS3	I(s)/O	MLB_SIG*1	I(s)/O	—	—	—	—	—	—	—	—	(8)
														AVB_GPTP_EXTERN*2	I(s)									
—	—	27	J3	P8_6	I(s)/O	—	—	LCD0_DATA6	O	ET_TXEN	O	IRQ6	I(s)	$\overline{\text{CTS1}}$	I(s)/O	TIOC0C	I(s)/O	—	—	—	—	—	—	(7)
—	—	28	J2	P8_7	I(s)/O	—	—	LCD0_DATA7	O	ET_RXD0	I(s)	IRQ7	I(s)	$\overline{\text{RTS1}}$	I(s)/O	TIOC0D	I(s)/O	—	—	—	—	—	—	(7)
—	—	29	J1	P8_8	I(s)/O	—	—	LCD0_TCON0	O	ET_RXD1	I(s)	AUDIO_XOUT	O	SCK2	I(s)/O	AUDIO_XOUT3	O	—	—	—	—	—	—	(7)
—	—	30	K1	P8_9	I(s)/O	—	—	LCD0_TCON1	O	ET_RXD2	I(s)	CAN1TX	O	RxD2	I(s)	AUDIO_XOUT2	O	—	—	—	—	—	—	(7)
27	J4	31	K2	P6_6	I(s)/O	—	—	D22	I(s)/O	LCD0_DATA14	O	SSITxD3	O	MLB_DAT*3	I(s)/O	—	—	—	—	—	—	—	—	(8)
28	J1	32	K3	P6_7	I(s)/O	—	—	D23	I(s)/O	LCD0_DATA15	O	SSIRxD3	I(s)	IRQ0	I(s)	TIOC3A	I(s)/O	RLIN30 RX*3	I(s)	—	—	TRACE DATA0*4	O	(8)
29	J3	33	K4	P6_8	I(s)/O	—	—	D24	I(s)/O	LCD0_DATA16	O	SSISCK0	I(s)/O	IRQ1	I(s)	TIOC3B	I(s)/O	RLIN30 TX*3	O	—	—	TRACE DATA1*4	O	(8)
—	—	34	—	Vss																				
—	—	35	L1	P8_10	I(s)/O	—	—	LCD0_TCON2	O	ET_RXD3	I(s)	CAN1RX	I(s)	TxD2	O	—	—	—	—	—	—	—	—	(7)
—	—	36	—	PVcc																				
—	—	37	L2	P8_11	I(s)/O	—	—	LCD0_TCON3	O	—	—	SSISCK2	I(s)/O	SCK4	I(s)/O	—	—	—	—	—	—	—	—	(7)
30	J2	38	L3	P6_9	I(s)/O	—	—	D25	I(s)/O	LCD0_DATA17	O	SSIWS0	I(s)/O	IRQ2	I(s)	TIOC3C	I(s)/O	—	—	—	—	TRACE DATA2*4	O	(8)
31	—	39	—	Vss																				
32	K1	40	M3	P6_10	I(s)/O	—	—	D26	I(s)/O	LCD0_DATA18	O	SSITxD0	O	IRQ3	I(s)	TIOC3D	I(s)/O	CAN1 TX	O	—	—	TRACE DATA3*4	O	(8)
33	K2	41	M2	P6_11	I(s)/O	—	—	D27	I(s)/O	LCD0_DATA19	O	SSIRxD0	I(s)	SSIDATA2	I(s)/O	SCK0	I(s)/O	CAN1 RX	I(s)	—	—	TRACE CTL*4	O	(8)

chanhvinh.com

176 QFP	176 BGA	208 QFP	233 BGA	Port function/ dedicated function		Mode function		Function 1		Function 2		Function 3		Function 4		Function 5		Function 6		Function 7		Function 8		Simplified Circuit Diagram
No.	No.	No.	No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Figure 1.5
34	—	42	—	Vcc																				
35	K3	43	N2	P6_12	I(s)/O	—	—	D28	I(s)/O	LCD0_ DATA20	O	RSPCK1	I(s)/O	SSI SCK2	I(s)/O	$\overline{\text{RTS0}}$	I(s)/O	DV0_ DATA0	I(s)	—	—	—	—	(8)
36	—	44	—	Vss																				
37	M1	45	N1	CKIO	O	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	(5)
38	L2	46	N3	P6_13	I(s)/O	—	—	D29	I(s)/O	LCD0_ DATA21	O	SSL10	I(s)/O	SSI WS2	I(s)/O	$\overline{\text{CTS0}}$	I(s)/O	DV0_ DATA1	I(s)	—	—	—	—	(8)
39	—	47	—	PVcc																				
40	M2	48	P2	P6_14	I(s)/O	—	—	D30	I(s)/O	LCD0_ DATA22	O	MOSI1	I(s)/O	SSI DATA2	I(s)/O	RxD0	I(s)	DV0_ DATA2	I(s)	—	—	—	—	(8)
41	L3	49	R1	P6_15	I(s)/O	—	—	D31	I(s)/O	LCD0_ DATA23	O	MISO1	I(s)/O	—	—	TxD0	O	DV0_ DATA3	I(s)	—	—	—	—	(8)
42	N2	50	R2	P7_0	I(s)/O	—	—	LCD0_ EXTCLK	I(s)	MMC_ CD	I(s)	SD_ CD_1	I(s)	SPDIF_O UT	O	TIOC2A	I(s)/O	DV0_ DATA4	I(s)	SCI_ SCK0	I(s)/O	TRACE CLK*4	O	(7)
43	M3	51	P3	P7_1	I(s)/O	—	—	$\overline{\text{CS1}}$	O	AUDIO_ XOUT	O	SD_ WP_1	I(s)	TxD2	O	—	—	DV0_ DATA5	I(s)	SCI_ RXD0/ IrRXD	I(s)	—	—	(7)
44	N1	52	T1	P7_2	I(s)/O	—	—	$\overline{\text{CS4}}$	O	MMC_D1	I(s)/O	SD_ D1_1	I(s)/O	IRQ4	I(s)	CAN0RX	I(s)	DV0_ DATA6	I(s)	SCI_ TXD0/ IrTXD	O	—	—	(7)
45	P3	53	P5	P7_3	I(s)/O	—	—	$\overline{\text{CS5}}$	O	MMC_D0	I(s)/O	SD_ D0_1	I(s)/O	IRQ3	I(s)	CAN0TX	O	DV0_ DATA7	I(s)	$\overline{\text{SCI_ CTS0/}}$ RTS0	I(s)/O	—	—	(7)
46	—	54	—	Vss																				
47	R2	55	U2	P7_4	I(s)/O	—	—	$\overline{\text{WAIT}}$	I(s)	MMC_ CLK	O	SD_ CLK_1	O	—	—	IETxD*3	O	LCD0_ CLK	O	SCI_ SCK1	I(s)/O	—	—	(7)
48	N4	56	R4	P7_5	I(s)/O	—	—	$\overline{\text{BS}}$	O	MMC_ CMD	I(s)/O	SD_ CMD_1	I(s)/O	TxD0	O	IERxD*3	I(s)	LCD0_ TCON4	O	SCI_ RXD1	I(s)	—	—	(7)
49	P4	57	T3	P7_6	I(s)/O	—	—	$\overline{\text{WE2/}}$ DQMUL	O	MMC_D3	I(s)/O	SD_ D3_1	I(s)/O	IRQ6	I(s)	$\overline{\text{CTS2}}$	I(s)/O	LCD0_ TCON5	O	SCI_ TXD1	O	—	—	(7)
50	—	58	—	PVcc																				
51	R4	59	R5	P7_7	I(s)/O	—	—	$\overline{\text{WE3/}}$ DQMUU/ AH	O	MMC_D2	I(s)/O	SD_ D2_1	I(s)/O	IRQ5	I(s)	$\overline{\text{RTS2}}$	I(s)/O	LCD0_ TCON6	O	$\overline{\text{SCI_ CTS1/}}$ RTS1	I(s)/O	—	—	(7)
52	N5	60	T4	P7_8	I(s)/O	—	—	$\overline{\text{CS2}}$	O	SSISCK1	I(s)/O	DV0_ CLK	I(s)	IRQ3	I(s)	TxD0	O	—	—	—	—	—	—	(7)
53	R5	61	P6	P7_9	I(s)/O	—	—	A25	O	SSIWS1	I(s)/O	DV0_ VSYNC	I(s)	IRQ5	I(s)	SCK3	I(s)/O	TIOC1A	I(s)/O	—	—	—	—	(7)
54	P5	62	U4	P7_10	I(s)/O	—	—	TEND0	O	SSITxD1	O	DV0_ HSYNC	I(s)	—	—	RxD3	I(s)	—	—	—	—	—	—	(7)
55	N6	63	P7	P7_11	I(s)/O	—	—	DACK0	O	SSIRxD1	I(s)	CAN_ CLK	I(s)	SCK2	I(s)/O	TxD3	O	AUDIO_ XOUT	O	AUDIO_ XOUT3	O	—	—	(7)
—	—	64	T5	P8_12	I(s)/O	—	—	LCD0_ TCON4	O	SPDIF_ IN	I(s)	SSIWS2	I(s)/O	RxD4	I(s)	—	—	—	—	—	—	—	—	(7)
—	—	65	—	Vss																				
—	—	66	R6	P8_13	I(s)/O	—	—	LCD0_ TCON5	O	SPDIF_ OUT	O	SSI DATA2	I(s)/O	TxD4	O	—	—	—	—	—	—	—	—	(7)
—	—	67	—	PVcc																				
56	P6	68	U5	P2_0	I(s)/O	—	—	$\overline{\text{CS3}}$	O	RLIN30 RX*3	I(s)	SPDIF_ IN	I(s)	IRQ7	I(s)	—	—	—	—	—	—	—	—	(7)
57	R6	69	T6	P2_1	I(s)/O	—	—	$\overline{\text{RAS}}$	O	RLIN30 TX*3	O	SPDIF_ OUT	O	IRQ6	I(s)	—	—	—	—	—	—	—	—	(7)
—	—	70	U6	P8_14	I(s)/O	—	—	LCD0_ TCON6	O	ET_COL	I(s)	SD_ CD_0	I(s)	SCK1	I(s)/O	—	—	—	—	—	—	—	—	(7)
—	—	71	R7	P8_15	I(s)/O	—	—	—	—	ET_CRS	I(s)	SD_ WP_0	I(s)	RxD1	I(s)	—	—	—	—	—	—	—	—	(7)
—	—	72	T7	P9_0	I(s)/O	—	—	—	—	ET_MDC	O	SD_ D1_0	I(s)/O	TxD1	O	—	—	—	—	—	—	—	—	(7)
—	—	73	U7	P9_1	I(s)/O	—	—	—	—	ET_ MDIO	I(s)/O	SD_ D0_0	I(s)/O	$\overline{\text{CTS0}}$	I(s)/O	—	—	—	—	—	—	—	—	(7)
58	—	74	—	Vcc																				
59	P7	75	R8	P2_2	I(s)/O	—	—	$\overline{\text{CAS}}$	O	CAN1RX	I(s)	TIOC0C	I(s)/O	IRQ5	I(s)	—	—	—	—	—	—	—	—	(7)
60	R7	76	T8	P2_3	I(s)/O	—	—	CKE	O	CAN1TX	O	TIOC0D	I(s)/O	—	—	—	—	—	—	—	—	—	—	(7)
61	—	77	—	Vss																				
62	—	78	—	PVcc																				
63	M7	79	R9	P0_0	I(s)	MD_ BOOT0	I(s)	—	—	RxD0	I(s)	IRQ4	I(s)	—	—	—	—	—	—	—	—	—	—	(3)
64	N7	80	U8	P0_1	I(s)	MD_ BOOT1	I(s)	—	—	RxD2	I(s)	SSIRxD3	I(s)	$\overline{\text{ADTRG}}$	I(s)	—	—	—	—	—	—	—	—	(3)
65	M8	81	T10	P0_2	I(s)	MD_ CLK	I(s)	—	—	RxD1	I(s)	IRQ7	I(s)	—	—	—	—	—	—	—	—	—	—	(3)
66	M6	82	R10	P0_3	I(s)	MD_ CLKS	I(s)	—	—	RxD3	I(s)	SPDIF_ IN	I(s)	—	—	—	—	—	—	—	—	—	—	(3)
67	P8	83	T9	RTC_X1	I	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	(11)
68	R8	84	U9	RTC_X2	O	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
69	M9	85	P10	PLLVcc																				

chanhvinh.com

176 QFP	176 BGA	208 QFP	233 BGA	Port function/ dedicated function		Mode function		Function 1		Function 2		Function 3		Function 4		Function 5		Function 6		Function 7		Function 8		Simplified Circuit Diagram
No.	No.	No.	No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Figure 1.5
70	R9	86	U11	EXTAL	I	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	(10)
71	P9	87	T11	XTAL	O	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
72	—	88	—	Vss																				
73	—	89	—	Vss																				
74	N9	90	R11	NMI	I(s)	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	(3)
75	—	91	—	Vss																				
76	R10	92	U12	$\overline{\text{RES}}$	I(s)	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	(1)
77	—	93	—	PVcc																				
78	P10	94	T12	P1_8	I(s)	—	—	AN0	I(a)	IRQ0	I(s)	RxD0	I(s)	DV0_ DATA4	I(s)	—	—	—	—	—	—	—	—	(4)
79	N10	95	U13	P1_9	I(s)	—	—	AN1	I(a)	IRQ1	I(s)	RxD1	I(s)	DV0_ DATA5	I(s)	—	—	—	—	—	—	—	—	(4)
80	R11	96	R12	P1_10	I(s)	—	—	AN2	I(a)	IRQ2	I(s)	RxD2	I(s)	DV0_ DATA6	I(s)	—	—	—	—	—	—	—	—	(4)
81	N11	97	T13	P1_11	I(s)	—	—	AN3	I(a)	IRQ3	I(s)	RxD3	I(s)	DV0_ DATA7	I(s)	—	—	—	—	—	—	—	—	(4)
82	P12	98	U14	P1_12	I(s)	—	—	AN4	I(a)	IRQ4	I(s)	ET_ RXD0	I(s)	VIO_D4	I(s)	—	—	—	—	—	—	—	—	(4)
83	P11	99	T14	P1_13	I(s)	—	—	AN5	I(a)	IRQ5	I(s)	ET_ RXD1	I(s)	VIO_D5	I(s)	—	—	—	—	—	—	—	—	(4)
84	N12	100	R13	P1_14	I(s)	—	—	AN6	I(a)	IRQ6	I(s)	ET_ RXD2	I(s)	VIO_D6	I(s)	—	—	—	—	—	—	—	—	(4)
85	R12	101	R14	P1_15	I(s)	—	—	AN7	I(a)	IRQ7	I(s)	ET_ RXD3	I(s)	VIO_D7	I(s)	—	—	—	—	—	—	—	—	(4)
86	P13	102	T15	AVcc																				
87	R13	103	U15	AVss																				
88	R14	104	U16	AVref																				
89	N14	105	N14	BSCANP	I(s)	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	(1)
90	—	106	—	PVcc																				
91	P15	107	T17	AUDIO_ X1	I	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	(10)
92	N15	108	R17	AUDIO_ X2	O	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
93	—	109	—	Vss																				
94	M14	110	P15	P2_4	I(s)/O	—	—	$\overline{\text{WE0}}$/ DQMLL	O	—	—	TIOC4A	I(s)/O	—	—	—	—	—	—	—	—	—	—	(7)
95	—	111	—	Vcc																				
96	M13	112	R16	P2_5	I(s)/O	—	—	$\overline{\text{WE1}}$/$\overline{\text{WE}}$/ DQMLU	O	—	—	TIOC3A	I(s)/O	—	—	—	—	—	—	—	—	—	—	(7)
97	L13	113	M14	$\overline{\text{TRST}}$	I(s)	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	(3)
98	L14	114	N15	JP0_1	I	—	—	TDO	O	—	—	—	—	—	—	—	—	—	—	—	—	—	—	(6)
99	L15	115	P16	JP0_0	I	—	—	TDI	I	—	—	—	—	—	—	—	—	—	—	—	—	—	—	(2)
100	L12	116	M15	TMS	I	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	(2)
101	K14	117	N16	TCK	I	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	(2)
102	—	118	—	Vss																				
103	K13	119	N17	P3_0	I(s)/O	—	—	A1	O	SD_ D2_0	I(s)/O	LCD0_ DATA0	O	ET_ TXCLK	I(s)	—	—	—	—	—	—	—	—	(7)
104	K12	120	M16	P3_1	I(s)/O	—	—	A2	O	SD_ D3_0	I(s)/O	LCD0_ DATA1	O	ET_ TXER	O	—	—	—	—	—	—	—	—	(7)
105	—	121	—	Vcc																				
106	J13	122	L15	P3_2	I(s)/O	—	—	A3	O	SD_ CMD_0	I(s)/O	LCD0_ DATA2	O	ET_ TXEN	O	—	—	—	—	—	—	—	—	(7)
107	—	123	—	Vss																				
108	J15	124	M17	P3_3	I(s)/O	—	—	A4	O	SD_ CLK_0	O	LCD0_ DATA3	O	ET_ RXCLK	I(s)	—	—	—	—	—	—	—	—	(7)
109	—	125	—	PVcc																				
110	J14	126	L16	P3_4	I(s)/O	—	—	A5	O	SD_ D0_0	I(s)/O	LCD0_ DATA4	O	ET_ RXER	I(s)	—	—	—	—	—	—	—	—	(7)
—	—	127	K17	P9_2	I(s)/O	—	—	RSPCK2	I(s)/O	ET_ RXCLK	I(s)	SD_ CLK_0	O	$\overline{\text{RTS0}}$	I(s)/O	TIOC1A	I(s)/O	—	—	—	—	—	—	(7)
—	—	128	—	Vss																				
—	—	129	K15	P9_3	I(s)/O	—	—	SSL20	I(s)/O	ET_ RXER	I(s)	SD_ CMD_0	I(s)/O	SCK0	I(s)/O	TIOC1B	I(s)/O	—	—	—	—	—	—	(7)
—	—	130	—	PVcc																				
111	H15	131	K16	P3_5	I(s)/O	—	—	A6	O	SD_ D1_0	I(s)/O	LCD0_ DATA5	O	ET_ RXDV	I(s)	—	—	—	—	—	—	—	—	(7)
112	H13	132	J15	P3_6	I(s)/O	—	—	A7	O	SD_ WP_0	I(s)	LCD0_ DATA6	O	ET_ COL	I(s)	—	—	—	—	—	—	—	—	(7)
113	H14	133	J16	P3_7	I(s)/O	—	—	A8	O	SD_ CD_0	I(s)	LCD0_ DATA7	O	ET_ $\overline{\text{CRS}}$	I(s)	—	—	—	—	—	—	—	—	(7)
—	—	134	—	Vss																				
—	—	135	H17	P9_4	I(s)/O	—	—	MOSI2	I(s)/O	ET_ RXDV	I(s)	SD_ D3_0	I(s)/O	RxD0	I(s)	TIOC2A	I(s)/O	—	—	—	—	—	—	(7)

chanhvinh.com

176 QFP	176 BGA	208 QFP	233 BGA	Port function/ dedicated function		Mode function		Function 1		Function 2		Function 3		Function 4		Function 5		Function 6		Function 7		Function 8		Simplified Circuit Diagram
No.	No.	No.	No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Figure 1.5
—	—	136	—	PVcc																				
—	—	137	H16	P9_5	I(s)/O	—	—	MISO2	I(s)/O	—	—	SD_ D2_0	I(s)/O	TxD0	O	TIOC2B	I(s)/O	—	—	—	—	—	—	(7)
114	G12	138	H15	P3_8	I(s)/O	—	—	A9	O	—	—	AUDIO_C LK	I(s)	DV0_ DATA8	I(s)	SCK3	I(s)/O	—	—	—	—	—	—	(7)
115	G13	139	H14	P3_9	I(s)/O	—	—	A10	O	—	—	SPDIF_ OUT	O	DV0_ DATA9	I(s)	TxD3	O	—	—	—	—	—	—	(7)
116	—	140	—	Vss																				
117	G15	141	G14	P3_10	I(s)/O	—	—	A11	O	SPBIO 01_0	I(s)/O	TIOC3B	I(s)/O	DV0_ DATA10	I(s)	RxD3	I(s)	—	—	—	—	—	—	(7)
118	G14	142	G17	P3_11	I(s)/O	—	—	A12	O	SPBIO 11_0	I(s)/O	TIOC3A	I(s)/O	DV0_ DATA11	I(s)	—	—	—	—	—	—	—	—	(7)
119	—	143	—	Vcc																				
120	F15	144	G16	P3_12	I(s)/O	—	—	A13	O	SPBIO 21_0	I(s)/O	TIOC3C	I(s)/O	DV0_ DATA12	I(s)	—	—	—	—	—	—	—	—	(7)
121	—	145	—	Vss																				
122	F14	146	G15	P3_13	I(s)/O	—	—	A14	O	SPBIO31_ 0	I(s)/O	TIOC3D	I(s)/O	DV0_ DATA13	I(s)	—	—	—	—	—	—	—	—	(7)
123	—	147	—	PVcc																				
124	F13	148	F17	P3_14	I(s)/O	—	—	A15	O	VIO_CLK	I(s)	SPDIF_ IN	I(s)	DV0_ DATA14	I(s)	SCK1	I(s)/O	AUDIO_ XOUT2	O	—	—	—	—	(7)
125	E14	149	F16	P1_0	I(s)/O(o)	—	—	RIIC0SCL	I(s)/ O(o)	IRQ4	I(s)	ET_ RXD0	I(s)	DV0_ DATA0	I(s)	—	—	—	—	—	—	—	—	(9)
126	E15	150	F15	P1_1	I(s)/O(o)	—	—	RIIC0SDA	I(s)/ O(o)	IRQ5	I(s)	ET_ RXD1	I(s)	DV0_ DATA1	I(s)	—	—	—	—	—	—	—	—	(9)
127	E13	151	E17	P1_2	I(s)/O(o)	—	—	RIIC1SCL	I(s)/ O(o)	IRQ6	I(s)	ET_ RXD2	I(s)	DV0_ DATA2	I(s)	—	—	—	—	—	—	—	—	(9)
128	D15	152	E16	P1_3	I(s)/O(o)	—	—	RIIC1SDA	I(s)/ O(o)	IRQ7	I(s)	ET_ RXD3	I(s)	DV0_ DATA3	I(s)	—	—	—	—	—	—	—	—	(9)
129	D14	153	E15	P1_4	I(s)/O(o)	—	—	RIIC2SCL	I(s)/ O(o)	IRQ0	I(s)	DREQ0	I(s)	VIO_D0	I(s)	—	—	—	—	—	—	—	—	(9)
130	C15	154	D17	P1_5	I(s)/O(o)	—	—	RIIC2SDA	I(s)/ O(o)	IRQ1	I(s)	—	—	VIO_D1	I(s)	—	—	—	—	—	—	—	—	(9)
131	C14	155	D16	P1_6	I(s)/O(o)	—	—	RIIC3SCL	I(s)/ O(o)	IRQ2	I(s)	SSIRxD0	I(s)	VIO_D2	I(s)	—	—	—	—	—	—	—	—	(9)
132	B15	156	C17	P1_7	I(s)/O(o)	—	—	RIIC3SDA	I(s)/ O(o)	IRQ3	I(s)	RxD2	I(s)	VIO_D3	I(s)	—	—	—	—	—	—	—	—	(9)
133	A14	157	A16	P3_15	I(s)/O	—	—	A16	O	VIO_ FLD	I(s)	—	—	DV0_ DATA15	I(s)	TxD1	O	—	—	—	—	—	—	(7)
134	B13	158	A15	P2_6	I(s)/O	—	—	RD/WR	O	SSIRxD3	I(s)	TIOC2A	I(s)/O	—	—	—	—	—	—	—	—	—	—	(7)
135	C12	159	B15	P2_7	I(s)/O	—	—	CS0	O	SSISCK3	I(s)/O	TIOC1A	I(s)/O	IRQ2	I(s)	—	—	—	—	—	—	—	—	(7)
136	A13	160	B14	P2_8	I(s)/O	—	—	RD	O	SSITxD3	O	TIOC0A	I(s)/O	—	—	CAN0TX	O	—	—	—	—	—	—	(7)
137	B12	161	C14	P2_9	I(s)/O	—	—	A0	O	SSIWS3	I(s)/O	SCK0	I(s)/O	IRQ1	I(s)	CAN0RX	I(s)	—	—	—	—	—	—	(7)
138	—	162	—	Vss																				
139	A12	163	A14	USB_X1	I	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	(10)
140	B11	164	B13	USB_X2	O	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
141	—	165	—	USBDP Vcc																				
142	—	166	—	USBDP Vss																				
143	B10	167	B12	DM1	I/O	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
144	A10	168	A12	DP1	I/O	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
145	C10	169	C12	VBUS1	I	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
146	—	170	—	USBD Vcc																				
147	—	171	—	USBD Vss																				
148	—	172	—	USBDP Vcc																				
149	—	173	—	USBDP Vss																				
150	A9	174	A11	DM0	I/O	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
151	B9	175	B11	DP0	I/O	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
152	C9	176	C11	VBUS0	I	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
153	—	177	—	USBD Vcc																				
154	—	178	—	USBD Vss																				
155	B8	179	B10	REFRIN	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
156	—	180	—	USBAP Vss																				
157	D10	181	D12	USBAP Vcc																				
158	C8	182	C10	USBA Vcc																				

chanhvinh.com

176 QFP	176 BGA	208 QFP	233 BGA	Port function/ dedicated function		Mode function		Function 1		Function 2		Function 3		Function 4		Function 5		Function 6		Function 7		Function 8		Simplified Circuit Diagram
No.	No.	No.	No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O	Figure 1.5
159	—	183	—	USBA Vss																				
160	—	184	—	USBU Vcc																				
161	—	185	—	USBU Vss																				
162	—	186	—	Vss																				
163	—	187	—	Vss																				
164	C7	188	A9	P4_0	I(s)/O	—	—	A17	O	VIO_VD	I(s)	TIOC1B	I(s)/O	ET_ MDC	O	$\overline{CTS1}$	I(s)/O	—	—	—	—	—	—	(7)
165	B7	189	C9	P4_1	I(s)/O	—	—	A18	O	VIO_HD	I(s)	TIOC2B	I(s)/O	ET_ MDIO	I(s)/O	$\overline{RTS1}$	I(s)/O	—	—	—	—	—	—	(7)
166	C6	190	A8	P4_2	I(s)/O	—	—	A19	O	SPBIO 20_0	I(s)/O	TRACE DATA2*3	O	—	—	—	—	—	—	—	—	—	—	(7)
167	A6	191	B8	P4_3	I(s)/O	—	—	A20	O	SPBIO 30_0	I(s)/O	TRACE DATA3*3	O	—	—	—	—	—	—	—	—	—	—	(7)
—	—	192	C8	P8_0	I(s)/O	—	—	LCD0_ DATA0	O	ET_ TXD0	O	SSISCK1	I(s)/O	SCK3	I(s)/O	—	—	—	—	—	—	—	—	(7)
—	—	193	—	Vss																				
—	—	194	B7	P8_1	I(s)/O	—	—	LCD0_ DATA1	O	ET_ TXD1	O	SSIWS1	I(s)/O	RxD3	I(s)	—	—	—	—	—	—	—	—	(7)
—	—	195	—	PVcc																				
168	B6	196	C7	P4_4	I(s)/O	—	—	A21	O	SPBCLK_ 0	O	TRACE CLK*3	O	—	—	—	—	—	—	—	—	—	—	(7)
169	A5	197	A6	P4_5	I(s)/O	—	—	A22	O	SPBSSL_ 0	O	TRACE CTL*3	O	—	—	—	—	—	—	—	—	—	—	(7)
—	—	198	B6	P8_2	I(s)/O	—	—	LCD0_ DATA2	O	ET_ TXD2	O	SSITxD1	O	TxD3	O	—	—	—	—	—	—	—	—	(7)
—	—	199	D7	P8_3	I(s)/O	—	—	LCD0_ DATA3	O	ET_ TXD3	O	SSIRxD1	I(s)	—	—	—	—	—	—	—	—	—	—	(7)
—	—	200	C6	P8_4	I(s)/O	—	—	LCD0_ DATA4	O	ET_ TXCLK	I(s)	—	—	$\overline{CTS2}$	I(s)/O	TIOC0A	I(s)/O	—	—	—	—	—	—	(7)
—	—	201	B5	P8_5	I(s)/O	—	—	LCD0_ DATA5	O	ET_ TXER	O	—	—	$\overline{RTS2}$	I(s)/O	TIOC0B	I(s)/O	—	—	—	—	—	—	(7)
170	B5	202	A5	P4_6	I(s)/O	—	—	A23	O	SPBIO 00_0	I(s)/O	TRACE DATA0*3	O	—	—	—	—	—	—	—	—	—	—	(7)
171	—	203	—	Vss																				
172	C5	204	C5	P4_7	I(s)/O	—	—	A24	O	SPBIO 10_0	I(s)/O	TRACE DATA1*3	O	—	—	—	—	—	—	—	—	—	—	(7)
173	A4	205	A4	P5_0	I(s)/O	—	—	D0	I(s)/O	MMC_D4	I(s)/O	ET_ TXD0	O	DV0_ DATA16	I(s)	LCD0_ TCON0	O	—	—	—	—	—	—	(8)
174	—	206	—	PVcc																				
175	B4	207	B4	P5_1	I(s)/O	—	—	D1	I(s)/O	MMC_D5	I(s)/O	ET_ TXD1	O	DV0_ DATA17	I(s)	LCD0_ TCON1	O	—	—	—	—	—	—	(8)
176	A3	208	A3	P5_2	I(s)/O	—	—	D2	I(s)/O	MMC_D6	I(s)/O	ET_ TXD2	O	DV0_ DATA18	I(s)	LCD0_ TCON2	O	—	—	—	—	—	—	(8)

[Legend]

(s): Schmitt

(a): Analog

(o): Open drain

Note: • Pins to which the PVcc, Vcc, and Vss functions can be allocated on 176-pin BGA products are listed below.
PVcc: D4, D11, D13, E4, E12, F12, K4, L4, M5, M10, M11, P1
Vcc: A2, B3, C4, D5, D6, D7, M12, N13, P14, R15
Vss: A1, A7, A8, A11, A15, B2, B14, C3, C11, C13, D8, D9, D12, G4, H4, H12, J12, K15, L1, M4, M15, N3, N8, P2, R1, R3

Note: • Pins to which the PVcc, Vcc, and Vss functions can be allocated on 233-pin BGA products are listed below.
PVcc: B17, C13, C16, D8, D9, D15, E14, F4, F14, G4, L4, M4, N4, P1, P8, P9, P12, P13
Vcc: A2, B3, C4, D5, D6, H4, J4, J14, K14, L14, P14, R15, T16, U17
Vss: A1, A7, A10, A13, A17, B2, B9, B16, C3, C15, D4, D10, D11, D13, D14, G7, G8, G9, G10, G11, H7, H8, H9, H10, H11, J7, J8, J9, J10, J11, J17, K7, K8, K9, K10, K11, L7, L8, L9, L10, L11, L17, M1, P4, P11, P17, R3, T2, U1, U3, U10

Note 1. RZ/A1L only. "-" for the RZ/A1LC.
Note 2. RZ/A1LU only. "-" for the RZ/A1LC.
Note 3. RZ/A1L only. "-" for the RZ/A1LU and RZ/A1LC.
Note 4. RZ/A1LU and RZ/A1LC only. "-" for the RZ/A1L.

chanhvinh.com

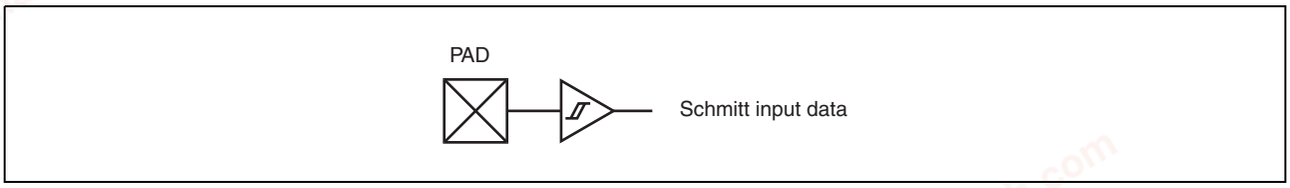


Figure 1.5 (1) Simplified Circuit Diagram (Schmitt Input Buffer)

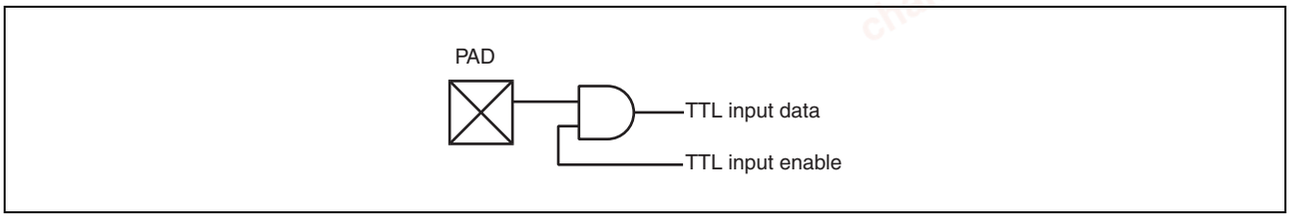


Figure 1.5 (2) Simplified Circuit Diagram (TTL AND Input Buffer)

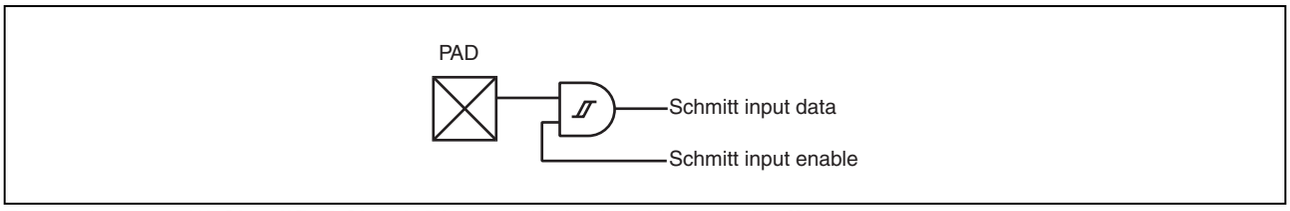


Figure 1.5 (3) Simplified Circuit Diagram (Schmitt AND Input Buffer)

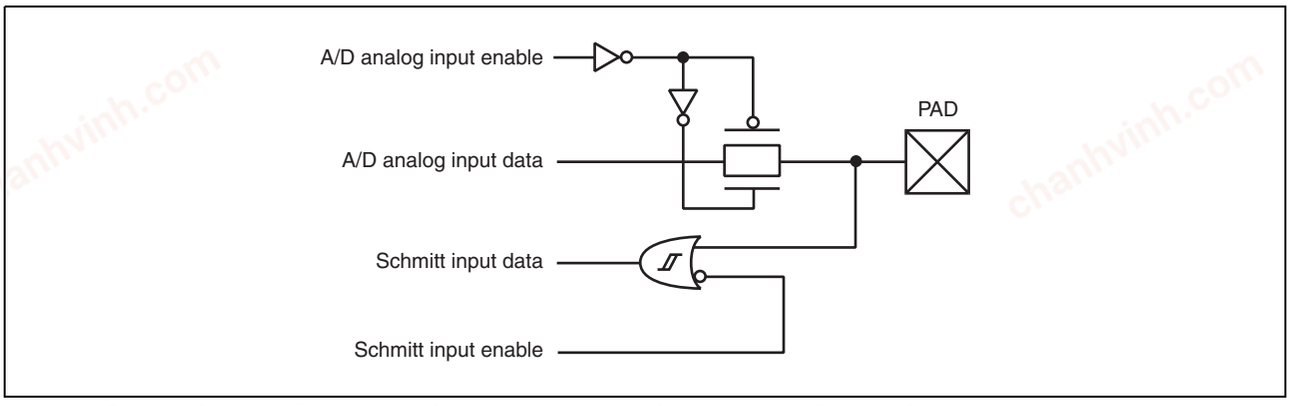


Figure 1.5 (4) Simplified Circuit Diagram (Schmitt OR Input and A/D Input Buffer)

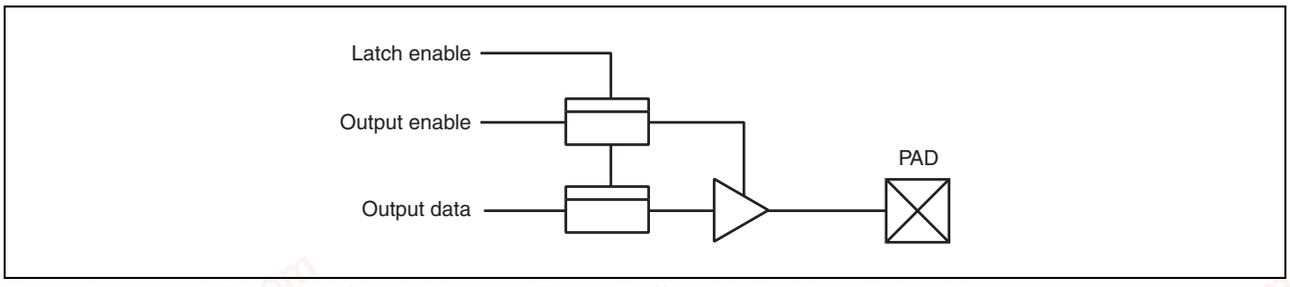


Figure 1.5 (5) Simplified Circuit Diagram (Output Buffer with Enable, with Latch)

chanhvinh.com

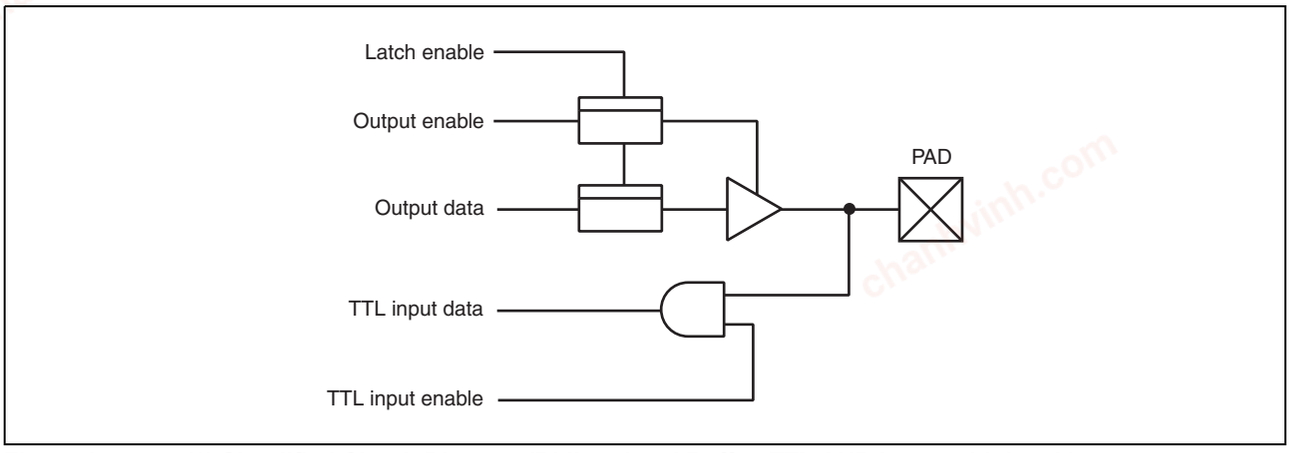


Figure 1.5 (6) Simplified Circuit Diagram (Bidirectional Buffer, TTL AND Input, with Latch)

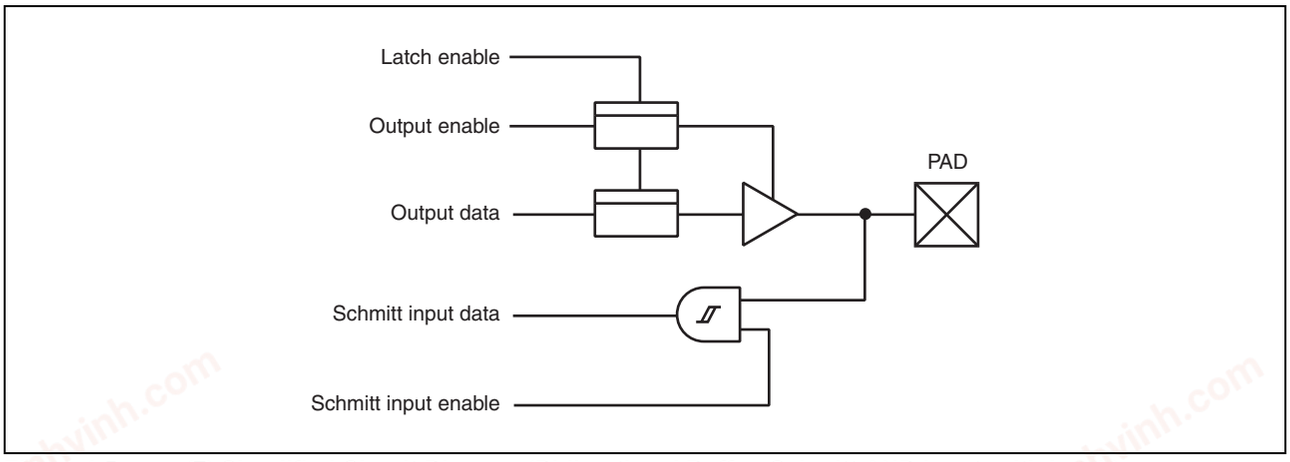


Figure 1.5 (7) Simplified Circuit Diagram (Bidirectional Buffer, Schmitt AND Input, with Latch)

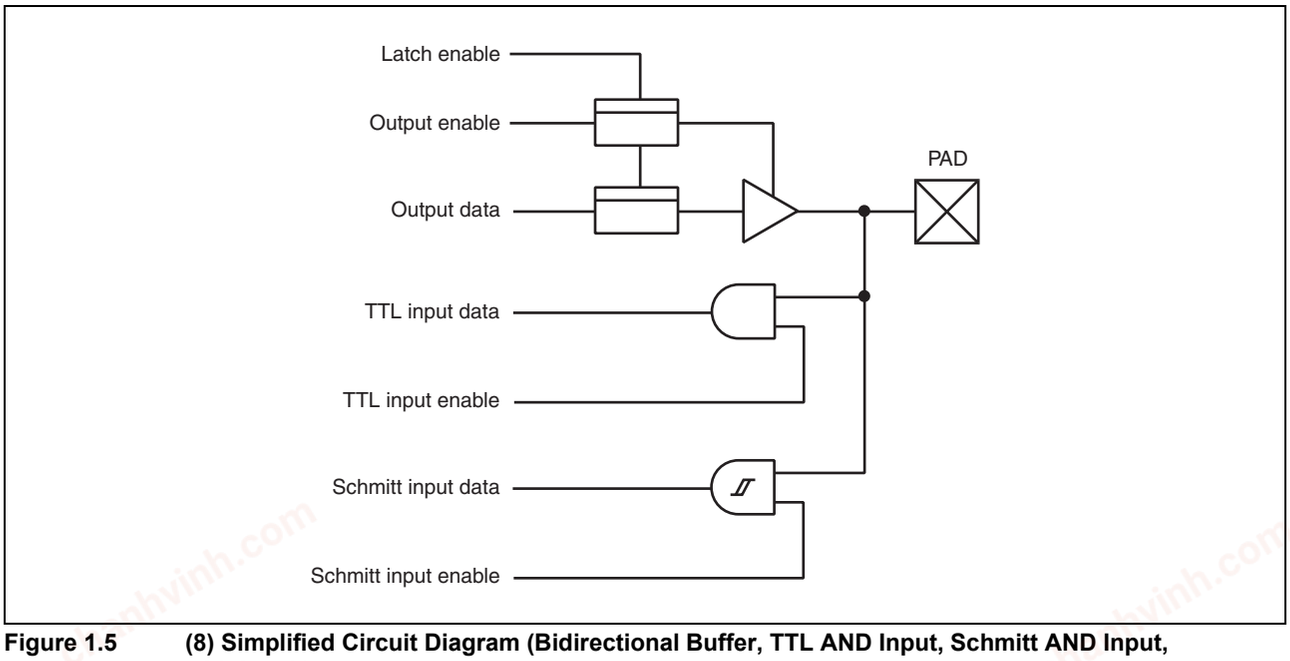


Figure 1.5 (8) Simplified Circuit Diagram (Bidirectional Buffer, TTL AND Input, Schmitt AND Input, with Latch)

chanhvinh.com

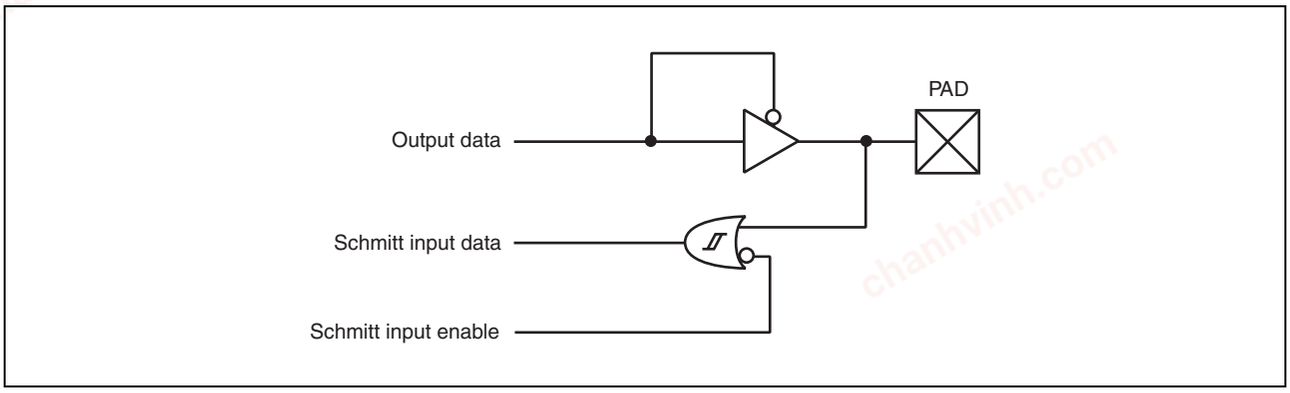


Figure 1.5 (9) Simplified Circuit Diagram (Open Drain Output and Schmitt OR Input Buffer)

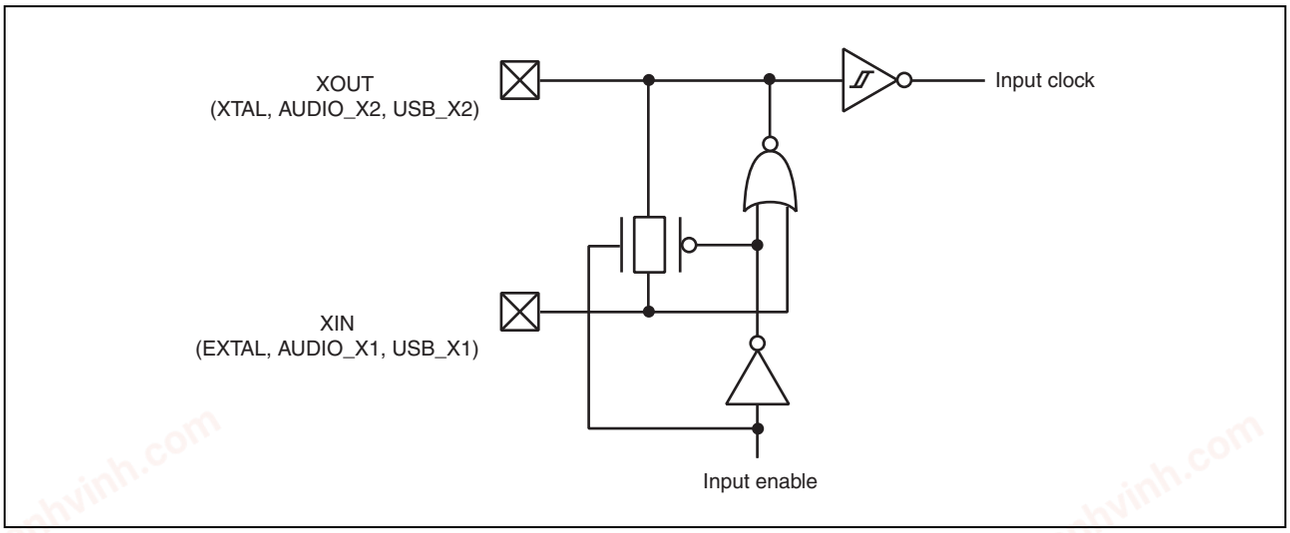


Figure 1.5 (10) Simplified Circuit Diagram (Oscillation Buffer 1)

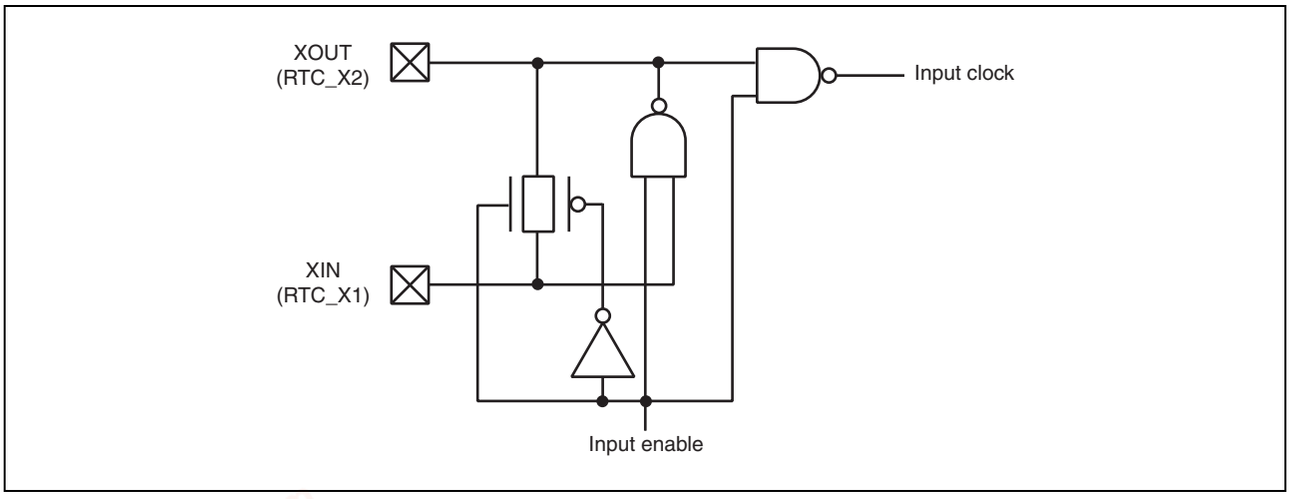


Figure 1.5 (11) Simplified Circuit Diagram (Oscillation Buffer 2)

chanhvinh.com

2. CPU

This product incorporates the Arm single-core Cortex-A9 MPCore, where the IP version is r3p0.

2.1 Features*1

- Instruction cache size: 32 Kbytes
- Data cache size*2: 32 Kbytes
- TLB entries: 128 entries
- Jazelle architecture extension: Full
- Media processing engine with NEON technology: Included
- FPU: Included
- PTM interface: Included
- Wrappers to support for power off and dormant mode: Not included
- Preload engine: Not included
- Number of interrupts: 0 (On-chip interrupt controller is not used.)
- Accelerator Coherence Port: Not included

Note 1. For details, refer to Cortex-A9 MPCore Technical Reference Manual issued by Arm Ltd.
Note 2. Contents of memory regions which are set as write-through are not cached even if data caching is enabled. For details, refer to Cortex-A9 Technical Reference Manual issued by Arm Ltd.

chanhvinh.com

2.2 Configuration Signals

Table 2.1 shows the Cortex-A9 configuration signals and the settings.

Table 2.1 Cortex-A9 Configuration Signal Settings

Configuration Signal	Setting Values
CFGEND	1’b0
CFGNMIF	1’b1
CLUSTERID	4'h0
FILTEREN*1	1’b1
FILTERSTART[31:20]*1	12’hE00
FILTEREND[31:20]*1	12’hFFF
PERIPHBASE[31:13]*2	19’b111_1000_0000_0000_0000
TEINIT	1’b0
VINITHI	NOR, SRAM boot: 1’b0; on-chip ROM boot: 1’b1

Note 1. Do not change the initial settings of these signals by software.
Note 2. The base address for the private memory area in the Cortex-A9 processor is H'F0000000. For details and overview of the registers located in the addresses relative to this base address, refer to Cortex-A9 MPCore Technical Reference Manual issued by Arm Ltd.

chanhvinh.com

3. Boot Mode

This LSI can be booted from the memory connected to the CS0 space and the serial flash memory.

Note 1. Booting from the NAND flash memory with an SD controller and the NAND flash memory with an MMC controller are not currently supported.

3.1 Features

- Four boot modes
  Boot mode 0: Boots the LSI from the memory (bus width: 16 bits) connected to the CS0 space
  Boot mode 1: Boots the LSI from the serial flash memory connected to the SPI multi I/O bus space
  Boot mode 2: Boots the LSI from the NAND flash memory with the SD controller*1
  Boot mode 3: Boots the LSI from the NAND flash memory with the MMC controller*2

Note 1. It is possible to boot the LSI from the embedded SD (eSD) defined by the SD specification part 1 eSD addendum version 2.10 standard.

Note 2. It is possible to boot the LSI from the eMMC device corresponding to the boot operating mode of the JEDEC standard JESD84 A44 (MMCA 4.4) Standard. (It is not possible to boot the LSI from the MMC card.)

3.2 Boot Mode and Pin Function Setting

This LSI can determine the boot mode using external pins when $\overline{\text{RES}}$ is low. The external pin settings for selecting the boot mode are shown in Table 3.1.

Table 3.1 External Pin (MD_BOOT1 and MD_BOOT0) Settings and Corresponding Boot Modes

MD_BOOT1	MD_BOOT0	Boot Mode
0	0	Boot Mode 0 (CS0-space 16-bit booting) Boots the LSI from the memory (bus width: 16 bits) connected to the CS0 space.
1	0	Boot Mode 1 (serial flash booting) Boots the LSI from the serial flash memory connected to the SPI multi I/O bus space.
0	1	Boot Mode 2 (eSD booting) Boots the LSI from the NAND flash memory with the SD controller. The only way of booting this LSI chip is from channel 0 (P3_0 to P3_5) in this mode.
1	1	Boot Mode 3 (eMMC booting) Boots the LSI from the NAND flash memory with the MMC controller.

chanhvinh.com

3.3 Hardware Used in Each Boot Mode

Table 3.2 gives information about the hardware used in each boot mode.

Table 3.2 Hardware Used in Each Boot Mode

Boot Mode	Peripheral Module	Pins Used	Remarks
Boot Mode 0 (CS0-space 16-bit booting)	Bus state controller	A[20:1] D[15:0] $\overline{CS0}$ $\overline{RD}$ CKIO	—
Boot Mode 1 (Serial flash booting)	SPI multi I/O bus controller	SPBCLK_0 SPBSSL_0 SPBMO0_0 SPBMI0_0	The internal baud rate generator generates SPBCLK_0 by dividing Bϕ by 8.
Boot Mode 2 (eSD booting)	SD host interface	SD_CLK_0 SD_CMD_0 SD_D[3:0]_0	The SD clock frequency (SD_CLK_0) is generated by dividing P1ϕ by 4.
Boot Mode 3 (eMMC booting)	MMC host interface	MMC_CLK MMC_CMD MMC_D[3:0]	The MMC clock frequency (MMC_CLK) is generated by diving P1ϕ by 4.

chanhvinh.com

3.4 Exception Vector Address at a Reset in Each Boot Mode

In this LSI, the exception vector address at a reset differs depending on the boot mode.
In this LSI, the exception vector at a reset starts from H'0000_0000 (low vector) in boot mode 0 and from H'FFFF_0000 (high vector) in boot modes 1 to 3.
In this LSI, an on-chip ROM is allocated in area H'FFFF_0000 to H'FFFF_FFFF. The on-chip ROM has a boot program which executes processing corresponding to the boot mode set by the MD_BOOT1 and MD_BOOT0 external pins.
Table 3.3 lists the exception vector address at a reset for each boot mode.

Table 3.3 Exception Vector Address at a Reset in Each Boot Mode

Boot Mode	Exception Vector Address at a Rest	Memory Allocated at the Exception Vector Address
Boot Mode 0 (CS0-space 16-bit booting)	H'0000 0000 (low vector)	Memory connected to the CS0 space
Boot Mode 1 (Serial flash booting)	H'FFFF 0000 (high vector)	On-chip ROM (boot program)
Boot Mode 2 (eSD booting)	H'FFFF 0000 (high vector)	On-chip ROM (boot program)
Boot Mode 3 (eMMC booting)	H'FFFF 0000 (high vector)	On-chip ROM (boot program)

chanhvinh.com

3.5 Operation

3.5.1 Boot Mode 0

In boot mode 0, this LSI is booted from the memory connected to the CS0 space. In these modes, this LSI operates as follows:
After the power-on reset is canceled, program execution is started from H'0000_0000 in the memory connected to the CS0 space.

3.5.2 Boot Mode 1

In boot mode 1, booting up is from the serial flash memory connected to the SPI multi I/O bus space. In this mode, this LSI operates as follows: After the power-on reset is canceled, the boot program stored in the on-chip ROM (starting from H'FFFF_0000) is executed.
The boot program configures the SPI multi I/O bus controller in external address space read mode. With this configuration, this LSI converts reads from the SPI multi I/O bus space to SPI communications and is ready to read directly from the connected serial flash memory. The boot program configures a read command (opcode: 03H, address: 3 bytes, dummy cycle: none) as the command to the serial flash memory used for SPI communication conversion. Figure 3.1 shows the control signals output to the serial flash memory through SPI communication conversion.

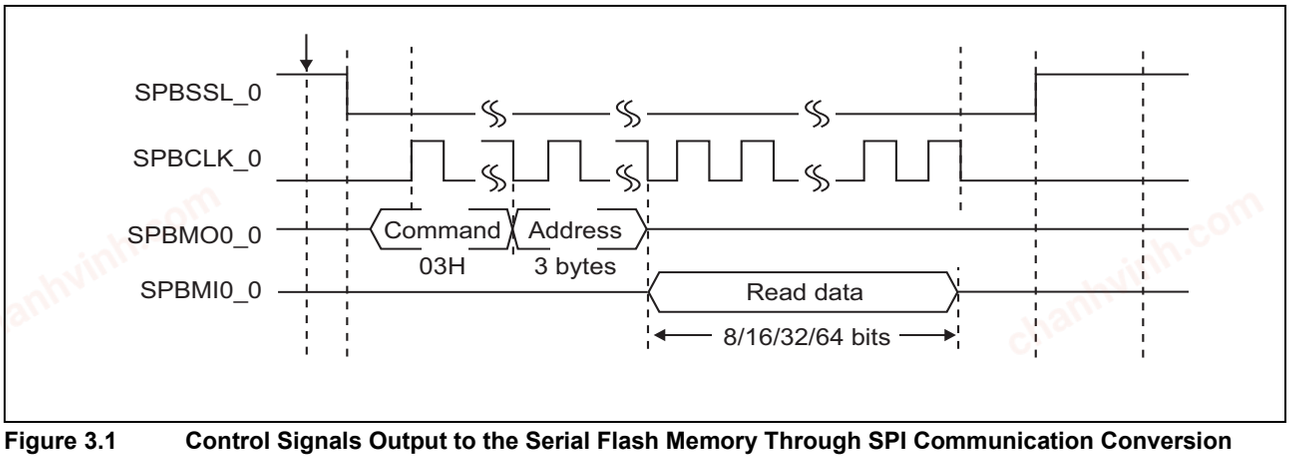


Figure 3.1 Control Signals Output to the Serial Flash Memory Through SPI Communication Conversion

The boot program uses the area at H'2002_0000 to H'2002_3FFF as work memory. It branches to H'1800_0000 (SPI multi I/O bus space) at the end of the processing. At this time, the I bit, F bit, T bit and bits Mode[4:0] in CPSR are set to the initial states with the I bit set to 1'b1 (IRQ masked state), the F bit set to 1'b1 (FIQ masked state), the T bit set to 1'b0 (Arm state) and bits Mode[4:0] set to 5'b10011 (supervisor mode).

chanhvinh.com

3.5.3 Boot Mode 2

In boot mode 2, booting up is from the NAND flash memory with the SD controller, which is connected to channel 0 of the SD host interface. The flow of operation of this LSI in boot mode 4 is as described below.

After a power-on reset, this LSI executes the boot program stored in the on-chip ROM (starting from H'FFFF_0000). The boot program transfers 28 Kbytes of program from the NAND flash memory with the SD controller connected to channel 0 of the SD host interface to the address range from H'2002_4000 to H'2002_AFFF of the large-capacity on-chip RAM.
The program (28 Kbytes) that the boot program transfers to the large-capacity on-chip RAM is called a loader program.

The boot program uses an area from H'2002_0000 to H'2002_3FFF as work memory. Note that the loader program must be stored in the NAND flash memory with the SD controller according to the loader program storage specifications.*1

Note 1. For the storage specifications of the loader program, contact Renesas Electronics Corporation's sales office.

After the boot program finishes processing, it branches to H'2002_4000 (large-capacity on-chip RAM). At this time, the I bit, F bit, T bit and bits Mode[4:0] in CPSR are set to the initial states with the I bit set to 1'b1 (IRQ masked state), the F bit set to 1'b1 (FIQ masked state), the T bit set to 1'b0 (Arm state) and bits Mode[4:0] set to 5'b10011 (supervisor mode).
The size of the loader program that the boot program transfers to the large-capacity on-chip RAM is fixed to 28 Kbytes. If your loader program exceeds this limit, use the loader program to transfer your program (application program) from the NAND flash memory with the SD controller to the large-capacity on-chip or external RAM using channel 0 of the SD host interface. Note that you must design a loader program.

chanhvinh.com

Figure 3.2 is a schematic view of the specification for boot mode 2.

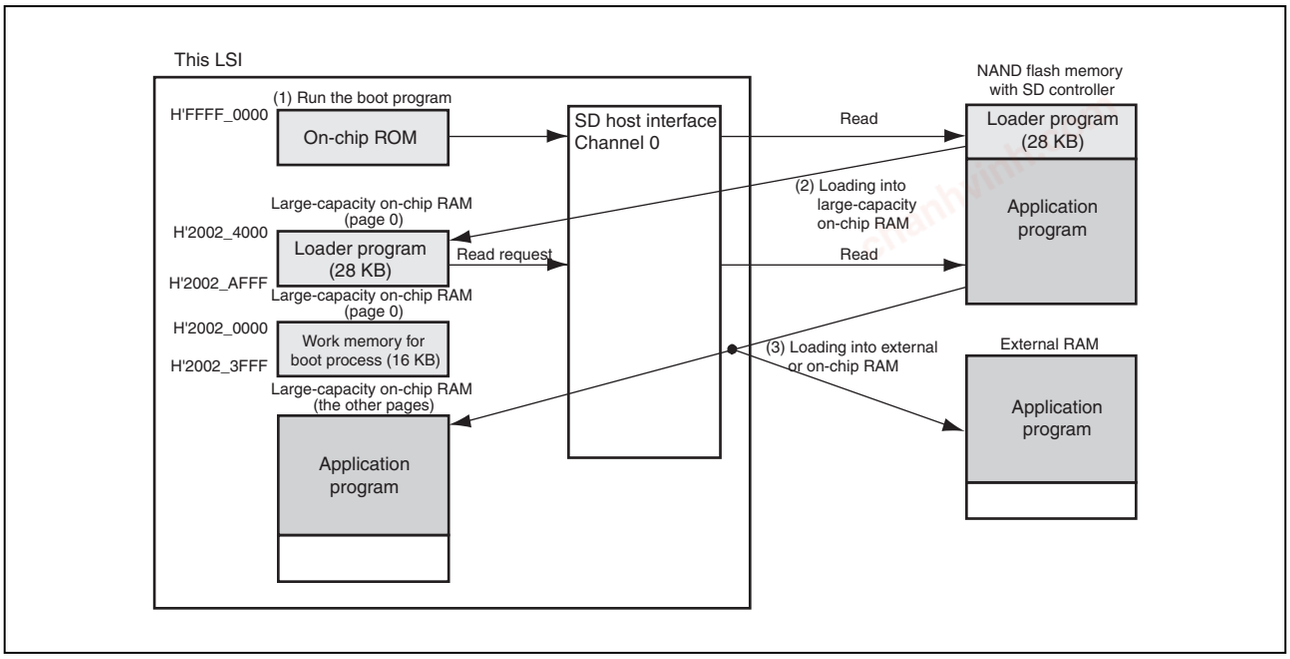


Figure 3.2 Schematic View of Specification for Boot Mode 2

chanhvinh.com

3.5.4 Boot Mode 3

In boot mode 3, booting up is from the NAND flash memory with the MMC controller, which is connected to the MMC host interface. The flow of operation of this LSI in boot mode 5 is as described below.

After a power-on reset, this LSI executes the boot program stored in the on-chip ROM (starting from H'FFFF_0000). The boot program transfers 28 Kbytes of program from the NAND flash memory with the MMC controller connected to the MMC host interface to the address range from H'2002_4000 to H'2002_AFFF of the large-capacity on-chip RAM. The program (28 Kbytes) that the boot program transfers to the large-capacity on-chip RAM is called a loader program.

The boot program uses an area from H'2002_0000 to H'2002_3FFF as work memory. Note that the loader program must be stored in the NAND flash memory with the MMC controller according to the loader program storage specifications.*1

Note 1. For the storage specifications of the loader program, contact Renesas Electronics Corporation's sales office.

After the boot program finishes processing, it branches to H'2002_4000 (large-capacity on-chip RAM). At this time, the I bit, F bit, T bit and bits Mode[4:0] in CPSR are set to the initial states with the I bit set to 1'b1 (IRQ masked state), the F bit set to 1'b1 (FIQ masked state), the T bit set to 1'b0 (Arm state) and bits Mode[4:0] set to 5'b10011 (supervisor mode).
The size of the loader program that the boot program transfers to the large-capacity on-chip RAM is fixed to 28 Kbytes. If your loader program exceeds this limit, use the loader program to transfer your program (application program) from the NAND flash memory with the MMC controller to the large-capacity on-chip or external RAM using the MMC host interface. Note that you must design a loader program.

chanhvinh.com

Figure 3.3 is a schematic view of the specification for boot mode 3.

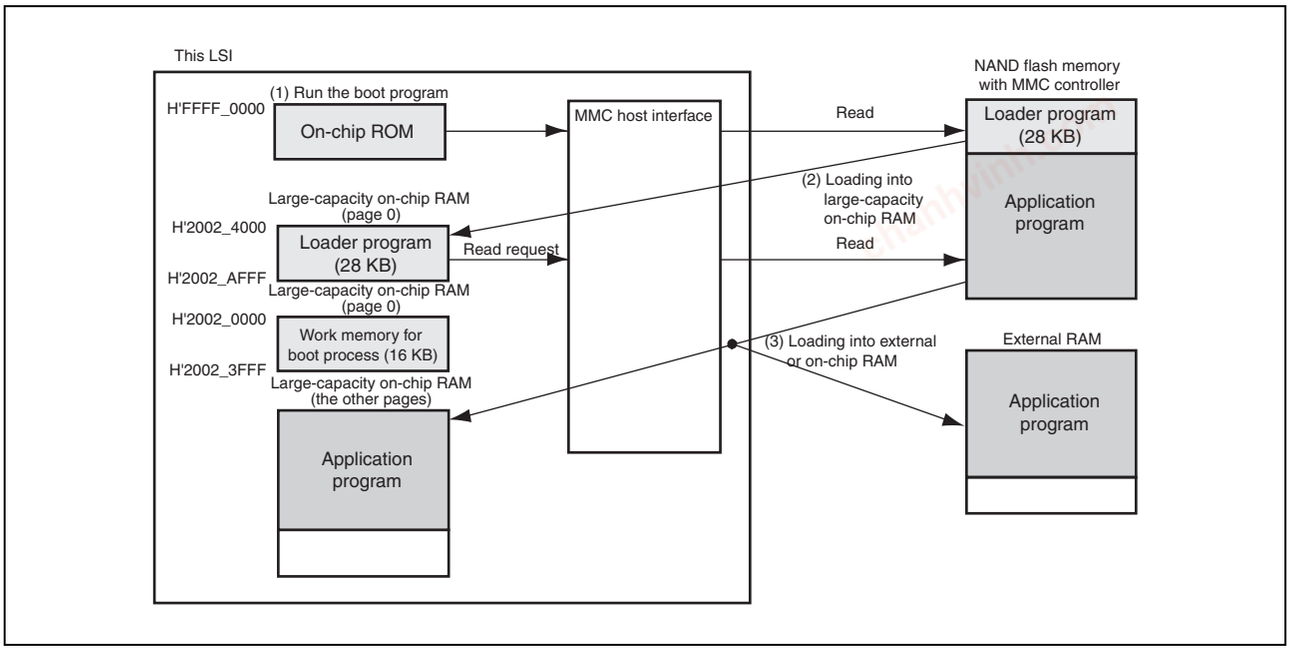


Figure 3.3 Schematic View of Specification for Boot Mode 3

chanhvinh.com

3.6 Notes

3.6.1 Boot Related Pins

The initial states and output states in deep standby mode of the pins related to CS0 space memory read, SPI multi I/O bus space memory read, channel 0 of the SD host interface, and the MMC host interface are different in each boot mode. For details, refer to section 8, Bus State Controller, section 41, Ports, and section 42, Power-Down Modes.

3.6.2 Operation when an Exception Occurs with the Exception Vector Set to the High Vector Address

In this LSI, the program counter loops to its own address (exception vector address) in the on-chip ROM if an exception (except for a reset) occurs when the exception vector is set to the high vector address. In boot mode 1, 2, or 3, set the V bit in SCTLR to 0 to set the exception vector to the low vector address before an exception (except for a reset) occurs. For the details about the CP15 system control register (SCTLR), refer to the Arm Architecture Reference Manual.

3.6.3 Notes on Serial Flash Booting (Boot Mode 1) after This LSI is Reset

In booting up from serial flash memory (boot mode 1), read commands (opcode: 03H, address: 3 bytes, dummy cycles: none) are set for sending to the serial flash memory. Therefore, if this chip enters the reset state while the serial flash memory cannot accept read commands, correct booting up of the chip may not be possible. For example, if the chip is reset while the serial flash memory is being erased (placing it in the busy state), the serial flash memory will not accept read commands. In such cases, that is, in system configurations where the chip may be reset while the serial flash memory is unable to accept read commands, ensure that the serial flash memory is able to accept read commands after the chip is released from the reset state by using serial flash memory that has its own reset pin or cutting off power to the serial flash memory when a reset occurs.

chanhvinh.com

4. Secondary Cache

This product incorporates Arm's PL310 as a secondary cache. The IP version is r3p2.

4.1 Features

- Total cache size: 128 Kbytes
- Number of cache ways: 8 ways
- Number of master ports : 2
- Number of slave ports: 2
- Lockdown by master: No
- Lockdown by line Defined
- Speculative read: No
- Sideband signal from CA9: No

For details, see CoreLink™ Level 2 Cache Controller L2C-310 Technical Reference Manual issued by Arm Ltd.

4.2 Configuration Signals

The setting values of the configuration signals are shown in Table 4.1.

Table 4.1 Setting Values of Configuration Signals

Configuration Signals	Setting Values
ASSOCIATIVITY*1	1'b0 (8 ways)
CACHEID[5:0]	6'b000000
CFGADDRFILTEN*1	1'b1
CFGADDRFILTEND[11:0]*1	12'h3FF
CFGADDRFILTSTART[11:0]*1	12'h180
CFGBIGEND	1'b0
DATAREADLAT[2:0]*1	3'b000
DATASETUPLAT[2:0]*1	3'b000
DATAWRITELAT[2:0]*1	3'b000
REGFILEBASE[19:0]*2	20'h3FFFF
TAGREADLAT[2:0]*1	3'b000
TAGSETUPLAT[2:0]*1	3'b000
TAGWRITELAT[2:0]*1	3'b000
WAYSIZE[2:0]*1	3'b001 (16 Kbytes)

Note 1. Do not change the initial settings of these signals by software.
Note 2. The base address for the PL310 registers is H'3FFFF000. For the details and overview of the registers, see CoreLink™ Level 2 Cache Controller L2C-310 Technical Reference Manual issued by Arm Ltd. The external ROM/RAM mirror area (0x4000_0000 to 0x5FFF_FFFF) is mirrored before the secondary cache. Accordingly, when a cache maintenance operation is to be executed for the external ROM/RAM mirror area, treat this as the normal external ROM/RAM area in the physical addresses range from 0x0000_0000 to 0x1FFF_FFFF.

chanhvinh.com

5. LSI Internal Bus

5.1 LSI Internal Bus

5.1.1 Configuration

This LSI has two main buses: the north main bus where peripheral modules are connected and the south main bus where on-chip RAM and external ROM and RAM are connected. Figure 5.1 is a schematic diagram of the internal buses.

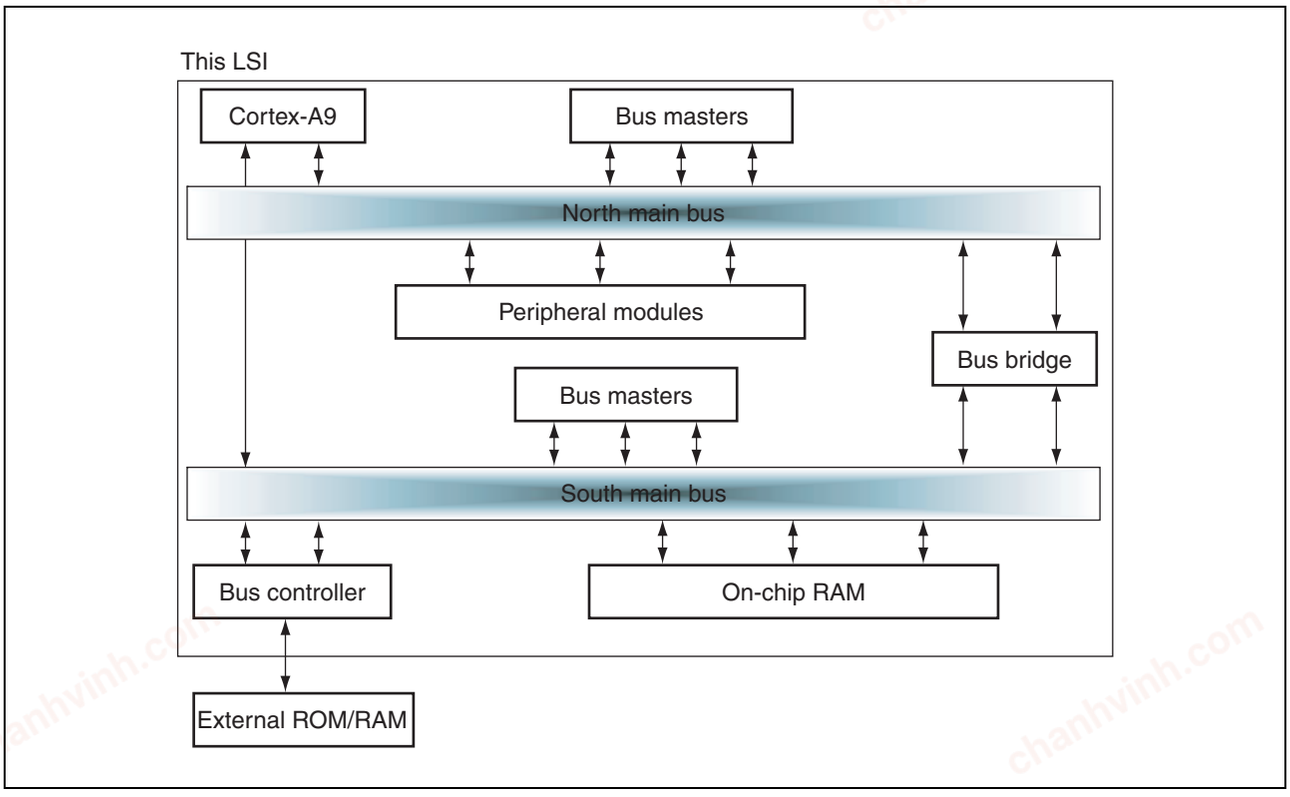


Figure 5.1 Schematic Diagram of LSI Internal Bus

5.1.2 Operation

Cortex-A9 has separate interfaces for the north main bus and south main bus. The addresses assigned to the north main bus are accessed through the north main bus interface, and those assigned to the south main bus are accessed through the south main bus interface.

When a bus master connected to the north main bus, except for Cortex-A9, accesses the on-chip RAM or external ROM or RAM, access is executed through the bus bridge for access from the north main bus to the south main bus. The bus masters connected to the south main bus cannot access an address assigned to the north main bus. The internal bus of this LSI operates in little endian.

chanhvinh.com

5.2 North Main Bus

5.2.1 Configuration

Various peripheral modules are connected to the north main bus. Figure 5.2 shows the configuration of the north main bus.

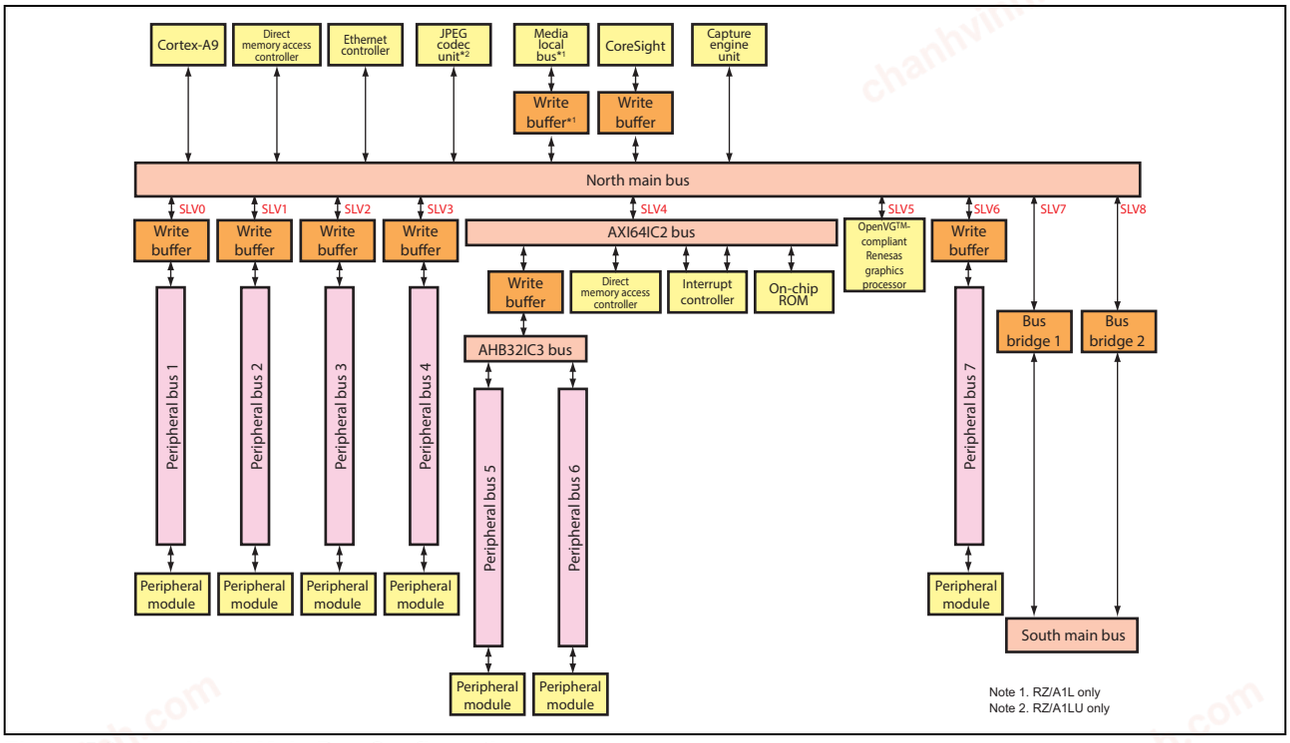


Figure 5.2 North Main Bus Configuration

5.2.2 Features

Table 5.1 shows the features of the north main bus.

Table 5.1 North Main Bus

Item	Description
Bus protocol	AMBA® AXI protocol
Bus system configuration	AXI interconnect with multi-layer configuration for all channels
Bus clock frequency	Bφ
Bus width	64 bits
Arbitration	Round robin

chanhvinh.com

5.2.3 Peripheral Buses

Table 5.2 is a list of the peripheral buses connected to the north main bus.

Table 5.2 List of Peripheral Buses

Item	Description
Peripheral bus 1	
Bus clock frequency	P0φ
Bus width	32 bits
Connected peripheral modules	Multi-function timer pulse unit 2
	Realtime clock
	Video display controller 5
Peripheral bus 2	
Bus clock frequency	P0φ
Bus width	32 bits
Connected peripheral modules	Clock pulse generator
	Interrupt controller
	Direct memory access controller
	OS timer channels 0 and 1
	I^2C bus interface channels 0 to 3
	IEBus controller (RZ/A1L only)
	LIN interface (RZ/A1L only)
	General I/O ports
Peripheral bus 3	
Bus clock frequency	P1φ
Bus width	32 bits
Connected peripheral modules	CAN interface
	Media local bus (RZ/A1L only)
	SD host interface channels 0 and 1
	MMC host interface

chanhvinh.com

Table 5.2 List of Peripheral Buses

Item	Description
Peripheral bus 4	
Bus clock frequency	P1φ
Bus width	32 bits
Connected peripheral modules	Serial communication interface with FIFO channels 0 to 4
	Serial communication interface channels 0 and 1
	Renesas serial peripheral interface channels 0 to 2
	Renesas SPDIF interface
	CD-ROM decoder (RZ/A1L only)
	A/D converter
	USB2.0 host/function module channel 0
	JPEG codec unit (RZ/A1LU only)
AXI64IC2 bus	
Bus protocol	AMBA AXI protocol
Bus system configuration	AXI interconnect with multi-layer configuration for all channels
Bus clock frequency	Bφ
Bus width	64 bits
Arbitration	Round robin
AHB32IC3 bus	
Bus protocol	AMBA AHB protocol
Bus clock frequency	Bφ
Bus width	32 bits
Peripheral bus 5	
Bus clock frequency	P1φ
Bus width	32 bits
Connected peripheral modules	Serial sound interface channels 0 to 3
	USB2.0 host/function module channel 1
	SCUX
Peripheral bus 6	
Bus clock frequency	Bφ
Bus width	32 bits
Connected peripheral modules	Ethernet controller
	Capture engine unit
	EthernetAVB (RZ/A1LU only)
Peripheral bus 7	
Bus clock frequency	P1φ
Bus width	32 bits
Connected peripheral modules	CoreSight

chanhvinh.com

5.3 South Main Bus

5.3.1 Configuration

On-chip RAM and external ROM and RAM are connected to the south main bus. Figure 5.3 shows the configuration of the south main bus.

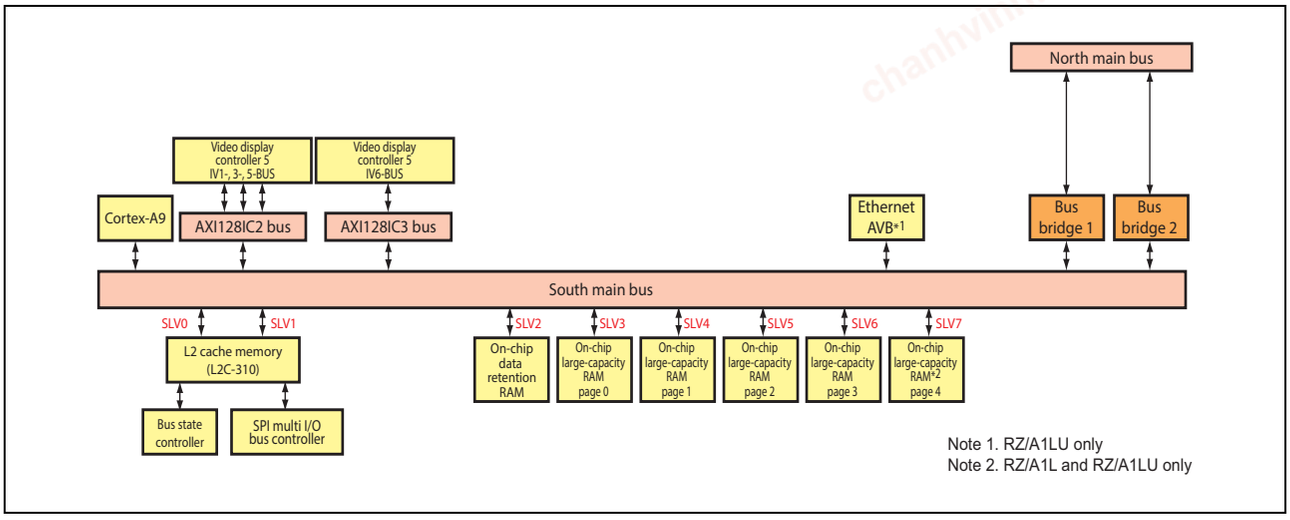


Figure 5.3 South Main Bus Configuration

5.3.2 Features

Table 5.3 shows the features of the south main bus.

Table 5.3 South Main Bus

Item	Description
Bus protocol	AMBA AXI protocol
Bus system configuration	AXI interconnect with multi-layer configuration for all channels
Bus clock frequency	Bφ
Bus width	128 bits
Arbitration	Round robin

chanhvinh.com

5.3.3 Connected Buses

Table 5.4 is a list of the buses connected to the south main bus and their features.

Table 5.4 List of Buses Connected to South Main Bus and their Features

Item	Description
AXI128IC2 and AXI128IC3 buses	
Bus protocol	AMBA AXI protocol
Bus system configuration	AXI interconnect with multi-layer configuration for all channels
Bus clock frequency	Bφ
Bus width	128 bits
Arbitration	Round robin

chanhvinh.com

5.4 Address Map

Table 5.5 shows the address map of this LSI.

Table 5.5 Address Map

Address	Area	Slave Area Viewed from North Main Bus Masters	Slave Area Viewed from South Main Bus Masters
0xFFFF_0000 to 0xFFFF_FFFF	I/O area	ROM in SLV4*1	—
0xFD00_0000 to 0xFFFE_FFFF	Reserved area	—	
0xFCFF_0000 to 0xFCFF_FFFF	I/O area	SLV0 *1	
0xFCFE_0000 to 0xFCFE_FFFF	I/O area	SLV1*1	
0xFC08_0000 to 0xFCFD_FFFF	Reserved area	—	
0xFC00_0000 to 0xFC07_FFFF	I/O area	SLV6*2, *4	
0xF000_2000 to 0xFBFF_FFFF	Reserved area	—	
0xF000_0000 to 0xF000_1FFF	Cortex-A9 private area	—	
0xE823_0000 to 0xEFFF_FFFF	Reserved area	—	
0xE820_0000 to 0xE822_FFFF	I/O area	SLV4 *1, *3	
0xE805_0000 to 0xE81F_FFFF	Reserved area	—	
0xE803_0000 to 0xE804_FFFF	I/O area	SLV2 *1	
0xE802_0000 to 0xE802_FFFF	Reserved area	—	
0xE800_0000 to 0xE801_FFFF	I/O area	SLV3 *1	—
0xE000_0000 to 0xE7FF_FFFF	Reserved area	—	
0x6030_0000 to 0xDFFF_FFFF	Reserved area	—	
0x6020_0000 to 0x602F_FFFF	On-chip large-capacity RAM page 4 mirror area (1 Mbyte) (RZ/A1L and RZ/A1LU only)	SLV7	SLV7
0x6018_0000 to 0x601F_FFFF	On-chip large-capacity RAM page 3 mirror area (512 Kbytes)		SLV6
0x6010_0000 to 0x6017_FFFF	On-chip large-capacity RAM page 2 mirror area (512 Kbytes)	SLV8	SLV5
0x6008_0000 to 0x600F_FFFF	On-chip large-capacity RAM page 1 mirror area (512 Kbytes)		SLV4
0x6002_0000 to 0x6007_FFFF	On-chip large-capacity RAM page 0 (including on-chip data retention RAM) mirror area (512 Kbytes)		SLV3
0x6000_0000 to 0x6001_FFFF			SLV2
0x5C00_0000 to 0x5FFF_FFFF	Reserved area	—	—

chanhvinh.com

Table 5.5 Address Map

Address	Area	Slave Area Viewed from North Main Bus Masters	Slave Area Viewed from South Main Bus Masters
0x5800_0000 to 0x5BFF_FFFF	SPI multi I/O bus area mirror area (64 Mbytes)	SLV7	SLV1
0x5400_0000 to 0x57FF_FFFF	CS5 space mirror area (64 Mbytes)		SLV0
0x5000_0000 to 0x53FF_FFFF	CS4 space mirror area (64 Mbytes)		
0x4C00_0000 to 0x4FFF_FFFF	CS3 space mirror area (64 Mbytes)		
0x4800_0000 to 0x4BFF_FFFF	CS2 space mirror area (64 Mbytes)		
0x4400_0000 to 0x47FF_FFFF	CS1 space mirror area (64 Mbytes)		
0x4080_0000 to 0x43FF_FFFF	CS0 space mirror area (64 Mbytes)		
0x4000_0000 to 0x407F_FFFF		—*5	
0x3FFF_C000 to 0x3FFF_FFFF	I/O area	SLV8	
0x3FEF_B000 to 0x3FFF_BFFF	Reserved area	—	—
0x3FEF_A000 to 0x3FEF_AFFF	I/O area	SLV8	SLV1
0x2030_0000 to 0x3FEF_9FFF	Reserved area	—	—
0x2020_0000 to 0x202F_FFFF	On-chip large-capacity RAM page 4 (1 Mbyte) (RZ/A1L and RZ/A1LU only)	SLV7	SLV7
0x2018_0000 to 0x201F_FFFF	On-chip large-capacity RAM page 3 (512 Kbytes)		SLV6
0x2010_0000 to 0x2017_FFFF	On-chip large-capacity RAM page 2 (512 Kbytes)	SLV8	SLV5
0x2008_0000 to 0x200F_FFFF	On-chip large-capacity RAM page 1 (512 Kbytes)		SLV4
0x2002_0000 to 0x2007_FFFF	On-chip large-capacity RAM page 0 (including on-chip data retention RAM) (512 Kbytes)		SLV3
0x2000_0000 to 0x2001_FFFF			SLV2
0x1C00_0000 to 0x1FFF_FFFF	Reserved area	—	—
0x1800_0000 to 0x1BFF_FFFF	SPI multi I/O bus area (64 Mbytes)	SLV8	SLV1
0x1400_0000 to 0x17FF_FFFF	CS5 space (64 Mbytes)		SLV0
0x1000_0000 to 0x13FF_FFFF	CS4 space (64 Mbytes)		
0x0C00_0000 to 0x0FFF_FFFF	CS3 space (64 Mbytes)		
0x0800_0000 to 0x0BFF_FFFF	CS2 space (64 Mbytes)		
0x0400_0000 to 0x07FF_FFFF	CS1 space (64 Mbytes)		
0x0000_0000 to 0x03FF_FFFF	CS0 space (64 Mbytes)		

chanhvinh.com

Note 1. Only Cortex-A9, the direct memory access controller, and CoreSight can access this area. If any other north main bus master accesses this area, a decode error will occur.
Note 2. Only Cortex-A9 and CoreSight can access this area. If any other north main bus master accesses this area, a decode error will occur.
Note 3. If any address from 0xE821_4800 to 0xE822_FFFF is accessed, a decode error or slave error will occur.
Note 4. A slave error may occur depending on the CoreSight state.
Note 5. Modules which are connected to the north main bus and capable of operating as bus masters are as follows.
- Direct memory access controller
- Ethernet controller
- Media local bus
- CoreSight
- Capture engine unit

These modules do not have access to addresses 0x4000_0000 to 0x407F_FFFF. If they attempt access, a decode error will occur. (RZ/A1L only)
Modules which are connected to the south main bus and capable of operating as bus masters are as follows.
- Cortex-A9
- Video display controller 5

These modules can access the address range shown above.
Note 6. If the on-chip large-capacity RAM is accessed while access is disabled, a slave error will occur.
Note 7. If the area indicated as "—" is accessed, a decode error or a slave error will occur.
Note 8. I/O areas should be accessed in the size specified for each slave module.

chanhvinh.com

5.5 Address Remapping

5.5.1 Overview

Execution in Cortex-A9 jumps to an exception vector placed in addresses 0x0000_0000 to 0x0000_001C when an exception such as a reset or an interrupt occurs. The interrupt response time depends on the time to access the memory connected to this area, and when low-speed memory is connected, the overhead is large. To avoid this, the exception vectors can be remapped to the on-chip high-speed RAM by using the MMU or vector base address register, or the address remapping function can be used to allocate the addresses where the exception vectors are placed to the on-chip high-speed RAM.

Figure 5.4 show the address maps before and after address remapping.

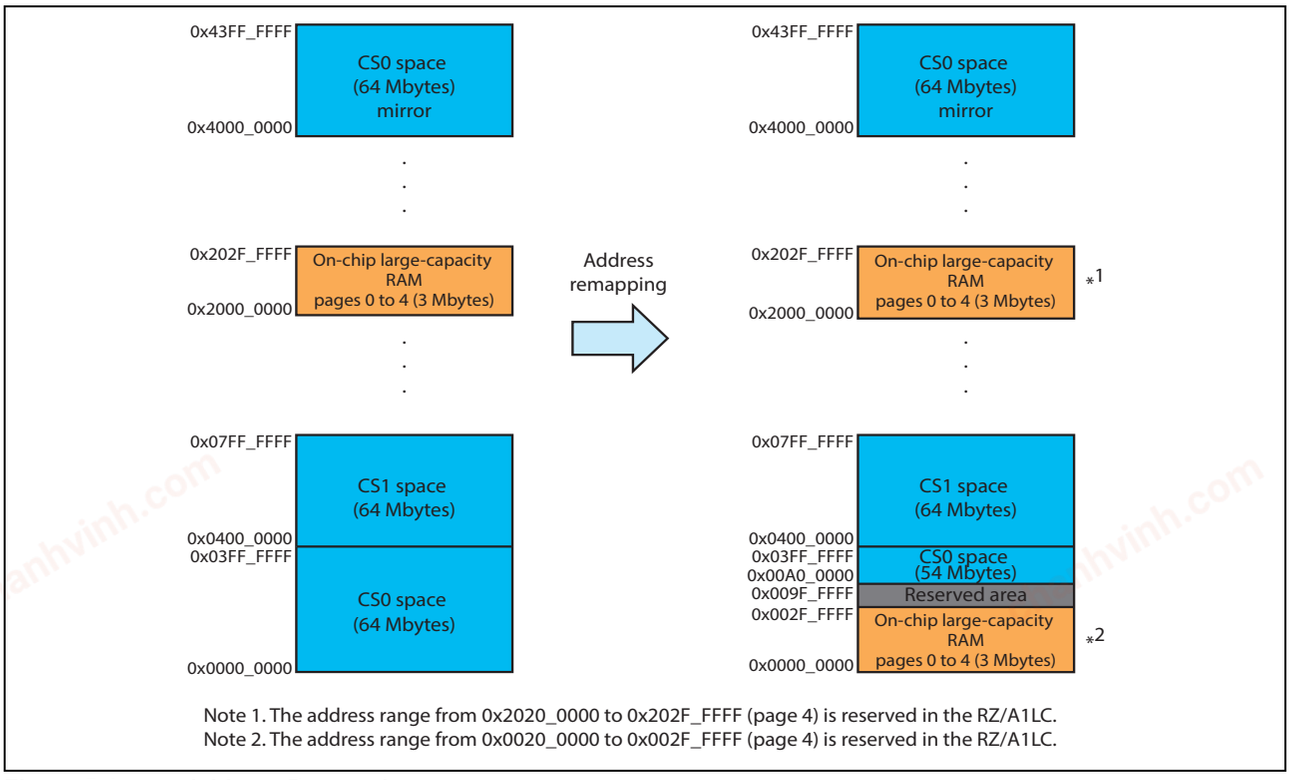


Figure 5.4 Address Remapping

5.5.2 Operation

Addresses are remapped by setting the AXI128 bit in the remap register to 0. After address remapping, pages 0 to 4 (pages 0 to 3 in the RZ/A1LC) of the on-chip large-capacity RAM are allocated to addresses 0x0000_0000 to 0x009F_FFFF. Note that the address range from 0x0030_0000 to 0x009F_FFFF (0x0020_0000 to 0x009F_FFFF in the RZ/A1LC) is reserved. Do not access this area. To access the CS0 space after address remapping, use the mirror area for the CS0 space.

During address remapping, access to addresses 0x0000_0000 to 0x009F_FFFF is prohibited. Accordingly, to modify the remap register, use the following steps.

(1) Stop the bus masters except for Cortex-A9,
or make settings so that addresses 0x0000_0000 to 0x009F_FFFF are never accessed.
(2) Execute a program outside addresses 0x0000_0000 to 0x009F_FFFF.
(3) After modifying the value of the remap register, execute a dummy read of the remap register.

chanhvinh.com

5.6 AXI Interconnect

5.6.1 Configuration

The AXI interconnect in this LSI has a multi-layer configuration in all channels (five channels).
Figure 5.5 shows a conceptual diagram of the AXI interconnect configuration.

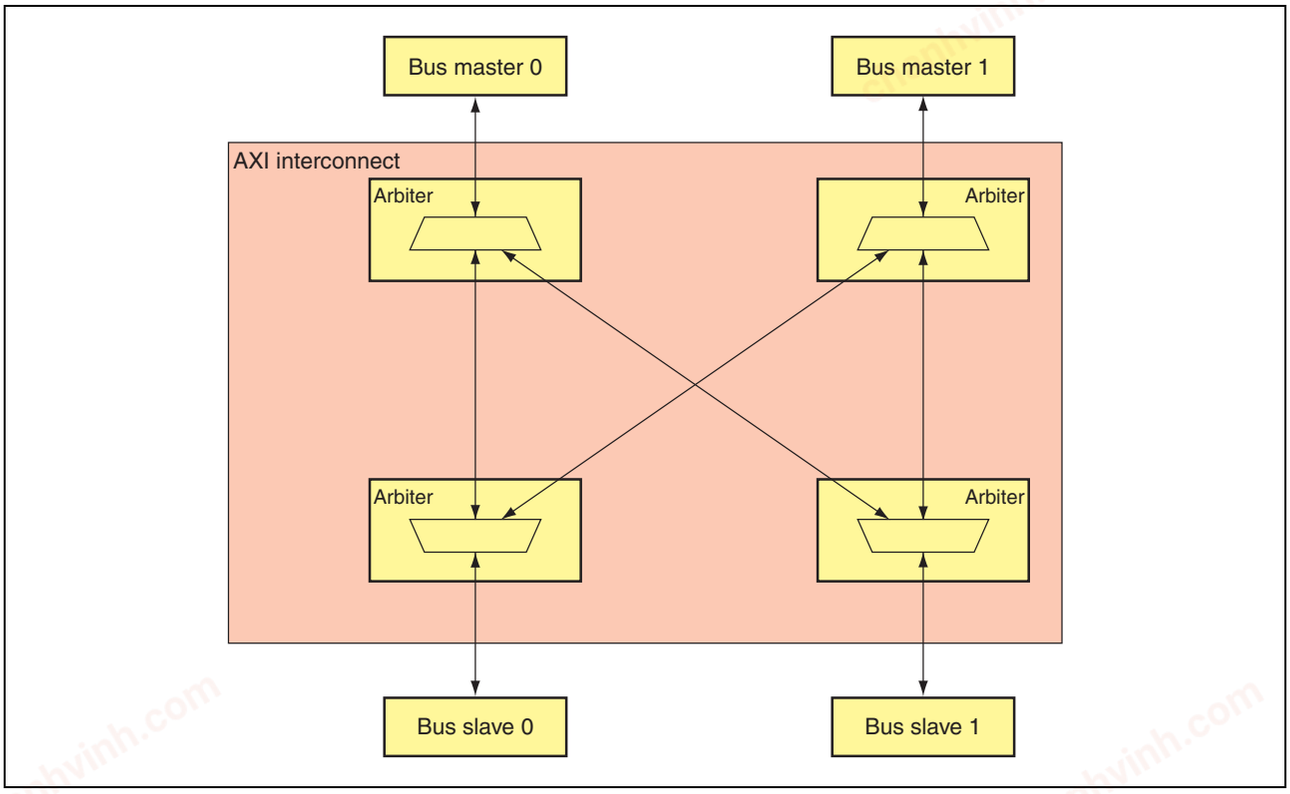


Figure 5.5 Conceptual Diagram of AXI Interconnect Configuration

5.6.2 Operation

In the AXI interconnect, the necessary wiring is prepared for connection between all bus masters and all bus slaves in all channels. When bus masters and slaves access bus slaves and masters, transfer will proceed after bus arbitration by the arbiter. The bus mastership priority changes in a round-robin manner. When multiple bus masters or slaves access different bus slaves or masters, multiple accesses can be executed in parallel. However, when multiple bus masters or slaves access a single bus slave or master at the same time, the bus arbiter executes bus arbitration. When a bus master or slave cannot obtain the bus mastership, it enters a wait state until the bus master or slave that has the bus mastership completes transfer unless the access destination bus master or slave cannot accept multiple transfers. When the destination bus master or slave can accept multiple transfers, bus arbitration is done again with the next transfer timing.

5.7 Bus Bridges

Access from the north main bus to the south main bus is executed through a bus bridge. There are two bus bridges and which bus bridge is used is determined depending on the slave area to be accessed. For assignment of the slave areas to be accessed, see Table 5.5, Address Map.
Each bus bridge can accept up to eight transfers at the same time. Out-of-order transfer is also supported. Therefore, when access to low-speed external ROM and access to on-chip high-speed RAM from different bus masters occur sequentially in this order, the on-chip RAM access can be done without waiting for completion of the previous external ROM access completion.

chanhvinh.com

5.8 AXI Protocol Control Signals

The AXI protocol control signals can be set as desired for each bus master. For details of the AXI protocol control signals, refer to the AMBA AXI Protocol Specification prepared by Arm Ltd.

5.8.1 Bus Masters other than Cortex-A9, CoreSight, and the Direct Memory Access Controller

(1) Cache Control Signals (ARCACHE[3:0] and AWCACHE[3:0])

Use the AXI bus control register (AXIBUSCTL) to make settings of the ARCACHE[3:0] and AWCACHE[3:0] signals for each bus master. Be sure to make settings while the target bus master does not use the AXI bus.

(2) Response Signals (RRESP[1:0] and BRESP[1:0])

Use the AXI bus response error status register (AXIRERRST) to read the RRESP[1:0] and BRESP[1:0] signals received by each bus master. The register value is updated when a response error occurs. The status register value can be cleared to 00 through the AXI bus response error clear register (AXIRERRCLR).
In addition, enabling interrupts through the AXI bus response error interrupt control register (AXIRERRCTL) allows an interrupt to be generated when a response error occurs.
This interrupt should be used only for debugging purposes. Make sure that no response error occurs during system operation.

(3) Protection unit information (ARPROT[2:0], AWPROT[2:0])

Signals ARPROT[2:0] and AWPROT[2:0] are fixed as follows and cannot be modified.
ARPROT[2], AWPROT[2]: 0 (data access)
ARPROT[1], AWPROT[1]: 1*(non-secure access)
ARPROT[0], AWPROT[0]: 0 (normal access)
Note: * For the EthernetAVB, signals ARPROT[1] and AWPROT[1] are fixed to 0 (secure access).

(4) Atomic access (ARLOCK[1:0], AWLOCK[1:0])

Signals ARLOCK[1:0] and AWLOCK[1:0] are fixed as follows and cannot be modified.
ARLOCK[1:0], AWLOCK[1:0]: 00 (normal access)

5.8.2 Cortex-A9

For details on the Cortex-A9, refer to the Arm Architecture Reference Manual.

5.8.3 CoreSight

For details on CoreSight, refer to the technical reference manual issued by Arm Ltd.
The bus master side (AHB access port) of CoreSight is connected to the main bus via the AHB-AXI bus conversion circuit.
The signals are converted as follows for connection to the AXI bus.

(1) Cache control (ARCACHE[3:0], AWCACHE[3:0])

ARCACHE[3], AWCACHE[3]: 0 when HPROT[3] is 0, 1 when HPROT[3] is 1.
ARCACHE[2], AWCACHE[2]: 0 when HPROT[3] is 0, 1 when HPROT[3] is 1.
ARCACHE[1], AWCACHE[1]: Value of HPROT[3] (cacheable)
ARCACHE[0], AWCACHE[0]: Value of HPROT[2] (bufferable)

chanhvinh.com

(2) Response unit (RRESP[1:0], BRESP[1:0])

OKAY is returned when RRESP[1:0] and BRESP[1:0] are 00 or 01.
ERROR is returned when RRESP[1:0] and BRESP[1:0] are 10 or 11.

(3) Protection unit information (ARPROT[2:0], AWPROT[2:0])

ARPROT[2], AWPROT[2]: Inverse of HPROT[0] (data/opcode)
ARPROT[1], AWPROT[1]: Fixed to 1 (non-secure access)
ARPROT[0], AWPROT[0]: Value of HPROT[1] (privileged)

(4) Atomic access (ARLOCK[1:0], AWLOCK[1:0])

ARLOCK[1:0], AWLOCK[1:0]: Fixed to 00 (normal access)

5.8.4 Direct Memory Access Controller

For details on the direct memory access controller, refer to section 9, Direct Memory Access Controller.

5.8.5 Slave Area

The control signals are handled as follows by the modules in the slave area.

(1) Cache control (ARCACHE[3:0], AWCACHE[3:0])

The L2 cache memory and write buffer refer to these signals. Other modules in the slave area do not refer to them.

(2) Response unit (RRESP[1:0], BRESP[1:0])

See Table 5.5, Address Map.

(3) Protection unit information (ARPROT[2:0], AWPROT[2:0])

ARPROT[2], AWPROT[2] (instruction/data): The modules in the slave area do not refer to these signals.
ARPROT[1], AWPROT[1] (non-secure/secure): The interrupt controller and L2 cache memory refer to these signals.
Other modules in the slave area do not refer to them.
ARPROT[0], AWPROT[0] (privileged/user): The modules in the slave area do not refer to these signals.

(4) Atomic access (ARLOCK[1:0], AWLOCK[1:0])

This LSI does not support atomic access. Signals ARLOCK[1:0] and AWLOCK[1:0] should be fixed to 00 for normal access by the bus master.*

Note: * This restriction means that instructions for exclusive access (LDREX, STREX, LDREXB, STREXB, LDREXD, STREXD, LDREXH, STREXH) and semaphore instructions (SWP, SWPB) cannot be used by the Cortex-A9 in the internal non-cacheable areas.

5.9 Write Buffers

A write buffer is provided at each connection between the north main bus and a peripheral bus and at each connection between the media local bus and the north main bus, and CoreSight and the north main bus. When the AWCACHE[1:0] cache control signals are set to cache-enabled or buffer-enabled (either of the AWCACHE[1:0] signals is set to 1), the write buffer sends a write completion response to the bus master before accessing the slave area under the write buffer. At this time, even if a slave error response is returned from the slave area to be accessed, it is ignored.

chanhvinh.com

5.10 Register Descriptions

Table 5.6 shows the registers related to the internal bus.

Table 5.6 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Remap register	RMPR	R/W	H'0000_0003	H'FCFE_1A00	32
AXI bus control register 0	AXIBUSCTL0	R/W	H'0000_0000	H'FCFE_1A04	32
AXI bus control register 2	AXIBUSCTL2	R/W	H'0000_0000	H'FCFE_1A0C	32
AXI bus control register 5 (RZ/A1L only)	AXIBUSCTL5	R/W	H'0000_0000	H'FCFE_1A18	32
AXI bus control register 6	AXIBUSCTL6	R/W	H'0000_0000	H'FCFE_1A1C	32
AXI bus control register 7	AXIBUSCTL7	R/W	H'0000_0000	H'FCFE_1A20	32
AXI bus response error interrupt control register 0	AXIRERRCTL0	R/W	H'0000_0000	H'FCFE_1A30	32
AXI bus response error interrupt control register 2	AXIRERRCTL2	R/W	H'0000_0000	H'FCFE_1A38	32
AXI bus response error status register 0	AXIRERRST0	R/W	H'0000_0000	H'FCFE_1A40	32
AXI bus response error status register 2	AXIRERRST2	R/W	H'0000_0000	H'FCFE_1A48	32
AXI bus response error clear register 0	AXIRERRCLR0	R/W	H'0000_0000	H'FCFE_1A50	32
AXI bus response error clear register 2	AXIRERRCLR2	R/W	H'0000_0000	H'FCFE_1A58	32

chanhvinh.com

5.10.1 Remap Register (RMPR)

This register controls the address remapping function.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AXI128	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	AXI128	1	R/W	AXI128 Address Remapping This bit enables or disables allocation of addresses H'0000_0000 to H'009F_FFFF to on-chip RAM pages 0 to 4 (pages 0 to 3 in the RZ/A1LC). 0: Address remapping is enabled. 1: Address remapping is disabled.
0	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.

chanhvinh.com

5.10.2 AXI Bus Control Register 0 (AXIBUSCTL0)

This register controls the cache operation for the JPEG codec unit (RZ/A1LU only) and Ethernet controller.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	JCUARCACHE[3:0]*				—	—	—	—	JCUAWCACHE[3:0]*			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	ETHARCACHE[3:0]				—	—	—	—	ETHAWCACHE[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27 to 24	JCUARCACHE[3:0]*	0000	R/W	ARCACHE[3:0] Signals for JPEG Codec Unit These bits specify the system cache operation when the JPEG codec unit performs read access. The values of these bits are used as the ARCACHE[3:0] signals for the JPEG codec unit. Modify the values of these bits only while the JPEG codec unit does not use the internal bus.
23 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19 to 16	JCUAWCACHE[3:0]*	0000	R/W	AWCACHE[3:0] Signals for JPEG Codec Unit These bits specify the system cache operation when the JPEG codec unit performs write access. The values of these bits are used as the AWCACHE[3:0] signals for the JPEG codec unit. Modify the values of these bits only while the JPEG codec unit does not use the internal bus.
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 8	ETHARCACHE [3:0]	0000	R/W	ARCACHE[3:0] Signals for Ethernet Controller These bits specify the system cache operation when the Ethernet controller performs read access. The values of these bits are used as the ARCACHE[3:0] signals for the Ethernet controller. Modify the values of these bits only while the Ethernet controller does not use the internal bus.
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	ETHAWCACHE [3:0]	0000	R/W	AWCACHE[3:0] Signals for Ethernet Controller These bits specify the system cache operation when the Ethernet controller performs write access. The values of these bits are used as the AWCACHE[3:0] signals for the Ethernet controller. Modify the values of these bits only while the Ethernet controller does not use the internal bus.

Note: * These bits are only present in the RZ/A1LU. For the RZ/A1L and RZ/A1LC, the write value should always be 0.

chanhvinh.com

5.10.3 AXI Bus Control Register 2 (AXIBUSCTL2)

This register controls the cache operation for the capture engine unit.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	CEUAWCACHE[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	CEUAWCACHE [3:0]	0000	R/W	AWCACHE[3:0] Signals for Capture Engine Unit These bits specify the system cache operation when capture engine unit performs write access. The values of these bits are used as the AWCACHE[3:0] signals for capture engine unit. Modify the values of these bits only while capture engine unit does not use the internal bus.

5.10.4 AXI Bus Control Register 5 (AXIBUSCTL5)

This register is only provided in the RZ/A1L.

This register controls the cache operation for the media local bus.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MLBAXCACHE [1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	MLBAXCACHE [1:0]	00	R/W	AWCACHE[3:0] and ARCACHE[3:0] Signals for Media Local Bus These bits specify the system cache operation when the media local bus performs read or write access. The values of these bits are used as the AWCACHE[3:0] and ARCACHE[3:0] signals for the media local bus. The MLBAXCACHE[0] value is used as ARCACHE[0] and AWCACHE[0] without change. When MLBAXCACHE[1] = 0, the ARCACHE[3:1] and AWCACHE[3:1] signals are all set to 0. When MLBAXCACHE[1] = 1, the ARCACHE[3:1] and AWCACHE[3:1] signals are all set to 1. Modify the values of these bits only while the media local bus does not use the internal bus.

chanhvinh.com

5.10.5 AXI Bus Control Register 6 (AXIBUSCTL6)

This register controls the cache operation for video display controller 5.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	VDC501ARCACHE[3:0]				—	—	—	—	VDC501AWCACHE[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	VDC502ARCACHE[3:0]				—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27 to 24	VDC501ARCACHE[3:0]	0000	R/W	ARCACHE[3:0] Signals for Video Display Controller 5 IV3-BUS These bits specify the system cache operation when the IV3-BUS of video display controller 5 performs read access. The values of these bits are used as the ARCACHE[3:0] signals for the IV3-BUS of video display controller 5. Modify the values of these bits only while video display controller 5 does not use the internal bus.
23 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19 to 16	VDC501AWCACHE[3:0]	0000	R/W	AWCACHE[3:0] Signals for Video Display Controller 5 IV1-BUS These bits specify the system cache operation when the IV1-BUS of video display controller 5 performs write access. The values of these bits are used as the AWCACHE[3:0] signals for the IV1-BUS of video display controller 5. Modify the values of these bits only while video display controller 5 does not use the internal bus.
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 8	VDC502ARCACHE[3:0]	0000	R/W	ARCACHE[3:0] Signals for Video Display Controller 5 IV5-BUS These bits specify the system cache operation when the IV5-BUS of video display controller 5 performs read access. The values of these bits are used as the ARCACHE[3:0] signals for the IV5-BUS of video display controller 5. Modify the values of these bits only while video display controller 5 does not use the internal bus.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

5.10.6 AXI Bus Control Register 7 (AXIBUSCTL7)

This register controls the cache operation for video display controller 5.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	VDC504ARCACHE[3:0]				—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 8	VDC504ARCACHE[3:0]	0000	R/W	ARCACHE[3:0] Signals for Video Display Controller 5 IV6-BUS These bits specify the system cache operation when the IV6-BUS of video display controller 5 performs read access. The values of these bits are used as the ARCACHE[3:0] signals for the IV6-BUS of video display controller 5. Modify the values of these bits only while video display controller 5 does not use the internal bus.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

5.10.7 AXI Bus Response Error Interrupt Control Register 0 (AXIRERRCTL0)

This register controls AXI bus response error interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	JCUR ERREN*	—	—	—	ETHR ERREN	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CEUR ERREN	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28	JCURERR EN*	0	R/W	Response Error Interrupt Enable for JPEG Codec Unit Enables or disables interrupt requests when access from the JPEG codec unit generates a response error. 0: Interrupt requests are disabled. 1: Interrupt requests are enabled.
27 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	ETHRERR EN	0	R/W	Response Error Interrupt Enable for Ethernet Controller Enables or disables interrupt requests when access from the Ethernet controller generates a response error. 0: Interrupt requests are disabled. 1: Interrupt requests are enabled.
23 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	CEURERR EN	0	R/W	Response Error Interrupt Enable for Capture Engine Unit Enables or disables interrupt requests when access from capture engine unit generates a response error. 0: Interrupt requests are disabled. 1: Interrupt requests are enabled.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * This bit is only present in the RZ/A1LU. For the RZ/A1L and RZ/A1LC, the write value should always be 0.

chanhvinh.com

5.10.8 AXI Bus Response Error Interrupt Control Register 2 (AXIRERRCTL2)

This register controls AXI bus response error interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	VDC501 RERREN	—	—	—	VDC502 RERREN	—	—	—	—	—	—	—	VDC504 RERREN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28	VDC501 RERREN	0	R/W	Response Error Interrupt Enable for Video Display Controller 5 IV1/3-BUS Enables or disables interrupt requests when access from the IV1-BUS or IV3-BUS of video display controller 5 generates a response error. 0: Interrupt requests are disabled. 1: Interrupt requests are enabled.
27 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	VDC502 RERREN	0	R/W	Response Error Interrupt Enable for Video Display Controller 5 IV5-BUS Enables or disables interrupt requests when access from the IV5-BUS of video display controller 5 generates a response error. 0: Interrupt requests are disabled. 1: Interrupt requests are enabled.
23 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	VDC504 RERREN	0	R/W	Response Error Interrupt Enable for Video Display Controller 5 IV6-BUS Enables or disables interrupt requests when access from the IV6-BUS of video display controller 5 generates a response error. 0: Interrupt requests are disabled. 1: Interrupt requests are enabled.
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

5.10.9 AXI Bus Response Error Status Register 0 (AXIRERRST0)

This register indicates occurrence of AXI bus response errors.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	JCURRESP [1:0]*		JCUBRESP [1:0]*		ETHRRESP [1:0]		ETHBRESP [1:0]		—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	CEUBRESP [1:0]		—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	JCURRESP [1:0]*	00	R	RRESP[1:0] Signals for JPEG Codec Unit These bits indicate the RRESP[1:0] signals received by the JPEG codec unit. The values of these bits are updated when a response error occurs. 00: OKAY 10: SLVERR 11: DECERR
29, 28	JCUBRESP [1:0]*	00	R	BRESP[1:0] Signals for JPEG Codec Unit These bits indicate the BRESP[1:0] signals received by the JPEG codec unit. The values of these bits are updated when a response error occurs. 00: OKAY 10: SLVERR 11: DECERR
27, 26	ETHRRESP[1:0]	00	R	RRESP[1:0] Signals for Ethernet Controller These bits indicate the RRESP[1:0] signals received by the Ethernet controller. The values of these bits are updated when a response error occurs. 00: OKAY 10: SLVERR 11: DECERR
25, 24	ETHBRESP [1:0]	00	R	BRESP[1:0] Signals for Ethernet Controller These bits indicate the BRESP[1:0] signals received by the Ethernet controller. The values of these bits are updated when a response error occurs. 00: OKAY 10: SLVERR 11: DECERR
23 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	CEUBRESP[1:0]	00	R	BRESP[1:0] Signals for Capture Engine Unit These bits indicate the BRESP[1:0] signals received by capture engine unit. The values of these bits are updated when a response error occurs. 00: OKAY 10: SLVERR 11: DECERR
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * These bits are only present in the RZ/A1LU. For the RZ/A1L and RZ/A1LC, these bits are always read as 0. The write value should always be 0.

chanhvinh.com

5.10.10 AXI Bus Response Error Status Register 2 (AXIRERRST2)

This register indicates occurrence of AXI bus response errors.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VDC501RRESP [1:0]		VDC501BRESP [1:0]		VDC502RRESP [1:0]		—	—	—	—	—	—	VDC504RRESP [1:0]		—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	VDC501RRESP [1:0]	00	R	RRESP[1:0] Signals for Video Display Controller 5 IV3-BUS These bits indicate the RRESP[1:0] signals received by the IV3-BUS of video display controller 5. The values of these bits are updated when a response error occurs. 00: OKAY 10: SLVERR 11: DECERR
29, 28	VDC501BRESP [1:0]	00	R	BRESP[1:0] Signals for Video Display Controller 5 IV1-BUS These bits indicate the BRESP[1:0] signals received by the IV1-BUS of video display controller 5. The values of these bits are updated when a response error occurs. 00: OKAY 10: SLVERR 11: DECERR
27, 26	VDC502RRESP [1:0]	00	R	RRESP[1:0] Signals for Video Display Controller 5 IV5-BUS These bits indicate the RRESP[1:0] signals received by the IV5-BUS of video display controller 5. The values of these bits are updated when a response error occurs. 00: OKAY 10: SLVERR 11: DECERR
25 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19, 18	VDC504RRESP [1:0]	00	R	RRESP[1:0] Signals for Video Display Controller 5 IV6-BUS These bits indicate the RRESP[1:0] signals received by the IV6-BUS of video display controller 5. The values of these bits are updated when a response error occurs. 00: OKAY 10: SLVERR 11: DECERR
17 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

5.10.11 AXI Bus Response Error Clear Register 0 (AXIRERRCLR0)

This register clears the AXI bus response error status.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	JCU RRESP CLR*	—	JCU BRESP CLR*	—	ETH RRESP CLR	—	ETH BRESP CLR	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CEU BRESP CLR	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30	JCURRESP CLR*	0	R/W	JCURRESP[1:0] Clear Writing 1 to this bit clears the JCURRESP[1:0] bits to 00. This bit is always read as 0.
29	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
28	JCUBRESP CLR*	0	R/W	JCUBRESP[1:0] Clear Writing 1 to this bit clears the JCUBRESP[1:0] bits to 00. This bit is always read as 0.
27	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
26	ETHRRESP CLR	0	R/W	ETHRRESP[1:0] Clear Writing 1 to this bit clears the ETHRRESP[1:0] bits to 00. This bit is always read as 0.
25	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
24	ETHBRESP CLR	0	R/W	ETHBRESP[1:0] Clear Writing 1 to this bit clears the ETHBRESP[1:0] bits to 00. This bit is always read as 0.
23 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	CEUBRESP CLR	0	R/W	CEUBRESP[1:0] Clear Writing 1 to this bit clears the CEUBRESP[1:0] bits to 00. This bit is always read as 0.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * This bit is only present in the RZ/A1LU. For the RZ/A1L and RZ/A1LC, the write value should always be 0.

chanhvinh.com

5.10.12 AXI Bus Response Error Clear Register 2 (AXIRERRCLR2)

This register clears the AXI bus response error status.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	VDC501 RRESP CLR	—	VDC501 BRESP CLR	—	VDC502 RRESP CLR	—	—	—	—	—	—	—	VDC504 RRESP CLR	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R	R	R	R	R	R	R/W	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30	VDC501 RRESP CLR	0	R/W	VDC501RRESP[1:0] Clear Writing 1 to this bit clears the VDC501RRESP[1:0] bits to 00. This bit is always read as 0.
29	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
28	VDC501 BRESP CLR	0	R/W	VDC501BRESP[1:0] Clear Writing 1 to this bit clears the VDC501BRESP[1:0] bits to 00. This bit is always read as 0.
27	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
26	VDC502 RRESP CLR	0	R/W	VDC502RRESP[1:0] Clear Writing 1 to this bit clears the VDC502RRESP[1:0] bits to 00. This bit is always read as 0.
25 to 19	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
18	VDC504 RRESP CLR	0	R/W	VDC504RRESP[1:0] Clear Writing 1 to this bit clears the VDC504RRESP[1:0] bits to 00. This bit is always read as 0.
17 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

5.11 Interrupt Request

When a decode error or a slave error occurs, an AXI bus response error interrupt request (PRRI) is issued.
An interrupt request is issued when a response is returned from the bus for which interrupt requests are enabled through the AXI bus response error interrupt control register (AXIRERRCTL). To check the response error type, read the AXI bus response error status register (AXIRERRST). To clear the interrupt request, clear the AXI bus response error status register through the AXI bus response error clear register (AXIRERRCLR).
This interrupt should be used only for debugging purposes. Make sure that no response error occurs during system operation.

chanhvinh.com

6. Clock Pulse Generator

This LSI has a clock pulse generator that generates a CPU clock (Iφ), internal bus clock (Bφ), peripheral clock 1 (P1φ), and peripheral clock 0 (P0φ). The clock pulse generator consists of a crystal oscillator, PLL circuits, and divider circuits.

6.1 Features

- Clock types
  A CPU clock (Iφ); an internal bus clock (Bφ); peripheral clock 1 (P1φ = CKIO) for the external bus interface; peripheral clock 0 (P0φ) for the on-chip peripheral modules
- Frequency change function
  The CPU clock frequency can be changed using the PLL (phase locked loop) circuits and divider circuits within this module. The frequency is changed by software using frequency control register (FRQCR) setting.
- Power-down mode control
  The clock can be stopped in sleep mode, software standby mode, and deep standby mode, and specific modules can be stopped using the module standby function. For details on clock control in the power-down modes, see section 42, Power-Down Modes.
- SSCG function
  The CPU's internal PLL (phase locked loop) circuit includes an SSCG (spread spectrum clock generator).
  The SSCG can be used to decrease the peak value of EMI (electromagnetic interference) noise by frequency modulation, that is, by slightly modulating the output frequency.
  The specification of the SSCG for this LSI is as follows.
  —Specification of SSCG
    (1) Modulation waveform (modulation profile): Triangle wave
    (2) Type of spreading: Down-spreading
    (3) Modulation rate: -3.3% (clock mode 0), -3.1% (clock mode 1)
    (4) Modulation frequency: 20.00 to 26.67 kHz (frequency on the EXTAL pin ÷ 500)
    24.00 kHz (frequency on the USB_X1 pin x (1/4) ÷ 500)

chanhvinh.com

Figure 6.1 shows a block diagram of the clock pulse generator.

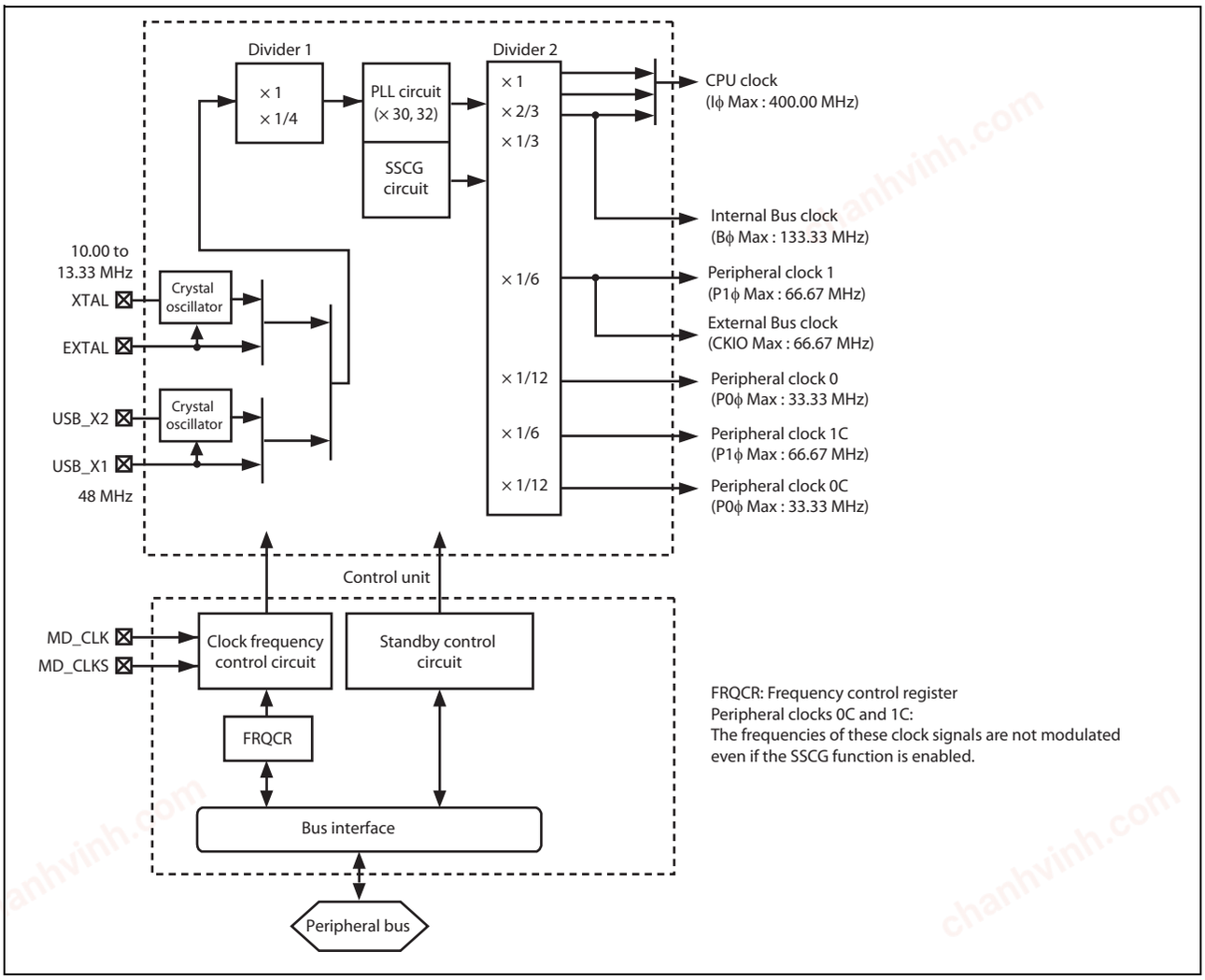


Figure 6.1 Block Diagram

chanhvinh.com

The blocks of this module function as follows:

(1) Crystal Oscillator

A crystal oscillator is connected to the XTAL and EXTAL pins or to the USB_X2 and USB_X1 pins. Either the EXTAL or USB_X1 pin is selected by the clock mode settings.

(2) PLL Circuit

The PLL circuit is capable of multiplying the frequency of the input clock signal from the EXTAL pin by 30. If the input clock signal from the USB_X1 pin is selected, the frequency is multiplied by 32.

(3) Divider 1 and Divider 2

The ratio for frequency division by divider 1 is fixed to 1/1 for the input from the EXTAL pin and 1/4 for the input from the USB_X1 pin.
Divider 2 generates a clock signal whose operating frequency can be used for the CPU clock, internal bus clock, peripheral clock 1, and peripheral clock 0. The division ratio of the CPU clock is set by the frequency control register (FRQCR). The division ratios for peripheral clocks 1 and 0 are fixed to 1/6 and 1/12, respectively.

(4) Clock Frequency Control Circuit

The clock frequency control circuit controls the clock frequency using the frequency control register (FRQCR).

(5) Standby Control Circuit

The standby control circuit controls the states of the on-chip oscillation circuit and other modules during clock switching, or, software standby or deep standby mode.
In addition, the standby control register is provided to control the power-down mode of other modules. For details on the standby control register, see section 42, Power-Down Modes.

(6) Frequency Control Register (FRQCR)

The frequency control register (FRQCR) has control bits assigned for the following functions: clock output/non-output from the CKIO pin during software standby mode or deep standby mode and the frequency division ratio of the CPU clock (Iφ).

(7) SSCG Circuit

Operation of the SSCG circuit is switched on or off (enabled or disabled) by the MD_CLKS pin. When the SSCG function is disabled, all of the internal clock frequencies are fixed, i.e. not modulated. When the SSCG function is enabled, the frequencies of clock signals supplied to peripheral modules other than those listed below are modulated.

Peripheral modules to which non-modulated clock signals are supplied:
IEBus™ controller (RZ/A1L only), multi-function timer pulse unit 2, serial communications interface with FIFO, CAN interface, OS timer, LIN interface (RZ/A1L only), and serial communication interface.

chanhvinh.com

6.2 Input/Output Pins

Table 6.1 lists the clock pulse generator pins and their functions.

Table 6.1 Pin Configuration and Functions of the Clock Pulse Generator

Pin Name	Symbol	I/O	Function
Mode control pin	MD_CLK	Input	Switches between the EXTAL input and the USB_X1 input.
	MD_CLKS	Input	Enables or disables the SSCG circuit.
Crystal input/output pins (clock input pins)	XTAL	Output	Connected to the crystal resonator. (Leave this pin open when the crystal resonator is not in use.)
	EXTAL	Input	Connected to the crystal resonator or used to input external clock.
	USB_X2	Output	Connected to the crystal resonator. (Leave this pin open when the crystal resonator is not in use.)
	USB_X1	Input	Connected to the crystal resonator or used to input external clock.
Clock output pin	CKIO	Output	Clock output pin.

chanhvinh.com

6.3 Clock Mode

Table 6.2 indicates the input/output clock frequency. Table 6.3 shows the usable frequency ranges.

Table 6.2 Input/Output Clock Frequency

Mode	MD_CLK Pin Setting	Clock I/O Source	Output	Divider 1	PLL Circuit	CKIO Frequency
0	0	EXTAL/crystal resonator	CKIO	1	ON (x30)	(EXTAL/crystal resonator) x5
1	1	USB_X1/crystal resonator	CKIO	1/4	ON (x32)	(USB_X1/crystal resonator) x5

In clock mode 0, the clock signal is the input from the EXTAL pin or the crystal oscillator. The PLL circuit shapes waveforms and multiples the frequency, and then supplies the clock to the LSI. The oscillating frequency for the crystal resonator and EXTAL pin input clock ranges from 10 to 13.33 MHz. The frequency range of CKIO is from 50 to 66.67 MHz.

In clock mode 1, the clock signal is the input from the USB_X1 pin or the crystal oscillator. The PLL circuit shapes waveforms and multiples the frequency, and then supplies the clock to the LSI. The oscillating frequency for the crystal resonator and USB_X1 pin input clock is 48 MHz. The frequency of CKIO is 64 MHz.
When changing the frequency, be sure to set the standby_mode_en bit of the power control register in the PL310. For details on the register, see CoreLink Level 2 Cache Controller L2C-310 Technical Reference Manual issued by Arm Ltd.

After the setting of IFC[1:0] in the frequency control register is changed, the hardware automatically stops the bus master and starts changing the frequency following the wait for completion of the issuing-finished request from the bus master. Since processing to change the frequency cannot start if completion of the issuing-finished request is not possible at this time, do not proceed with access to the registers of modules in the module-standby state and so on. Furthermore, as the issuing of unintended requests by the bus master is inhibited, using software to stop all bus masters in preparation for proceeding to change the frequency is also effective.

Table 6.3 Settable Frequency Ranges

Mode	FRQCR Setting *1	PLL Frequency Multiplier: PLL Circuit	Ratio of Internal Clock Frequencies (I : B : P1 : P0)*2	Selectable Frequency Range (MHz): Input Clock*3	Output Clock (CKIO Pin)	CPU Clock (Iφ)	Internal Bus Clock (Bφ)	Peripheral Clock 1 (P1φ)	Peripheral Clock 0 (P0φ)
0	H'x035	ON (× 30)	30 : 10 : 5 : 5/2	10.00 to 13.33	50.00 to 66.67	300.00 to 400.00	100.00 to 133.33	50.00 to 66.67	25.00 to 33.33
	H'x135		20 : 10 : 5 : 5/2			200.00 to 266.67			
	H'x335		10 : 10 : 5 : 5/2			100.00 to 133.33			
1	H'x035	ON (× 32)	8 : 8/3 : 4/3 : 2/3	48.00	64.00	384.00	128.00	64.00	32.00
	H'x135		16/3 : 8/3 : 4/3 : 2/3			256.00			
	H'x335		8/ 3 : 8/3 : 4/3 : 2/3			128.00			

Note 1. x in the FRQCR register setting depends on the set value in bits 12, 13, and 14.
Note 2. The ratio of clock frequencies, where the input clock frequency is assumed to be 1.
Note 3. In clock mode 0, the frequency of the EXTAL pin input clock or the crystal resonator.
In clock mode 1, the frequency of the USB_X1 pin input clock or the crystal resonator.
Caution: Do not use this LSI for frequency settings other than those in Table 6.3.

chanhvinh.com

The clock source of the chip is switched by the setting of the MD_CLK pin while the $\overline{\text{RES}}$ pin is being held low. The following table shows the correspondence between clock source and pin settings.

Table 6.4 Clock Source Selection

MD_CLK Pin Setting	Clock Source
0	EXTAL/crystal resonator
1	USB_X1/crystal resonator

The SSCG function of the chip is switched on or off by the setting of the MD_CLKS pin while the $\overline{\text{RES}}$ pin is being held low. The following table shows the correspondence between SSCG operation and pin settings. Note that the pin setting does not affect the PLL frequency multipliers and division ratios for individual clock signals.

Table 6.5 SSCG Operation Setting

MD_CLKS Pin Setting	SSCG Operation
0	Off
1	On

chanhvinh.com

6.4 Register Descriptions

Table 6.6 shows the register configuration of the clock pulse generator.

Table 6.6 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Frequency control register	FRQCR	R/W	H'0335	H'FCFE0010	16

6.4.1 Frequency Control Register (FRQCR)

FRQCR is a 16-bit readable/writable register used to specify whether a clock is output from the CKIO pin during normal operation mode, change of gain of crystal oscillator for the XTAL pin, software standby mode, deep standby mode, and standby mode cancellation. The register specifies the frequency division ratio for the CPU clock (Iφ). FRQCR can be accessed in 16-bit units.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	CKOEN2	CKOEN[1:0]		-	-	IFC[1:0]		-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	1	1	0	0	1	1	0	1	0	1
R/W:	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	CKOEN2	0	R/W	Clock Output Enable 2 Specifies whether the CKIO pin outputs clock signals or is fixed to the low level when the gain of the crystal oscillator for the XTAL pin is changed. If this bit is set to 1, the CKIO pin is fixed to the low level when the gain of the crystal oscillator for the XTAL pin is changed. Therefore, the malfunction of an external circuit caused by an unstable CKIO clock while changing the gain of the crystal oscillator for the XTAL pin can be prevented. 0: Unstable clock output 1: Low-level output
13, 12	CKOEN[1:0]	00	R/W	Clock Output Enable These bits specify whether the CKIO pin outputs clock signals, or is set to a fixed level or high impedance (Hi-Z) during normal operation mode, deep standby mode, software standby mode, or cancellation of standby mode. If these bits are set to 01, the CKIO pin is fixed at low during deep standby mode, software standby mode, or cancellation of software standby mode. Therefore, the malfunction of an external circuit caused by an unstable CKIO clock during cancellation of software standby mode can be prevented. Table 6.7 lists CKOEN[1:0] settings.
11, 10	—	00	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	IFC[1:0]	11	R/W	CPU Clock Frequency Division Ratio These bits specify the frequency division ratio of the CPU clock with respect to the output frequency of PLL circuit. Note: See section 6.5.1. 00: 1/1 time 01: 2/3 time 10: Reserved (setting prohibited) 11: 1/3 time
7, 6	—	00	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
5, 4	—	11	R	Reserved These bits are always read as 1. The write value should always be 1.
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.

Table 6.7 CKOEN[1:0] Settings

Setting	Normal Operation	Software Standby Mode	Deep Standby Mode*
00	Output	Output off (Hi-Z)	Output off (Hi-Z)
01	Output	Low-level output	Low-level output
10	Output	Output (unstable clock output)	Low-level or high-level output
11	Output off (Hi-Z)	Output off (Hi-Z)	Output off (Hi-Z)

Note: * Note that the first cycle of the output CKIO clock may be missing after release from deep standby.

chanhvinh.com

6.5 Changing the Frequency

The frequency of the CPU clock (Iφ) can be changed by changing the division rate of divider. The division rate can be changed by software through the frequency control register (FRQCR).

6.5.1 Changing the Division Ratio

The division rate of divider can be changed by the following operation.

1. In the initial state, IFC[1:0] = B'11.
2. Set the desired value in the IFC[1:0] bits. Note that if the wrong value is set, this LSI will malfunction.
3. After the register bits (IFC[1:0]) have been set, the clock is supplied of the new division ratio.

Note: When executing the WFI instruction after changing the frequency, be sure to read the frequency control register (FRQCR) to confirm that the new setting is in place and read the ISBUSY0 bit in the CPU status register (CPUSTS) to confirm that it is set to 0 beforehand.
For the CPUSTS register, see section 42, Power-Down Modes.

chanhvinh.com

6.6 Usage of the Clock Pins

For the connection of a crystal resonator or the input of a clock signal, this LSI circuit has the pins listed in Table 6.8. With regard to these pins, take care on the following points. Furthermore, Xin pin and Xout pin are used in this section to refer to the pins listed in the table.

Table 6.8 Clock Pins

Xin Pins (Used for Connection of a Crystal Resonator and Input of External Clock Signals)	Xout Pins (Used for Connection of a Crystal Resonator)
EXTAL	XTAL
USB_X1	USB_X2
AUDIO_X1	AUDIO_X2
RTC_X1	RTC_X2

6.6.1 In the Case of Inputting an External Clock

An example of the connection of an external clock is shown in Figure 6.2. In cases where the Xout pin is left open state, take the parasitic capacitance as less than 10 pF.

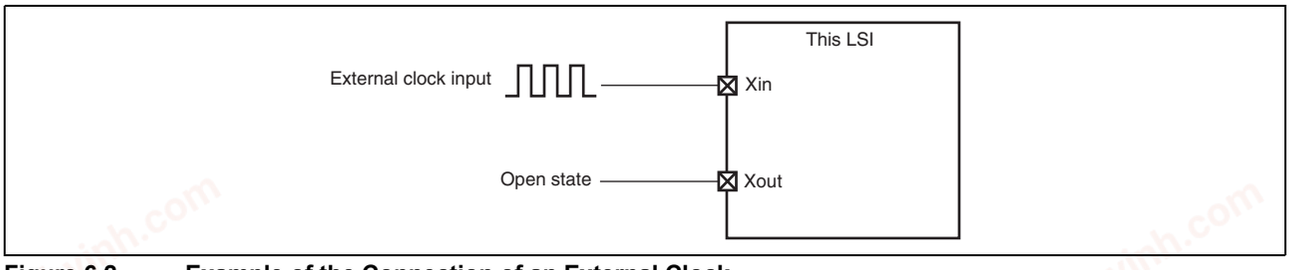


Figure 6.2 Example of the Connection of an External Clock

chanhvinh.com

6.6.2 In the Case of Using a Crystal Resonator

An example of the connection of crystal resonator is shown in Figure 6.3.

Place the crystal resonator and capacitors (CL1 and CL2) as close to pins Xin and Xout as possible. Furthermore, to avoid inductance so that oscillation is correct, use the points where the capacitors are connected to the crystal resonator in common and do not place wiring patterns close to these components.

Since the design of the user board is closely connected with the effective characteristics of the crystal resonator, refer to the example of connection of the crystal resonator that is introduced in this section and perform thorough evaluation on the user side as well. The rated value of the crystal resonator will vary with the floating capacitances and so on of the crystal resonator and mounted circuit, so proceed with decisions on the basis of full discussions with the maker of the crystal resonator. Ensure that voltages applied to the clock pins do not exceed the maximum rated values.

Although the feedback resistor is included in this LSI, an external feedback resistor may be required in some cases. This depends on the characteristics of the crystal resonator.

Set the parameters (of resistors and capacitors) with thorough evaluation on the user side.

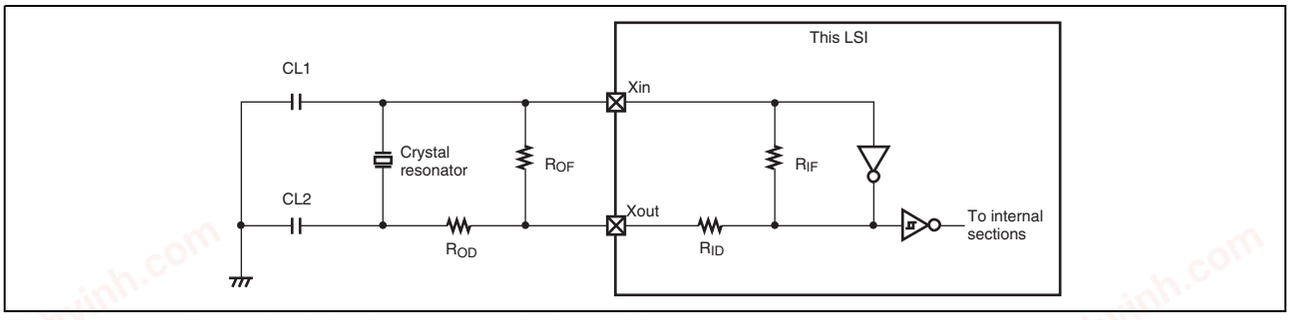


Figure 6.3 Example of the Connection of a Crystal Resonator

6.6.3 In the Case of Not Using the Clock Pin

In cases where the pins are not in use, fix the level on the Xin pin (pull it up or down, or connect it to the power-supply or ground level), and leave the Xout pin open state.

chanhvinh.com

6.7 Oscillation Stabilizing Time

6.7.1 Oscillation Stabilizing Time of the On-chip Crystal Oscillator

In the case of using a crystal resonator, wait for the oscillation stabilizing time of the on-chip oscillation circuit at the following cases, to keep the oscillation stabilizing time of the on-chip crystal oscillator (in the case of inputting an external clock input, it is not necessary).

- Power on
- Releasing the software standby mode or deep standby mode by $\overline{\text{RES}}$ pin
- Changing from halting oscillation to running oscillation by power-on reset or register setting (AUDIO_X1)
- Changing from halting oscillation to running oscillation by register setting (RTC_X1)
- Changing the gain of the on-chip crystal oscillator by $\overline{\text{RES}}$ pin (EXTAL)

6.7.2 Oscillation Stabilizing Time of the PLL circuit

The clock from EXTAL is supplied to the PLL circuit. So, regardless of whether using a crystal resonator or inputting an external clock from EXTAL, wait for at least the oscillation stabilizing time at the following cases, to keep the oscillation stabilizing time of the PLL circuit.

- Power on (in the case of using the crystal resonator)/start inputting external clock (in the case of inputting the external clock)
- Releasing the software standby mode or deep standby mode by $\overline{\text{RES}}$ pin

[Remarks]
The oscillation stabilizing time is kept by the counter running in the LSI at the following cases.

- Releasing the software standby mode or deep standby mode by the other than $\overline{\text{RES}}$ pin
- Changing the gain of the on-chip crystal oscillator by the register setting (EXTAL)

chanhvinh.com

6.8 Notes on Board Design

6.8.1 Note on Using a PLL Oscillation Circuit

In the PLLVcc connection pattern for the PLL, signal lines from the board power supply pins must be as short as possible and pattern width must be as wide as possible to reduce inductive interferences.

Since the analog power supply pins of the PLL are sensitive to the noise, the system may malfunction due to inductive interference at the other power supply pins. To prevent such malfunction, the analog power supply pins and the digital power supply pins Vcc and PVcc should not supply the same resources on the board if at all possible.

chanhvinh.com

6.9 Definition of Modulation Rate and Frequency in the SSCG Specification

The SSCG circuit can be used to decrease the peak value of electromagnetic interference noise by frequency modulation, i.e. by slightly modulating the output frequency. In this case, the rate of change in the frequency and the size of the change to the input clock frequency are defined as the modulation rate and modulation frequency, respectively. Figure 6.4 shows the modulation rate and modulation frequency.

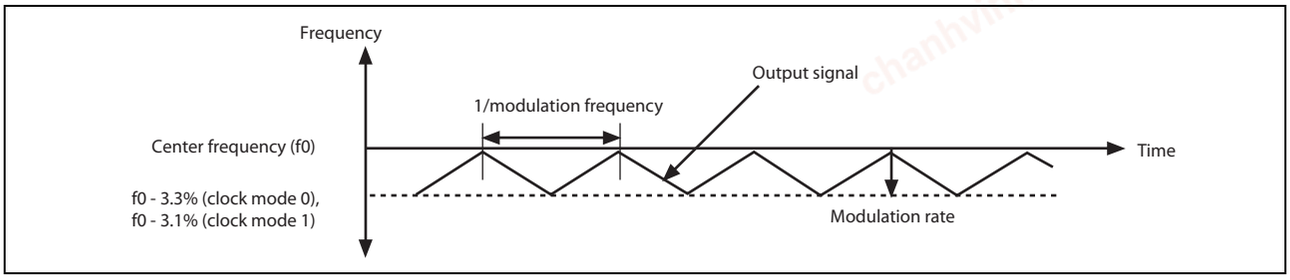


Figure 6.4 Definition of SSCG Modulation Rate and Frequency

chanhvinh.com

6.10 Clock Signals

6.10.1 Clock Signals for the System and Realtime Clock

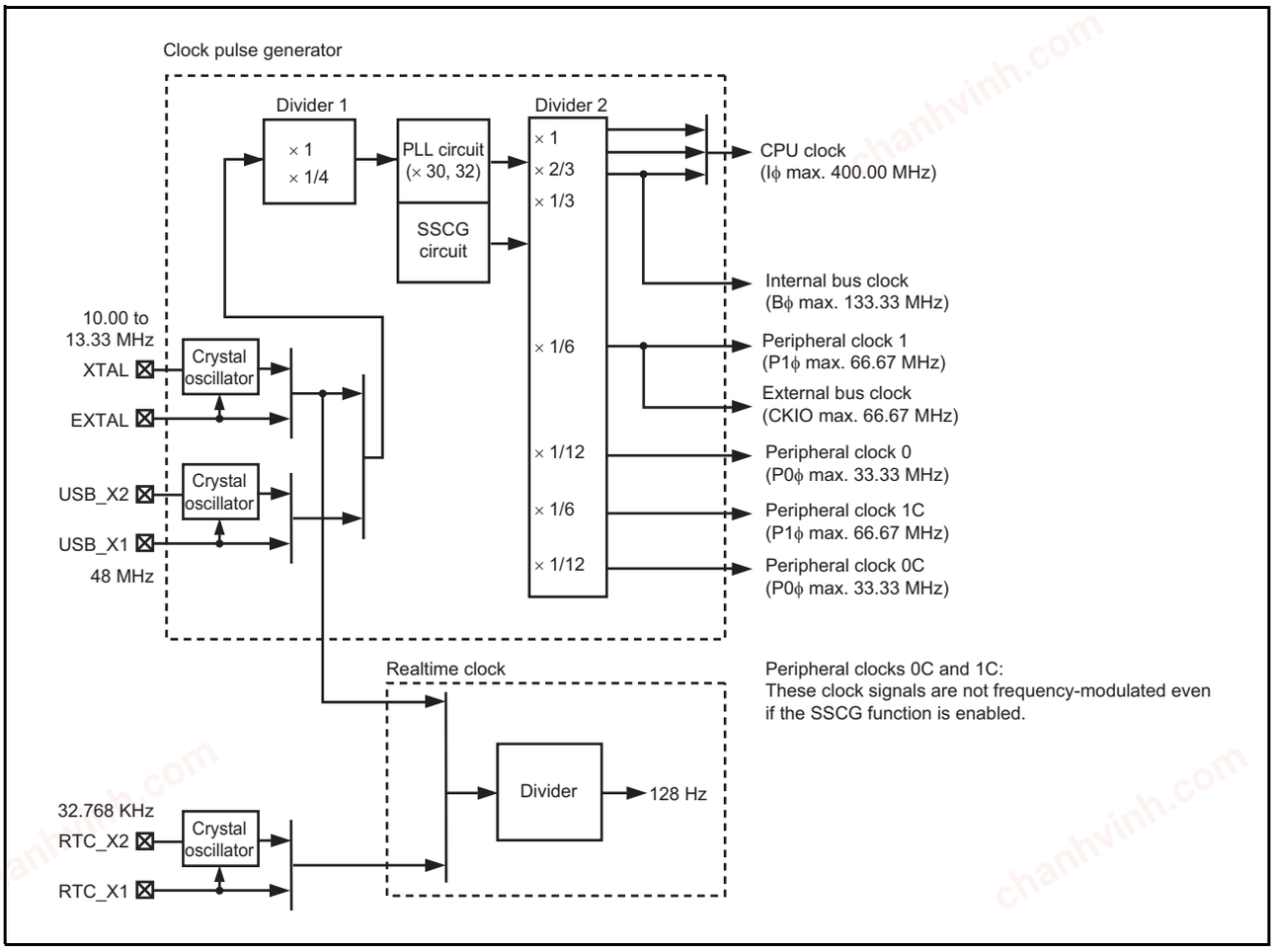


Figure 6.5 Clock Signals for the System and Realtime Clock

chanhvinh.com

6.10.2 Audio and USB Clock Signals

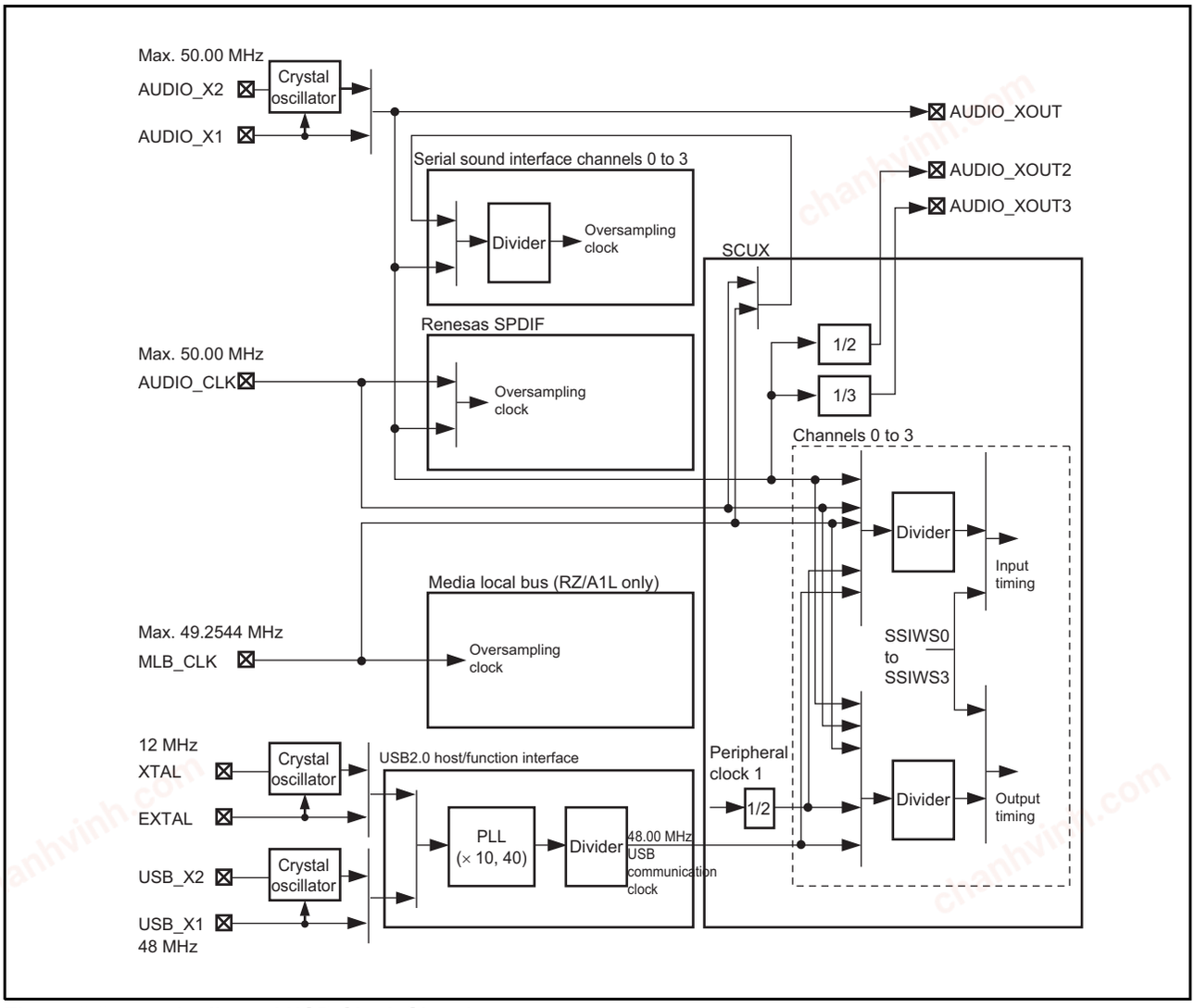


Figure 6.6 Audio and USB Clock Signals

6.10.3 Video Image Clock Signals

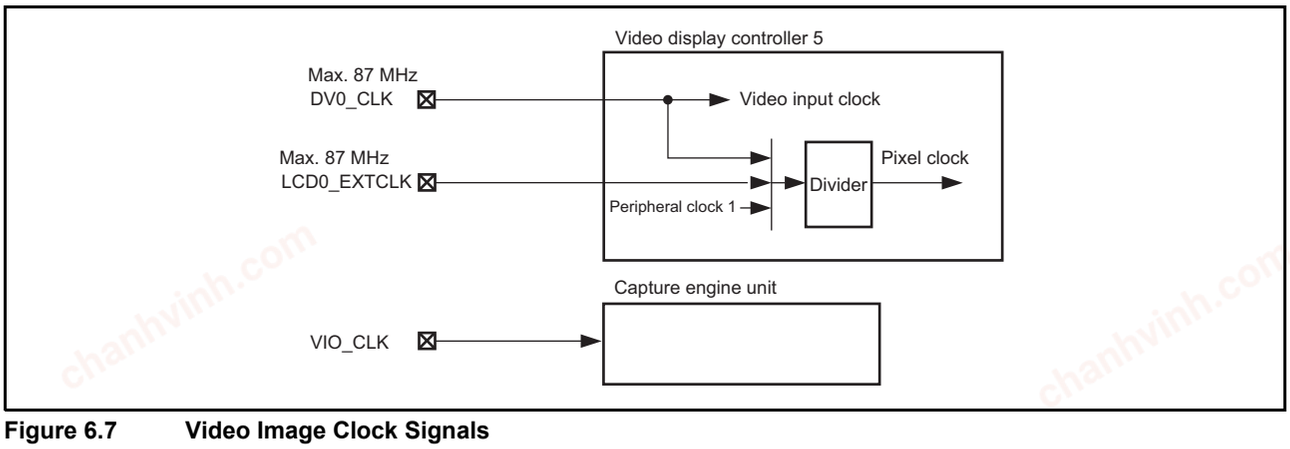


Figure 6.7 Video Image Clock Signals

chanhvinh.com

6.10.4 Other Clock Signals

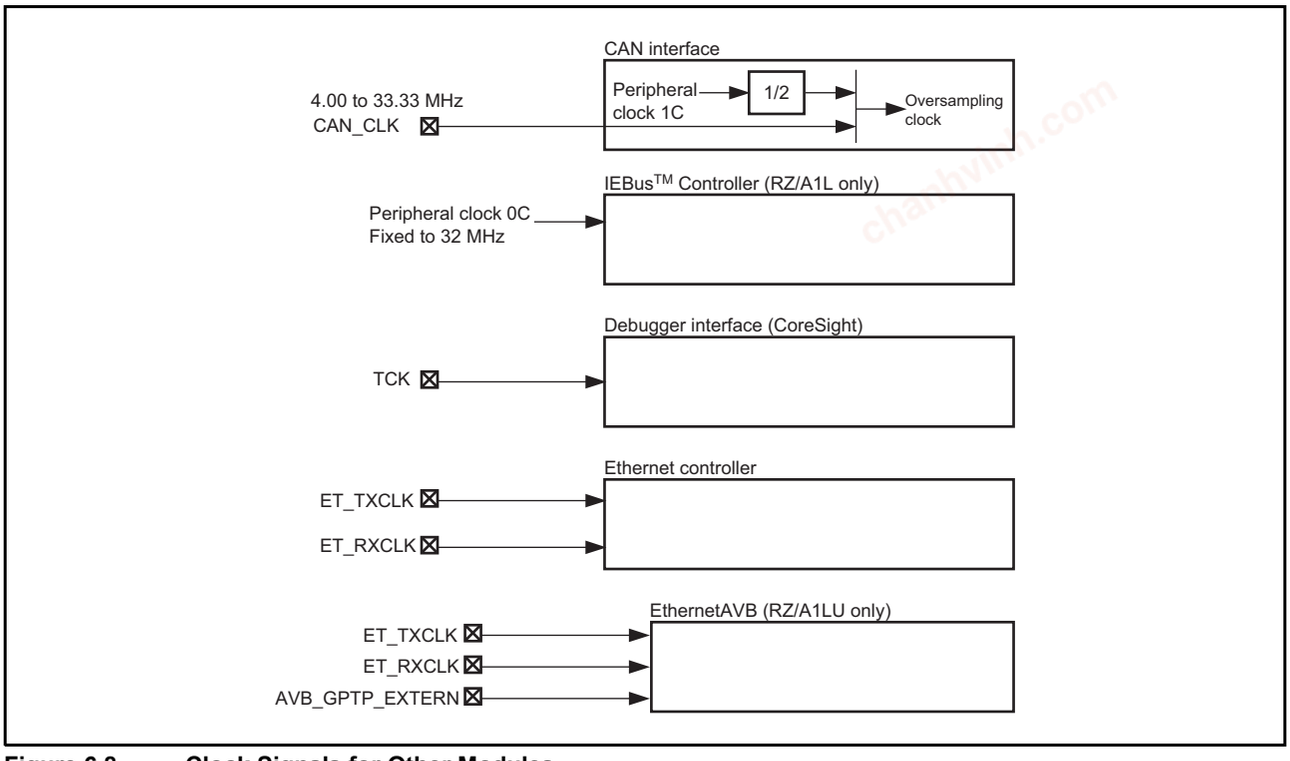


Figure 6.8 Clock Signals for Other Modules

chanhvinh.com

6.10.5 Internal Clock Signals (1)

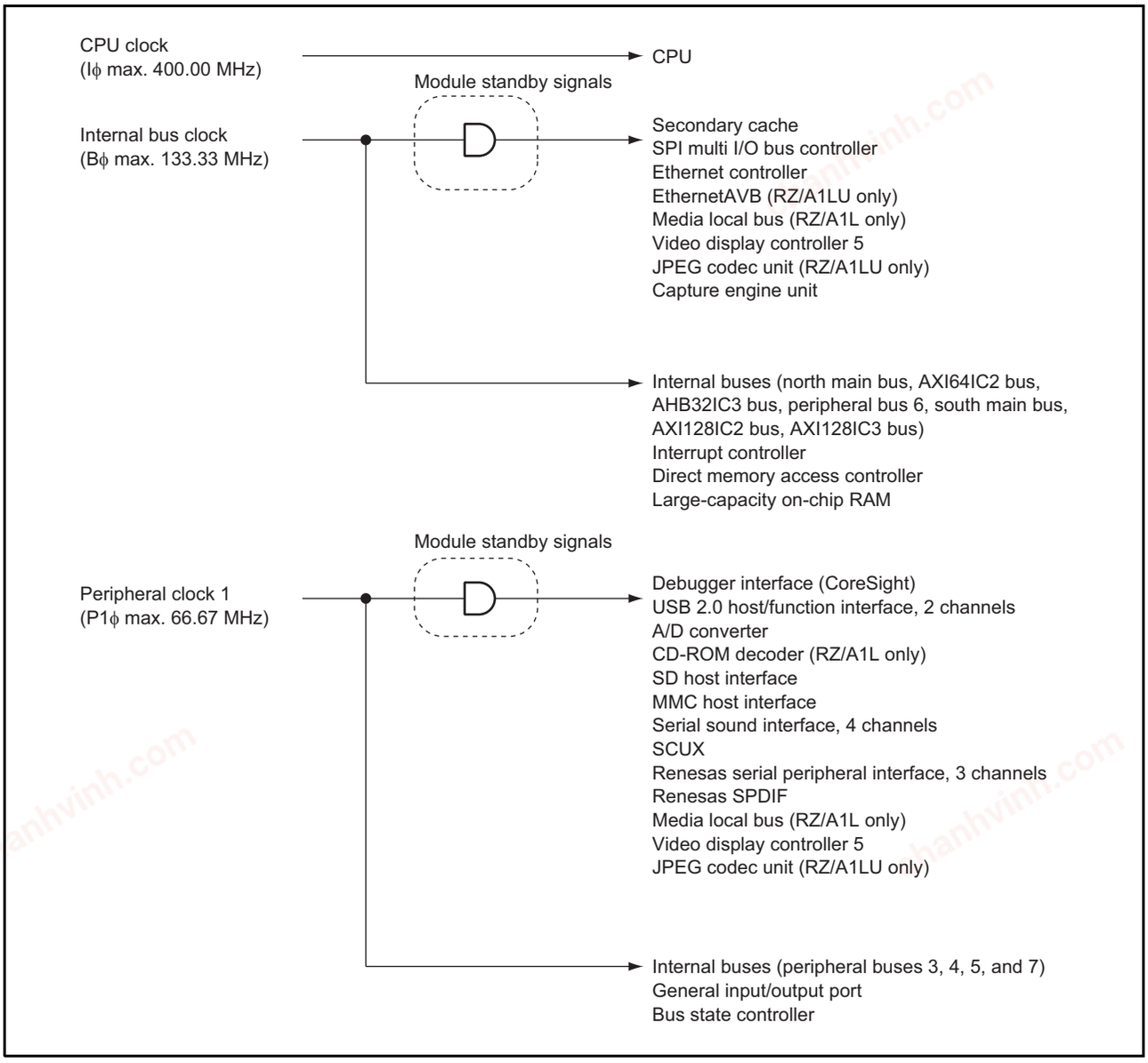


Figure 6.9 Distribution of Internal Clock Signals (1)

chanhvinh.com

6.10.6 Internal Clock Signals (2)

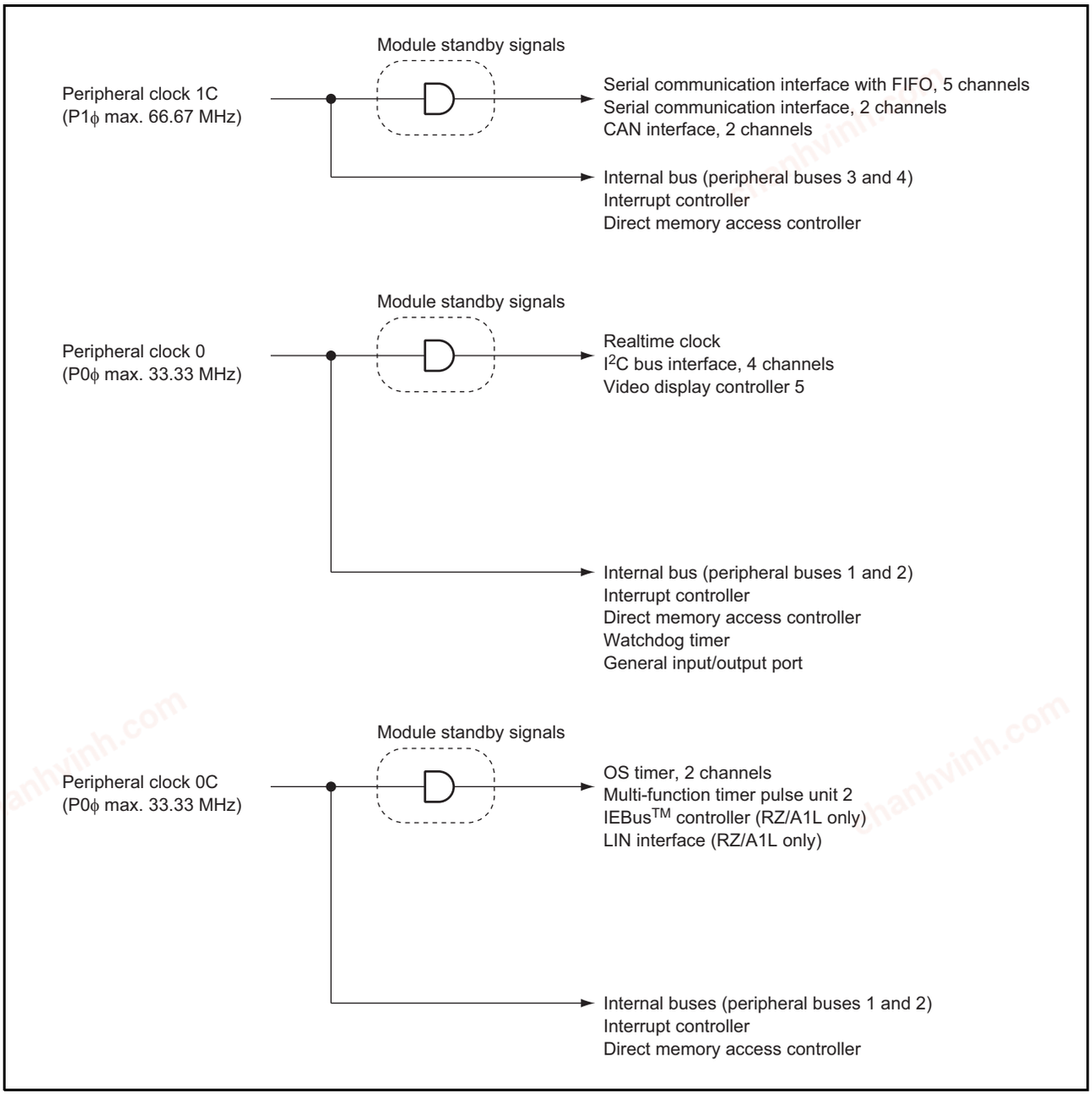


Figure 6.10 Distribution of Internal Clock Signals (2)

chanhvinh.com

6.11 Usage Note

6.11.1 Notes on the SSCG

When the SSCG is to be used, secure the SSCG stabilizing time (t_{SSCG}) shown in Table 47.5, Clock Timing in section 47, Electrical Characteristics. Furthermore, avoid deep standby while the SSCG is in use.

chanhvinh.com

7. Interrupt Controller

The interrupt controller ascertains the priority of interrupt sources and controls interrupt requests to the CPU. The interrupt controller registers set the order of priority of each interrupt, allowing the user to process interrupt requests according to the user-set priority.

7.1 Features

- 32 levels of interrupt priority can be set.
  By setting the interrupt priority registers, the priorities of IRQ interrupts, on-chip peripheral module interrupts, and pin interrupts can be selected from 32 levels for request sources.
- NMI noise canceler function
  An NMI input-level bit indicates the NMI pin state. By reading this bit in the interrupt exception service routine, the pin state can be checked, enabling it to be used as the noise canceler function.
- Arm PrimeCell® generic interrupt controller (PL390)*

Note: * The PL390 supports version 1 of the specification for the architecture of the Arm generic interrupt controller (GIC).

chanhvinh.com

Figure 7.1 shows a block diagram.

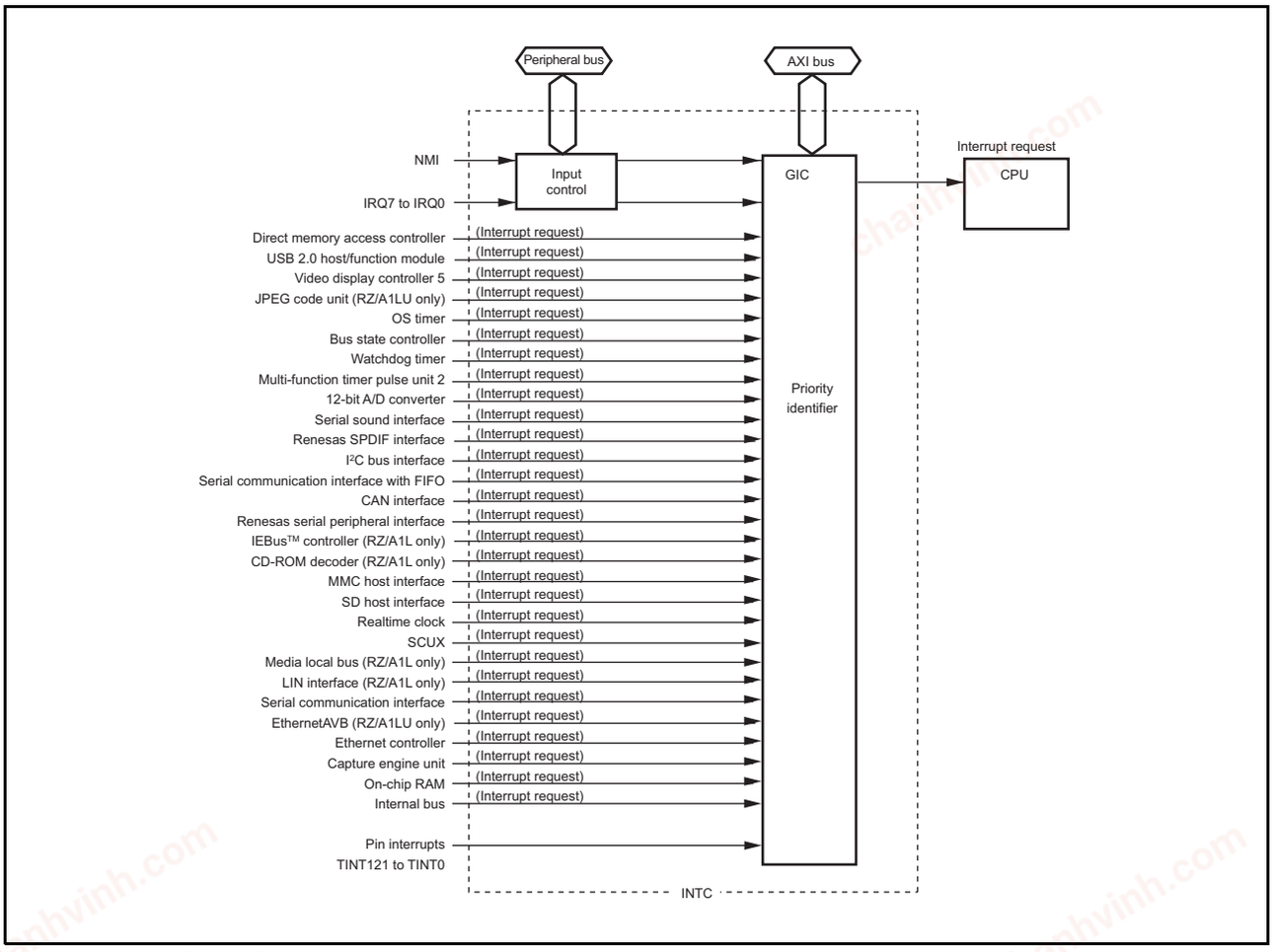


Figure 7.1 Block Diagram

7.2 Input/Output Pins

Table 7.1 shows the pin configuration.

Table 7.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Nonmaskable interrupt input pin	NMI	Input	Input of nonmaskable interrupt request signal
Interrupt request input pins	IRQ7 to IRQ0	Input	Input of maskable interrupt request signals
	TINT121 to TINT0	Input	Input of maskable interrupt request signals

chanhvinh.com

7.3 Register Descriptions

Table 7.2 shows the register configuration. These registers are used to set the interrupt priorities and control detection of the external interrupt input signal.

For a description of the registers other than interrupt control register 0, interrupt control register 1, and IRQ interrupt request register, see the Arm Generic Interrupt Controller Architecture Specification and the PrimeCell® Generic Interrupt Controller (PL390) Technical Reference Manual from Arm.

Table 7.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt control register 0	ICR0	R/W	*1	H'FCFEF800	16
Interrupt control register 1	ICR1	R/W	H'0000	H'FCFEF802	16
IRQ interrupt request register	IRQRR	R/(W)*2	H'0000	H'FCFEF804	16
Distributor control register	ICDDCR	RW	H'00000000	H'E8201000	32
Interrupt controller type register	ICDICTR	R	H'0000FC31*3	H'E8201004	32
Distributor implementer identification register	ICDIIDR	R	H'0000043B	H'E8201008	32
Interrupt security register 0	ICDISR0	R/W	H'00000000	H'E8201080	32
Interrupt security register 1	ICDISR1	R/W	H'00000000	H'E8201084	32
Interrupt security register 2	ICDISR2	R/W	H'00000000	H'E8201088	32
Interrupt security register 3	ICDISR3	R/W	H'00000000	H'E820108C	32
Interrupt security register 4	ICDISR4	R/W	H'00000000	H'E8201090	32
Interrupt security register 5	ICDISR5	R/W	H'00000000	H'E8201094	32
Interrupt security register 6	ICDISR6	R/W	H'00000000	H'E8201098	32
Interrupt security register 7	ICDISR7	R/W	H'00000000	H'E820109C	32
Interrupt security register 8	ICDISR8	R/W	H'00000000	H'E82010A0	32
Interrupt security register 9	ICDISR9	R/W	H'00000000	H'E82010A4	32
Interrupt security register 10	ICDISR10	R/W	H'00000000	H'E82010A8	32
Interrupt security register 11	ICDISR11	R/W	H'00000000	H'E82010AC	32
Interrupt security register 12	ICDISR12	R/W	H'00000000	H'E82010B0	32
Interrupt security register 13	ICDISR13	R/W	H'00000000	H'E82010B4	32
Interrupt security register 14	ICDISR14	R/W	H'00000000	H'E82010B8	32
Interrupt security register 15	ICDISR15	R/W	H'00000000	H'E82010BC	32
Interrupt security register 16	ICDISR16	R/W	H'00000000	H'E82010C0	32
Interrupt set-enable register 0	ICDISER0	R/W	H'00000000	H'E8201100	32
Interrupt set-enable register 1	ICDISER1	R/W	H'00000000	H'E8201104	32
Interrupt set-enable register 2	ICDISER2	R/W	H'00000000	H'E8201108	32
Interrupt set-enable register 3	ICDISER3	R/W	H'00000000	H'E820110C	32
Interrupt set-enable register 4	ICDISER4	R/W	H'00000000	H'E8201110	32
Interrupt set-enable register 5	ICDISER5	R/W	H'00000000	H'E8201114	32
Interrupt set-enable register 6	ICDISER6	R/W	H'00000000	H'E8201118	32
Interrupt set-enable register 7	ICDISER7	R/W	H'00000000	H'E820111C	32
Interrupt set-enable register 8	ICDISER8	R/W	H'00000000	H'E8201120	32
Interrupt set-enable register 9	ICDISER9	R/W	H'00000000	H'E8201124	32
Interrupt set-enable register 10	ICDISER10	R/W	H'00000000	H'E8201128	32
Interrupt set-enable register 11	ICDISER11	R/W	H'00000000	H'E820112C	32
Interrupt set-enable register 12	ICDISER12	R/W	H'00000000	H'E8201130	32
Interrupt set-enable register 13	ICDISER13	R/W	H'00000000	H'E8201134	32
Interrupt set-enable register 14	ICDISER14	R/W	H'00000000	H'E8201138	32

chanhvinh.com

Table 7.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt set-enable register 15	ICDISER15	R/W	H'00000000	H'E820113C	32
Interrupt set-enable register 16	ICDISER16	R/W	H'00000000	H'E8201140	32
Interrupt clear-enable register 0	ICDICER0	R/W	H'00000000	H'E8201180	32
Interrupt clear-enable register 1	ICDICER1	R/W	H'00000000	H'E8201184	32
Interrupt clear-enable register 2	ICDICER2	R/W	H'00000000	H'E8201188	32
Interrupt clear-enable register 3	ICDICER3	R/W	H'00000000	H'E820118C	32
Interrupt clear-enable register 4	ICDICER4	R/W	H'00000000	H'E8201190	32
Interrupt clear-enable register 5	ICDICER5	R/W	H'00000000	H'E8201194	32
Interrupt clear-enable register 6	ICDICER6	R/W	H'00000000	H'E8201198	32
Interrupt clear-enable register 7	ICDICER7	R/W	H'00000000	H'E820119C	32
Interrupt clear-enable register 8	ICDICER8	R/W	H'00000000	H'E82011A0	32
Interrupt clear-enable register 9	ICDICER9	R/W	H'00000000	H'E82011A4	32
Interrupt clear-enable register 10	ICDICER10	R/W	H'00000000	H'E82011A8	32
Interrupt clear-enable register 11	ICDICER11	R/W	H'00000000	H'E82011AC	32
Interrupt clear-enable register 12	ICDICER12	R/W	H'00000000	H'E82011B0	32
Interrupt clear-enable register 13	ICDICER13	R/W	H'00000000	H'E82011B4	32
Interrupt clear-enable register 14	ICDICER14	R/W	H'00000000	H'E82011B8	32
Interrupt clear-enable register 15	ICDICER15	R/W	H'00000000	H'E82011BC	32
Interrupt clear-enable register 16	ICDICER16	R/W	H'00000000	H'E82011C0	32
Interrupt set-pending register 0	ICDISPR0	R/W	H'00000000	H'E8201200	32
Interrupt set-pending register 1	ICDISPR1	R/W	H'00000000	H'E8201204	32
Interrupt set-pending register 2	ICDISPR2	R/W	H'00000000	H'E8201208	32
Interrupt set-pending register 3	ICDISPR3	R/W	H'00000000	H'E820120C	32
Interrupt set-pending register 4	ICDISPR4	R/W	H'00000000	H'E8201210	32
Interrupt set-pending register 5	ICDISPR5	R/W	H'00000000	H'E8201214	32
Interrupt set-pending register 6	ICDISPR6	R/W	H'00000000	H'E8201218	32
Interrupt set-pending register 7	ICDISPR7	R/W	H'00000000	H'E820121C	32
Interrupt set-pending register 8	ICDISPR8	R/W	H'00000000	H'E8201220	32
Interrupt set-pending register 9	ICDISPR9	R/W	H'00000000	H'E8201224	32
Interrupt set-pending register 10	ICDISPR10	R/W	H'00000000	H'E8201228	32
Interrupt set-pending register 11	ICDISPR11	R/W	H'00000000	H'E820122C	32
Interrupt set-pending register 12	ICDISPR12	R/W	H'00000000	H'E8201230	32
Interrupt set-pending register 13	ICDISPR13	R/W	H'00000000	H'E8201234	32
Interrupt set-pending register 14	ICDISPR14	R/W	H'00000000	H'E8201238	32
Interrupt set-pending register 15	ICDISPR15	R/W	H'00000000	H'E820123C	32
Interrupt set-pending register 16	ICDISPR16	R/W	H'00000000	H'E8201240	32
Interrupt clear-pending register 0	ICDICPR0	R/W	H'00000000	H'E8201280	32
Interrupt clear-pending register 1	ICDICPR1	R/W	H'00000000	H'E8201284	32
Interrupt clear-pending register 2	ICDICPR2	R/W	H'00000000	H'E8201288	32
Interrupt clear-pending register 3	ICDICPR3	R/W	H'00000000	H'E820128C	32
Interrupt clear-pending register 4	ICDICPR4	R/W	H'00000000	H'E8201290	32
Interrupt clear-pending register 5	ICDICPR5	R/W	H'00000000	H'E8201294	32
Interrupt clear-pending register 6	ICDICPR6	R/W	H'00000000	H'E8201298	32
Interrupt clear-pending register 7	ICDICPR7	R/W	H'00000000	H'E820129C	32
Interrupt clear-pending register 8	ICDICPR8	R/W	H'00000000	H'E82012A0	32

chanhvinh.com

Table 7.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt clear-pending register 9	ICDICPR9	R/W	H'00000000	H'E82012A4	32
Interrupt clear-pending register 10	ICDICPR10	R/W	H'00000000	H'E82012A8	32
Interrupt clear-pending register 11	ICDICPR11	R/W	H'00000000	H'E82012AC	32
Interrupt clear-pending register 12	ICDICPR12	R/W	H'00000000	H'E82012B0	32
Interrupt clear-pending register 13	ICDICPR13	R/W	H'00000000	H'E82012B4	32
Interrupt clear-pending register 14	ICDICPR14	R/W	H'00000000	H'E82012B8	32
Interrupt clear-pending register 15	ICDICPR15	R/W	H'00000000	H'E82012BC	32
Interrupt clear-pending register 16	ICDICPR16	R/W	H'00000000	H'E82012C0	32
Active bit register 0	ICDABR0	R/W	H'00000000	H'E8201300	32
Active bit register 1	ICDABR1	R/W	H'00000000	H'E8201304	32
Active bit register 2	ICDABR2	R/W	H'00000000	H'E8201308	32
Active bit register 3	ICDABR3	R/W	H'00000000	H'E820130C	32
Active bit register 4	ICDABR4	R/W	H'00000000	H'E8201310	32
Active bit register 5	ICDABR5	R/W	H'00000000	H'E8201314	32
Active bit register 6	ICDABR6	R/W	H'00000000	H'E8201318	32
Active bit register 7	ICDABR7	R/W	H'00000000	H'E820131C	32
Active bit register 8	ICDABR8	R/W	H'00000000	H'E8201320	32
Active bit register 9	ICDABR9	R/W	H'00000000	H'E8201324	32
Active bit register 10	ICDABR10	R/W	H'00000000	H'E8201328	32
Active bit register 11	ICDABR11	R/W	H'00000000	H'E820132C	32
Active bit register 12	ICDABR12	R/W	H'00000000	H'E8201330	32
Active bit register 13	ICDABR13	R/W	H'00000000	H'E8201334	32
Active bit register 14	ICDABR14	R/W	H'00000000	H'E8201338	32
Active bit register 15	ICDABR15	R/W	H'00000000	H'E820133C	32
Active bit register 16	ICDABR16	R/W	H'00000000	H'E8201340	32
Interrupt priority register 0	ICDIPR0	R/W	H'00000000	H'E8201400	32
Interrupt priority register 1	ICDIPR1	R/W	H'00000000	H'E8201404	32
Interrupt priority register 2	ICDIPR2	R/W	H'00000000	H'E8201408	32
Interrupt priority register 3	ICDIPR3	R/W	H'00000000	H'E820140C	32
Interrupt priority register 4	ICDIPR4	R/W	H'00000000	H'E8201410	32
Interrupt priority register 5	ICDIPR5	R/W	H'00000000	H'E8201414	32
Interrupt priority register 6	ICDIPR6	R/W	H'00000000	H'E8201418	32
Interrupt priority register 7	ICDIPR7	R/W	H'00000000	H'E820141C	32
Interrupt priority register 8	ICDIPR8	R/W	H'00000000	H'E8201420	32
Interrupt priority register 9	ICDIPR9	R/W	H'00000000	H'E8201424	32
Interrupt priority register 10	ICDIPR10	R/W	H'00000000	H'E8201428	32
Interrupt priority register 11	ICDIPR11	R/W	H'00000000	H'E820142C	32
Interrupt priority register 12	ICDIPR12	R/W	H'00000000	H'E8201430	32
Interrupt priority register 13	ICDIPR13	R/W	H'00000000	H'E8201434	32
Interrupt priority register 14	ICDIPR14	R/W	H'00000000	H'E8201438	32
Interrupt priority register 15	ICDIPR15	R/W	H'00000000	H'E820143C	32
Interrupt priority register 16	ICDIPR16	R/W	H'00000000	H'E8201440	32
Interrupt priority register 17	ICDIPR17	R/W	H'00000000	H'E8201444	32
Interrupt priority register 18	ICDIPR18	R/W	H'00000000	H'E8201448	32
Interrupt priority register 19	ICDIPR19	R/W	H'00000000	H'E820144C	32

chanhvinh.com

Table 7.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt priority register 20	ICDIPR20	R/W	H'00000000	H'E8201450	32
Interrupt priority register 21	ICDIPR21	R/W	H'00000000	H'E8201454	32
Interrupt priority register 22	ICDIPR22	R/W	H'00000000	H'E8201458	32
Interrupt priority register 23	ICDIPR23	R/W	H'00000000	H'E820145C	32
Interrupt priority register 24	ICDIPR24	R/W	H'00000000	H'E8201460	32
Interrupt priority register 25	ICDIPR25	R/W	H'00000000	H'E8201464	32
Interrupt priority register 26	ICDIPR26	R/W	H'00000000	H'E8201468	32
Interrupt priority register 27	ICDIPR27	R/W	H'00000000	H'E820146C	32
Interrupt priority register 28	ICDIPR28	R/W	H'00000000	H'E8201470	32
Interrupt priority register 29	ICDIPR29	R/W	H'00000000	H'E8201474	32
Interrupt priority register 30	ICDIPR30	R/W	H'00000000	H'E8201478	32
Interrupt priority register 31	ICDIPR31	R/W	H'00000000	H'E820147C	32
Interrupt priority register 32	ICDIPR32	R/W	H'00000000	H'E8201480	32
Interrupt priority register 33	ICDIPR33	R/W	H'00000000	H'E8201484	32
Interrupt priority register 34	ICDIPR34	R/W	H'00000000	H'E8201488	32
Interrupt priority register 35	ICDIPR35	R/W	H'00000000	H'E820148C	32
Interrupt priority register 36	ICDIPR36	R/W	H'00000000	H'E8201490	32
Interrupt priority register 37	ICDIPR37	R/W	H'00000000	H'E8201494	32
Interrupt priority register 38	ICDIPR38	R/W	H'00000000	H'E8201498	32
Interrupt priority register 39	ICDIPR39	R/W	H'00000000	H'E820149C	32
Interrupt priority register 40	ICDIPR40	R/W	H'00000000	H'E82014A0	32
Interrupt priority register 41	ICDIPR41	R/W	H'00000000	H'E82014A4	32
Interrupt priority register 42	ICDIPR42	R/W	H'00000000	H'E82014A8	32
Interrupt priority register 43	ICDIPR43	R/W	H'00000000	H'E82014AC	32
Interrupt priority register 44	ICDIPR44	R/W	H'00000000	H'E82014B0	32
Interrupt priority register 45	ICDIPR45	R/W	H'00000000	H'E82014B4	32
Interrupt priority register 46	ICDIPR46	R/W	H'00000000	H'E82014B8	32
Interrupt priority register 47	ICDIPR47	R/W	H'00000000	H'E82014BC	32
Interrupt priority register 48	ICDIPR48	R/W	H'00000000	H'E82014C0	32
Interrupt priority register 49	ICDIPR49	R/W	H'00000000	H'E82014C4	32
Interrupt priority register 50	ICDIPR50	R/W	H'00000000	H'E82014C8	32
Interrupt priority register 51	ICDIPR51	R/W	H'00000000	H'E82014CC	32
Interrupt priority register 52	ICDIPR52	R/W	H'00000000	H'E82014D0	32
Interrupt priority register 53	ICDIPR53	R/W	H'00000000	H'E82014D4	32
Interrupt priority register 54	ICDIPR54	R/W	H'00000000	H'E82014D8	32
Interrupt priority register 55	ICDIPR55	R/W	H'00000000	H'E82014DC	32
Interrupt priority register 56	ICDIPR56	R/W	H'00000000	H'E82014E0	32
Interrupt priority register 57	ICDIPR57	R/W	H'00000000	H'E82014E4	32
Interrupt priority register 58	ICDIPR58	R/W	H'00000000	H'E82014E8	32
Interrupt priority register 59	ICDIPR59	R/W	H'00000000	H'E82014EC	32
Interrupt priority register 60	ICDIPR60	R/W	H'00000000	H'E82014F0	32
Interrupt priority register 61	ICDIPR61	R/W	H'00000000	H'E82014F4	32
Interrupt priority register 62	ICDIPR62	R/W	H'00000000	H'E82014F8	32
Interrupt priority register 63	ICDIPR63	R/W	H'00000000	H'E82014FC	32
Interrupt priority register 64	ICDIPR64	R/W	H'00000000	H'E8201500	32

chanhvinh.com

Table 7.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt priority register 65	ICDIPR65	R/W	H'00000000	H'E8201504	32
Interrupt priority register 66	ICDIPR66	R/W	H'00000000	H'E8201508	32
Interrupt priority register 67	ICDIPR67	R/W	H'00000000	H'E820150C	32
Interrupt priority register 68	ICDIPR68	R/W	H'00000000	H'E8201510	32
Interrupt priority register 69	ICDIPR69	R/W	H'00000000	H'E8201514	32
Interrupt priority register 70	ICDIPR70	R/W	H'00000000	H'E8201518	32
Interrupt priority register 71	ICDIPR71	R/W	H'00000000	H'E820151C	32
Interrupt priority register 72	ICDIPR72	R/W	H'00000000	H'E8201520	32
Interrupt priority register 73	ICDIPR73	R/W	H'00000000	H'E8201524	32
Interrupt priority register 74	ICDIPR74	R/W	H'00000000	H'E8201528	32
Interrupt priority register 75	ICDIPR75	R/W	H'00000000	H'E820152C	32
Interrupt priority register 76	ICDIPR76	R/W	H'00000000	H'E8201530	32
Interrupt priority register 77	ICDIPR77	R/W	H'00000000	H'E8201534	32
Interrupt priority register 78	ICDIPR78	R/W	H'00000000	H'E8201538	32
Interrupt priority register 79	ICDIPR79	R/W	H'00000000	H'E820153C	32
Interrupt priority register 80	ICDIPR80	R/W	H'00000000	H'E8201540	32
Interrupt priority register 81	ICDIPR81	R/W	H'00000000	H'E8201544	32
Interrupt priority register 82	ICDIPR82	R/W	H'00000000	H'E8201548	32
Interrupt priority register 83	ICDIPR83	R/W	H'00000000	H'E820154C	32
Interrupt priority register 84	ICDIPR84	R/W	H'00000000	H'E8201550	32
Interrupt priority register 85	ICDIPR85	R/W	H'00000000	H'E8201554	32
Interrupt priority register 86	ICDIPR86	R/W	H'00000000	H'E8201558	32
Interrupt priority register 87	ICDIPR87	R/W	H'00000000	H'E820155C	32
Interrupt priority register 88	ICDIPR88	R/W	H'00000000	H'E8201560	32
Interrupt priority register 89	ICDIPR89	R/W	H'00000000	H'E8201564	32
Interrupt priority register 90	ICDIPR90	R/W	H'00000000	H'E8201568	32
Interrupt priority register 91	ICDIPR91	R/W	H'00000000	H'E820156C	32
Interrupt priority register 92	ICDIPR92	R/W	H'00000000	H'E8201570	32
Interrupt priority register 93	ICDIPR93	R/W	H'00000000	H'E8201574	32
Interrupt priority register 94	ICDIPR94	R/W	H'00000000	H'E8201578	32
Interrupt priority register 95	ICDIPR95	R/W	H'00000000	H'E820157C	32
Interrupt priority register 96	ICDIPR96	R/W	H'00000000	H'E8201580	32
Interrupt priority register 97	ICDIPR97	R/W	H'00000000	H'E8201584	32
Interrupt priority register 98	ICDIPR98	R/W	H'00000000	H'E8201588	32
Interrupt priority register 99	ICDIPR99	R/W	H'00000000	H'E820158C	32
Interrupt priority register 100	ICDIPR100	R/W	H'00000000	H'E8201590	32
Interrupt priority register 101	ICDIPR101	R/W	H'00000000	H'E8201594	32
Interrupt priority register 102	ICDIPR102	R/W	H'00000000	H'E8201598	32
Interrupt priority register 103	ICDIPR103	R/W	H'00000000	H'E820159C	32
Interrupt priority register 104	ICDIPR104	R/W	H'00000000	H'E82015A0	32
Interrupt priority register 105	ICDIPR105	R/W	H'00000000	H'E82015A4	32
Interrupt priority register 106	ICDIPR106	R/W	H'00000000	H'E82015A8	32
Interrupt priority register 107	ICDIPR107	R/W	H'00000000	H'E82015AC	32
Interrupt priority register 108	ICDIPR108	R/W	H'00000000	H'E82015B0	32
Interrupt priority register 109	ICDIPR109	R/W	H'00000000	H'E82015B4	32

chanhvinh.com

Table 7.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt priority register 110	ICDIPR110	R/W	H'00000000	H'E82015B8	32
Interrupt priority register 111	ICDIPR111	R/W	H'00000000	H'E82015BC	32
Interrupt priority register 112	ICDIPR112	R/W	H'00000000	H'E82015C0	32
Interrupt priority register 113	ICDIPR113	R/W	H'00000000	H'E82015C4	32
Interrupt priority register 114	ICDIPR114	R/W	H'00000000	H'E82015C8	32
Interrupt priority register 115	ICDIPR115	R/W	H'00000000	H'E82015CC	32
Interrupt priority register 116	ICDIPR116	R/W	H'00000000	H'E82015D0	32
Interrupt priority register 117	ICDIPR117	R/W	H'00000000	H'E82015D4	32
Interrupt priority register 118	ICDIPR118	R/W	H'00000000	H'E82015D8	32
Interrupt priority register 119	ICDIPR119	R/W	H'00000000	H'E82015DC	32
Interrupt priority register 120	ICDIPR120	R/W	H'00000000	H'E82015E0	32
Interrupt priority register 121	ICDIPR121	R/W	H'00000000	H'E82015E4	32
Interrupt priority register 122	ICDIPR122	R/W	H'00000000	H'E82015E8	32
Interrupt priority register 123	ICDIPR123	R/W	H'00000000	H'E82015EC	32
Interrupt priority register 124	ICDIPR124	R/W	H'00000000	H'E82015F0	32
Interrupt priority register 125	ICDIPR125	R/W	H'00000000	H'E82015F4	32
Interrupt priority register 126	ICDIPR126	R/W	H'00000000	H'E82015F8	32
Interrupt priority register 127	ICDIPR127	R/W	H'00000000	H'E82015FC	32
Interrupt priority register 128	ICDIPR128	R/W	H'00000000	H'E8201600	32
Interrupt priority register 129	ICDIPR129	R/W	H'00000000	H'E8201604	32
Interrupt priority register 130	ICDIPR130	R/W	H'00000000	H'E8201608	32
Interrupt priority register 131	ICDIPR131	R/W	H'00000000	H'E820160C	32
Interrupt priority register 132	ICDIPR132	R/W	H'00000000	H'E8201610	32
Interrupt priority register 133	ICDIPR133	R/W	H'00000000	H'E8201614	32
Interrupt priority register 134	ICDIPR134	R/W	H'00000000	H'E8201618	32
Interrupt processor target register 0	ICDIPTR0	R	H'00000000	H'E8201800	32
Interrupt processor target register 1	ICDIPTR1	R	H'00000000	H'E8201804	32
Interrupt processor target register 2	ICDIPTR2	R	H'00000000	H'E8201808	32
Interrupt processor target register 3	ICDIPTR3	R	H'00000000	H'E820180C	32
Interrupt processor target register 4	ICDIPTR4	R	H'00000000	H'E8201810	32
Interrupt processor target register 5	ICDIPTR5	R	H'00000000	H'E8201814	32
Interrupt processor target register 6	ICDIPTR6	R	H'00000000	H'E8201818	32
Interrupt processor target register 7	ICDIPTR7	R	H'00000000	H'E820181C	32
Interrupt processor target register 8	ICDIPTR8	R/W	H'00000000	H'E8201820	32
Interrupt processor target register 9	ICDIPTR9	R/W	H'00000000	H'E8201824	32
Interrupt processor target register 10	ICDIPTR10	R/W	H'00000000	H'E8201828	32
Interrupt processor target register 11	ICDIPTR11	R/W	H'00000000	H'E820182C	32
Interrupt processor target register 12	ICDIPTR12	R/W	H'00000000	H'E8201830	32
Interrupt processor target register 13	ICDIPTR13	R/W	H'00000000	H'E8201834	32
Interrupt processor target register 14	ICDIPTR14	R/W	H'00000000	H'E8201838	32
Interrupt processor target register 15	ICDIPTR15	R/W	H'00000000	H'E820183C	32
Interrupt processor target register 16	ICDIPTR16	R/W	H'00000000	H'E8201840	32
Interrupt processor target register 17	ICDIPTR17	R/W	H'00000000	H'E8201844	32
Interrupt processor target register 18	ICDIPTR18	R/W	H'00000000	H'E8201848	32
Interrupt processor target register 19	ICDIPTR19	R/W	H'00000000	H'E820184C	32

chanhvinh.com

Table 7.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt processor target register 20	ICDIPTR20	R/W	H'00000000	H'E8201850	32
Interrupt processor target register 21	ICDIPTR21	R/W	H'00000000	H'E8201854	32
Interrupt processor target register 22	ICDIPTR22	R/W	H'00000000	H'E8201858	32
Interrupt processor target register 23	ICDIPTR23	R/W	H'00000000	H'E820185C	32
Interrupt processor target register 24	ICDIPTR24	R/W	H'00000000	H'E8201860	32
Interrupt processor target register 25	ICDIPTR25	R/W	H'00000000	H'E8201864	32
Interrupt processor target register 26	ICDIPTR26	R/W	H'00000000	H'E8201868	32
Interrupt processor target register 27	ICDIPTR27	R/W	H'00000000	H'E820186C	32
Interrupt processor target register 28	ICDIPTR28	R/W	H'00000000	H'E8201870	32
Interrupt processor target register 29	ICDIPTR29	R/W	H'00000000	H'E8201874	32
Interrupt processor target register 30	ICDIPTR30	R/W	H'00000000	H'E8201878	32
Interrupt processor target register 31	ICDIPTR31	R/W	H'00000000	H'E820187C	32
Interrupt processor target register 32	ICDIPTR32	R/W	H'00000000	H'E8201880	32
Interrupt processor target register 33	ICDIPTR33	R/W	H'00000000	H'E8201884	32
Interrupt processor target register 34	ICDIPTR34	R/W	H'00000000	H'E8201888	32
Interrupt processor target register 35	ICDIPTR35	R/W	H'00000000	H'E820188C	32
Interrupt processor target register 36	ICDIPTR36	R/W	H'00000000	H'E8201890	32
Interrupt processor target register 37	ICDIPTR37	R/W	H'00000000	H'E8201894	32
Interrupt processor target register 38	ICDIPTR38	R/W	H'00000000	H'E8201898	32
Interrupt processor target register 39	ICDIPTR39	R/W	H'00000000	H'E820189C	32
Interrupt processor target register 40	ICDIPTR40	R/W	H'00000000	H'E82018A0	32
Interrupt processor target register 41	ICDIPTR41	R/W	H'00000000	H'E82018A4	32
Interrupt processor target register 42	ICDIPTR42	R/W	H'00000000	H'E82018A8	32
Interrupt processor target register 43	ICDIPTR43	R/W	H'00000000	H'E82018AC	32
Interrupt processor target register 44	ICDIPTR44	R/W	H'00000000	H'E82018B0	32
Interrupt processor target register 45	ICDIPTR45	R/W	H'00000000	H'E82018B4	32
Interrupt processor target register 46	ICDIPTR46	R/W	H'00000000	H'E82018B8	32
Interrupt processor target register 47	ICDIPTR47	R/W	H'00000000	H'E82018BC	32
Interrupt processor target register 48	ICDIPTR48	R/W	H'00000000	H'E82018C0	32
Interrupt processor target register 49	ICDIPTR49	R/W	H'00000000	H'E82018C4	32
Interrupt processor target register 50	ICDIPTR50	R/W	H'00000000	H'E82018C8	32
Interrupt processor target register 51	ICDIPTR51	R/W	H'00000000	H'E82018CC	32
Interrupt processor target register 52	ICDIPTR52	R/W	H'00000000	H'E82018D0	32
Interrupt processor target register 53	ICDIPTR53	R/W	H'00000000	H'E82018D4	32
Interrupt processor target register 54	ICDIPTR54	R/W	H'00000000	H'E82018D8	32
Interrupt processor target register 55	ICDIPTR55	R/W	H'00000000	H'E82018DC	32
Interrupt processor target register 56	ICDIPTR56	R/W	H'00000000	H'E82018E0	32
Interrupt processor target register 57	ICDIPTR57	R/W	H'00000000	H'E82018E4	32
Interrupt processor target register 58	ICDIPTR58	R/W	H'00000000	H'E82018E8	32
Interrupt processor target register 59	ICDIPTR59	R/W	H'00000000	H'E82018EC	32
Interrupt processor target register 60	ICDIPTR60	R/W	H'00000000	H'E82018F0	32
Interrupt processor target register 61	ICDIPTR61	R/W	H'00000000	H'E82018F4	32
Interrupt processor target register 62	ICDIPTR62	R/W	H'00000000	H'E82018F8	32
Interrupt processor target register 63	ICDIPTR63	R/W	H'00000000	H'E82018FC	32
Interrupt processor target register 64	ICDIPTR64	R/W	H'00000000	H'E8201900	32

chanhvinh.com

Table 7.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt processor target register 65	ICDIPTR65	R/W	H'00000000	H'E8201904	32
Interrupt processor target register 66	ICDIPTR66	R/W	H'00000000	H'E8201908	32
Interrupt processor target register 67	ICDIPTR67	R/W	H'00000000	H'E820190C	32
Interrupt processor target register 68	ICDIPTR68	R/W	H'00000000	H'E8201910	32
Interrupt processor target register 69	ICDIPTR69	R/W	H'00000000	H'E8201914	32
Interrupt processor target register 70	ICDIPTR70	R/W	H'00000000	H'E8201918	32
Interrupt processor target register 71	ICDIPTR71	R/W	H'00000000	H'E820191C	32
Interrupt processor target register 72	ICDIPTR72	R/W	H'00000000	H'E8201920	32
Interrupt processor target register 73	ICDIPTR73	R/W	H'00000000	H'E8201924	32
Interrupt processor target register 74	ICDIPTR74	R/W	H'00000000	H'E8201928	32
Interrupt processor target register 75	ICDIPTR75	R/W	H'00000000	H'E820192C	32
Interrupt processor target register 76	ICDIPTR76	R/W	H'00000000	H'E8201930	32
Interrupt processor target register 77	ICDIPTR77	R/W	H'00000000	H'E8201934	32
Interrupt processor target register 78	ICDIPTR78	R/W	H'00000000	H'E8201938	32
Interrupt processor target register 79	ICDIPTR79	R/W	H'00000000	H'E820193C	32
Interrupt processor target register 80	ICDIPTR80	R/W	H'00000000	H'E8201940	32
Interrupt processor target register 81	ICDIPTR81	R/W	H'00000000	H'E8201944	32
Interrupt processor target register 82	ICDIPTR82	R/W	H'00000000	H'E8201948	32
Interrupt processor target register 83	ICDIPTR83	R/W	H'00000000	H'E820194C	32
Interrupt processor target register 84	ICDIPTR84	R/W	H'00000000	H'E8201950	32
Interrupt processor target register 85	ICDIPTR85	R/W	H'00000000	H'E8201954	32
Interrupt processor target register 86	ICDIPTR86	R/W	H'00000000	H'E8201958	32
Interrupt processor target register 87	ICDIPTR87	R/W	H'00000000	H'E820195C	32
Interrupt processor target register 88	ICDIPTR88	R/W	H'00000000	H'E8201960	32
Interrupt processor target register 89	ICDIPTR89	R/W	H'00000000	H'E8201964	32
Interrupt processor target register 90	ICDIPTR90	R/W	H'00000000	H'E8201968	32
Interrupt processor target register 91	ICDIPTR91	R/W	H'00000000	H'E820196C	32
Interrupt processor target register 92	ICDIPTR92	R/W	H'00000000	H'E8201970	32
Interrupt processor target register 93	ICDIPTR93	R/W	H'00000000	H'E8201974	32
Interrupt processor target register 94	ICDIPTR94	R/W	H'00000000	H'E8201978	32
Interrupt processor target register 95	ICDIPTR95	R/W	H'00000000	H'E820197C	32
Interrupt processor target register 96	ICDIPTR96	R/W	H'00000000	H'E8201980	32
Interrupt processor target register 97	ICDIPTR97	R/W	H'00000000	H'E8201984	32
Interrupt processor target register 98	ICDIPTR98	R/W	H'00000000	H'E8201988	32
Interrupt processor target register 99	ICDIPTR99	R/W	H'00000000	H'E820198C	32
Interrupt processor target register 100	ICDIPTR100	R/W	H'00000000	H'E8201990	32
Interrupt processor target register 101	ICDIPTR101	R/W	H'00000000	H'E8201994	32
Interrupt processor target register 102	ICDIPTR102	R/W	H'00000000	H'E8201998	32
Interrupt processor target register 103	ICDIPTR103	R/W	H'00000000	H'E820199C	32
Interrupt processor target register 104	ICDIPTR104	R/W	H'00000000	H'E82019A0	32
Interrupt processor target register 105	ICDIPTR105	R/W	H'00000000	H'E82019A4	32
Interrupt processor target register 106	ICDIPTR106	R/W	H'00000000	H'E82019A8	32
Interrupt processor target register 107	ICDIPTR107	R/W	H'00000000	H'E82019AC	32
Interrupt processor target register 108	ICDIPTR108	R/W	H'00000000	H'E82019B0	32
Interrupt processor target register 109	ICDIPTR109	R/W	H'00000000	H'E82019B4	32

chanhvinh.com

Table 7.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt processor target register 110	ICDIPTR110	R/W	H'00000000	H'E82019B8	32
Interrupt processor target register 111	ICDIPTR111	R/W	H'00000000	H'E82019BC	32
Interrupt processor target register 112	ICDIPTR112	R/W	H'00000000	H'E82019C0	32
Interrupt processor target register 113	ICDIPTR113	R/W	H'00000000	H'E82019C4	32
Interrupt processor target register 114	ICDIPTR114	R/W	H'00000000	H'E82019C8	32
Interrupt processor target register 115	ICDIPTR115	R/W	H'00000000	H'E82019CC	32
Interrupt processor target register 116	ICDIPTR116	R/W	H'00000000	H'E82019D0	32
Interrupt processor target register 117	ICDIPTR117	R/W	H'00000000	H'E82019D4	32
Interrupt processor target register 118	ICDIPTR118	R/W	H'00000000	H'E82019D8	32
Interrupt processor target register 119	ICDIPTR119	R/W	H'00000000	H'E82019DC	32
Interrupt processor target register 120	ICDIPTR120	R/W	H'00000000	H'E82019E0	32
Interrupt processor target register 121	ICDIPTR121	R/W	H'00000000	H'E82019E4	32
Interrupt processor target register 122	ICDIPTR122	R/W	H'00000000	H'E82019E8	32
Interrupt processor target register 123	ICDIPTR123	R/W	H'00000000	H'E82019EC	32
Interrupt processor target register 124	ICDIPTR124	R/W	H'00000000	H'E82019F0	32
Interrupt processor target register 125	ICDIPTR125	R/W	H'00000000	H'E82019F4	32
Interrupt processor target register 126	ICDIPTR126	R/W	H'00000000	H'E82019F8	32
Interrupt processor target register 127	ICDIPTR127	R/W	H'00000000	H'E82019FC	32
Interrupt processor target register 128	ICDIPTR128	R/W	H'00000000	H'E8201A00	32
Interrupt processor target register 129	ICDIPTR129	R/W	H'00000000	H'E8201A04	32
Interrupt processor target register 130	ICDIPTR130	R/W	H'00000000	H'E8201A08	32
Interrupt processor target register 131	ICDIPTR131	R/W	H'00000000	H'E8201A0C	32
Interrupt processor target register 132	ICDIPTR132	R/W	H'00000000	H'E8201A10	32
Interrupt processor target register 133	ICDIPTR133	R/W	H'00000000	H'E8201A14	32
Interrupt processor target register 134	ICDIPTR134	R/W	H'00000000	H'E8201A18	32
Interrupt configuration register 0	ICDICFR0	R	H'AAAAAAAA	H'E8201C00	32
Interrupt configuration register 1	ICDICFR1	R/W	H'55555555	H'E8201C04	32
Interrupt configuration register 2	ICDICFR2	R/W	H'55555555	H'E8201C08	32
Interrupt configuration register 3	ICDICFR3	R/W	H'55555555	H'E8201C0C	32
Interrupt configuration register 4	ICDICFR4	R/W	H'55555555	H'E8201C10	32
Interrupt configuration register 5	ICDICFR5	R/W	H'55555555	H'E8201C14	32
Interrupt configuration register 6	ICDICFR6	R/W	H'55555555	H'E8201C18	32
Interrupt configuration register 7	ICDICFR7	R/W	H'55555555	H'E8201C1C	32
Interrupt configuration register 8	ICDICFR8	R/W	H'55555555	H'E8201C20	32
Interrupt configuration register 9	ICDICFR9	R/W	H'55555555	H'E8201C24	32
Interrupt configuration register 10	ICDICFR10	R/W	H'55555555	H'E8201C28	32
Interrupt configuration register 11	ICDICFR11	R/W	H'55555555	H'E8201C2C	32
Interrupt configuration register 12	ICDICFR12	R/W	H'55555555	H'E8201C30	32
Interrupt configuration register 13	ICDICFR13	R/W	H'55555555	H'E8201C34	32
Interrupt configuration register 14	ICDICFR14	R/W	H'55555555	H'E8201C38	32
Interrupt configuration register 15	ICDICFR15	R/W	H'55555555	H'E8201C3C	32
Interrupt configuration register 16	ICDICFR16	R/W	H'55555555	H'E8201C40	32
Interrupt configuration register 17	ICDICFR17	R/W	H'55555555	H'E8201C44	32
Interrupt configuration register 18	ICDICFR18	R/W	H'55555555	H'E8201C48	32
Interrupt configuration register 19	ICDICFR19	R/W	H'55555555	H'E8201C4C	32

chanhvinh.com

Table 7.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt configuration register 20	ICDICFR20	R/W	H'55555555	H'E8201C50	32
Interrupt configuration register 21	ICDICFR21	R/W	H'55555555	H'E8201C54	32
Interrupt configuration register 22	ICDICFR22	R/W	H'55555555	H'E8201C58	32
Interrupt configuration register 23	ICDICFR23	R/W	H'55555555	H'E8201C5C	32
Interrupt configuration register 24	ICDICFR24	R/W	H'55555555	H'E8201C60	32
Interrupt configuration register 25	ICDICFR25	R/W	H'55555555	H'E8201C64	32
Interrupt configuration register 26	ICDICFR26	R/W	H'55555555	H'E8201C68	32
Interrupt configuration register 27	ICDICFR27	R/W	H'55555555	H'E8201C6C	32
Interrupt configuration register 28	ICDICFR28	R/W	H'55555555	H'E8201C70	32
Interrupt configuration register 29	ICDICFR29	R/W	H'55555555	H'E8201C74	32
Interrupt configuration register 30	ICDICFR30	R/W	H'55555555	H'E8201C78	32
Interrupt configuration register 31	ICDICFR31	R/W	H'55555555	H'E8201C7C	32
Interrupt configuration register 32	ICDICFR32	R/W	H'55555555	H'E8201C80	32
Interrupt configuration register 33	ICDICFR33	R/W	H'55555555	H'E8201C84	32
PPI status register	ppi_status	R	H'00000000	H'E8201D00	32
SPI status register 0	spi_status0	R	H'00000000	H'E8201D04	32
SPI status register 1	spi_status1	R	H'00000000	H'E8201D08	32
SPI status register 2	spi_status2	R	H'00000000	H'E8201D0C	32
SPI status register 3	spi_status3	R	H'00000000	H'E8201D10	32
SPI status register 4	spi_status4	R	H'00000000	H'E8201D14	32
SPI status register 5	spi_status5	R	H'00000000	H'E8201D18	32
SPI status register 6	spi_status6	R	H'00000000	H'E8201D1C	32
SPI status register 7	spi_status7	R	H'00000000	H'E8201D20	32
SPI status register 8	spi_status8	R	H'00000000	H'E8201D24	32
SPI status register 9	spi_status9	R	H'00000000	H'E8201D28	32
SPI status register 10	spi_status10	R	H'00000000	H'E8201D2C	32
SPI status register 11	spi_status11	R	H'00000000	H'E8201D30	32
SPI status register 12	spi_status12	R	H'00000000	H'E8201D34	32
SPI status register 13	spi_status13	R	H'00000000	H'E8201D38	32
SPI status register 14	spi_status14	R	H'00000000	H'E8201D3C	32
Software generation interrupt register	ICDSGIR	W	H'00000000	H'E8201F00	32
CPU interface control register	ICCICR	R/W	H'00000000	H'E8202000	32
Interrupt priority mask register	ICCPMR	R/W	H'00000000	H'E8202004	32
Binary point register	ICCBPR	R/W	H'00000002	H'E8202008	32
Interrupt acknowledge register	ICCIAR	R	H'000003FF	H'E820200C	32
End-of-interrupt register	ICCEOIR	W	-	H'E8202010	32
Running priority register	ICCRPR	R	H'000000FF	H'E8202014	32
Highest pending interrupt register	ICCHPIR	R	H'000003FF	H'E8202018	32
Aliased binary point register	ICCABPR	R/W	H'00000003	H'E820201C	32
CPU interface implementer identification register	ICCIIDR	R	H'3901043B	H'E82020FC	32

Note 1. When the NMI pin is high, becomes H'8000; when low, becomes H'0000.
Note 2. Only 0 can be written after reading 1, to clear the flag.
Note 3. Use the following expression to calculate the maximum number of interrupt IDs from the number of IT lines.
(17+1) × 32 + 1 to (17 + 2) × 32: 577 to 608

chanhvinh.com

7.3.1 Interrupt Control Register 0 (ICR0)

ICR0 is a 16-bit register that sets the input signal detection mode for the external interrupt input pin NMI, and indicates the input level at the NMI pin.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NMIL	-	-	-	-	-	-	NMIE	-	-	-	-	-	-	NMIF	-
Initial value:	*1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R/(W)*2	R

Notes: 1. 1 when the NMI pin is high, and 0 when the NMI pin is low.
2. Only 0 can be written after reading 1, to clear the flag.

Bit	Bit Name	Initial Value	R/W	Description
15	NMIL	*1	R	NMI Input Level Sets the level of the signal input at the NMI pin. The NMI pin level can be obtained by reading this bit. This bit cannot be modified. 0: Low level is input to NMI pin 1: High level is input to NMI pin
14 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	NMIE	0	R/W	NMI Edge Select Selects whether the falling or rising edge of the interrupt request signal on the NMI pin is detected. 0: Interrupt request is detected on falling edge of NMI input 1: Interrupt request is detected on rising edge of NMI input
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	NMIF	0	R/(W)*2	NMI Interrupt Request This bit indicates the status of the NMI interrupt request. This bit cannot be modified. 0: NMI interrupt request has not occurred [Clearing conditions] • Cleared by changing NMIE of ICR0 • Cleared by reading NMIF while NMIF = 1, then writing 0 to NMIF 1: NMI interrupt request is detected [Setting condition] • Edge corresponding to NMIE of ICR0 has occurred at NMI pin
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

chanhvinh.com

7.3.2 Interrupt Control Register 1 (ICR1)

ICR1 is a 16-bit register that specifies the detection mode for external interrupt input pins IRQ7 to IRQ0 individually: low level, falling edge, rising edge, or both edges.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ71S	IRQ70S	IRQ61S	IRQ60S	IRQ51S	IRQ50S	IRQ41S	IRQ40S	IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ71S	0	R/W	IRQ Sense Select These bits select whether interrupt signals corresponding to pins IRQ7 to IRQ0 are detected by a low level, falling edge, rising edge, or both edges. 00: Interrupt request is detected on low level of IRQn input 01: Interrupt request is detected on falling edge of IRQn input 10: Interrupt request is detected on rising edge of IRQn input 11: Interrupt request is detected on both edges of IRQn input
14	IRQ70S	0	R/W	
13	IRQ61S	0	R/W	
12	IRQ60S	0	R/W	
11	IRQ51S	0	R/W	
10	IRQ50S	0	R/W	
9	IRQ41S	0	R/W	
8	IRQ40S	0	R/W	
7	IRQ31S	0	R/W	
6	IRQ30S	0	R/W	
5	IRQ21S	0	R/W	
4	IRQ20S	0	R/W	
3	IRQ11S	0	R/W	
2	IRQ10S	0	R/W	
1	IRQ01S	0	R/W	
0	IRQ00S	0	R/W	

[Legend]
n = 7 to 0

chanhvinh.com

7.3.3 IRQ Interrupt Request Register (IRQRR)

IRQRR is a 16-bit register that indicates interrupt requests from external input pins IRQ7 to IRQ0. If edge detection is set for the IRQ7 to IRQ0 interrupts, writing 0 to the IRQ7F to IRQ0F bits after reading IRQ7F to IRQ0F = 1 cancels the retained interrupts.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	IRQ7F	0	R/(W)*	IRQ Interrupt Request These bits indicate the status of the IRQ7 to IRQ0 interrupt requests. Level detection: 0: IRQn interrupt request has not occurred [Clearing condition] • IRQn input is high 1: IRQn interrupt has occurred [Setting condition] • IRQn input is low Edge detection: 0: IRQn interrupt request is not detected [Clearing condition] • Cleared by reading IRQnF while IRQnF = 1, then writing 0 to IRQnF 1: IRQn interrupt request is detected [Setting condition] • Edge corresponding to IRQn1S or IRQn0S of ICR1 has occurred at IRQn pin
6	IRQ6F	0	R/(W)*	
5	IRQ5F	0	R/(W)*	
4	IRQ4F	0	R/(W)*	
3	IRQ3F	0	R/(W)*	
2	IRQ2F	0	R/(W)*	
1	IRQ1F	0	R/(W)*	
0	IRQ0F	0	R/(W)*	

[Legend]
n = 7 to 0

chanhvinh.com

7.4 Interrupt Sources

There are four types of interrupt sources: NMI, IRQ, on-chip peripheral modules, and pin interrupts. Each interrupt has a priority level (0 to 31), with 0 the highest and 31 the lowest.

7.4.1 NMI Interrupt

The NMI interrupt with the highest priority is accepted by the CPU as an FIQ exception all times. NMI interrupt requests are edge-detected, and the NMI edge select bit (NMIE) in ICR0 selects whether the rising edge or falling edge is detected. The status of the interrupt request can be checked by reading the NMI interrupt request bit (NMIF) in the ICR0. When the NMIE bit is changed, the NMI interrupt request that is retained is cleared.
When deep standby mode is entered, deep standby mode is canceled by the NMI interrupt.

7.4.2 IRQ Interrupts

IRQ interrupts are input from pins IRQ7 to IRQ0. For the IRQ interrupts, low-level, falling-edge, rising-edge, or both-edge detection can be selected individually for each pin by the IRQ sense select bits (IRQ71S to IRQ01S and IRQ70S to IRQ00S) in interrupt control register 1 (ICR1).
When using low-level sensing for IRQ interrupts, an interrupt request signal is sent to the interrupt controller while the IRQ7 to IRQ0 pins are low. An interrupt request signal is stopped being sent to the interrupt controller when the IRQ7 to IRQ0 pins are driven high. The status of the interrupt requests can be checked by reading the IRQ interrupt request bits (IRQ7F to IRQ0F) in the IRQ interrupt request register (IRQRR).
When using edge-sensing for IRQ interrupts, an interrupt request is detected due to change of the IRQ7 to IRQ0 pin states, and an interrupt request signal is sent to the interrupt controller. The result of IRQ interrupt request detection is retained until that interrupt request is accepted. Whether IRQ interrupt requests have been detected or not can be checked by reading the IRQ7F to IRQ0F bits in the IRQ interrupt request register (IRQRR). Writing 0 to these bits after reading them as 1 clears the result of IRQ interrupt request detection.
When returning from IRQ interrupt exception service routine, execute the return instruction after confirming that the interrupt request has been cleared by the IRQ interrupt request register (IRQRR) so as not to accidentally receive the interrupt request again.

chanhvinh.com

7.4.3 On-Chip Peripheral Module Interrupts

On-chip peripheral module interrupts are generated by the following on-chip peripheral modules:

- Direct memory access controller
- USB 2.0 host/function module
- Video display controller 5
- JPEG codec unit (RZ/A1LU only)
- OS timer
- Bus state controller
- Watchdog timer
- Multi-function timer pulse unit 2
- 12-bit A/D converter
- Serial sound interface
- Renesas SPDIF interface
- I^2C bus interface
- Serial communication interface with FIFO
- CAN interface
- Renesas serial peripheral interface
- IEBus™ controller (RZ/A1L only)
- CD-ROM decoder (RZ/A1L only)
- MMC host interface
- SD host interface
- Realtime clock
- SCUX
- Media local bus (RZ/A1L only)
- LIN interface (RZ/A1L only)
- Serial communication interface
- EthernetAVB (RZ/A1LU only)
- Ethernet controller
- Capture engine unit
- On-chip RAM
- Internal bus

When returning from the interrupt exception service routine for an interrupt request at the peripheral-module level, execute the return instruction after clearing the source flag at the source of the request and reading the source flag so that the interrupt request is not accidentally received again.

chanhvinh.com

7.4.4 Pin Interrupts

Pin interrupts are input from pins TINT121 to TINT0. Signals input on pins TINT121 to TINT0 are conveyed as interrupt signals regardless of mode settings and pin function settings for the general-purpose I/O port pin.
Accordingly, if pin interrupts are to be used in port mode, set the pin as an input port pin. Alternatively, if an alternative mode is selected, the alternative pin with an input function can only be used for pin interrupts. For the settings of general-purpose I/O port pins, see section 41., Ports.
For the pin interrupts, high-level or rising-edge detection can be selected individually for each pin by the interrupt configuration registers (ICDICFRn). For a description of the interrupt configuration registers (ICDICFRn), see the GIC architecture specification.

chanhvinh.com

7.5 Interrupt IDs

Table 7.3 lists the interrupt sources and their interrupt IDs, and the registers for setting the interrupt sources. Do not make settings other than those in table 7.3, otherwise, the operation cannot be guaranteed.
Each interrupt source is allocated a different interrupt ID. To control notification of the interrupt source to the CPU and reference its status, it is necessary to set and reference the following registers which correspond to given interrupt IDs.

- Interrupt security register ICDISRn
- Interrupt set-enable register ICDISERn
- Interrupt clear-enable register ICDICERn
- Interrupt set-pending register ICDISPRn
- Interrupt clear-pending register ICDICPRn
- Active bit register ICDABRn
- Interrupt configuration register ICDICFRn
- Interrupt priority register ICDIPRn
- Interrupt processor target register ICDIPTRn

For the procedure for the initial settings of the registers, see section 7.6.1 Initial Settings. For details on individual registers, see the Arm Generic Interrupt Controller Architecture Specification and the PrimeCell® Generic Interrupt Controller (PL390) Technical Reference Manual from Arm.

Table 7.3 List of Interrupt IDs

Interrupt Source					Register Allocation					
Module	Channel	Request Source Name	Interrupt Request Edge/ Level	Inter -rupt ID	ICDISRn ICDISERn ICDICERn ICDISPRn ICDICPRn ICDABRn Bit	ICDICFRn	Bit	ICDIPRn ICDIPTRn	Bit
GIC software interrupt	-	-	-	0	0	0	0	1 to 0	0	7 to 0
				1		1		3 to 2		15 to 8
				2		2		5 to 4		23 to 16
				3		3		7 to 6		31 to 24
				4		4		9 to 8	1	7 to 0
				5		5		11 to 10		15 to 8
				6		6		13 to 12		23 to 16
				7		7		15 to 14		31 to 24
				8		8		17 to 16	2	7 to 0
				9		9		19 to 18		15 to 8
				10		10		21 to 20		23 to 16
				11		11		23 to 22		31 to 24
				12		12		25 to 24	3	7 to 0
				13		13		27 to 26		15 to 8
				14		14		29 to 28		23 to 16
				15		15		31 to 30		31 to 24
CPU	0	PMUIRQ0	Level	16		16	1	1 to 0	4	7 to 0
		COMMRX0	Level	17		17		3 to 2		15 to 8
		COMMTX0	Level	18		18		5 to 4		23 to 16
		CTIIRQ0	Level	19		19		7 to 6		31 to 24

chanhvinh.com

Table 7.3 List of Interrupt IDs

Interrupt Source					Register Allocation					
Module	Channel	Request Source Name	Interrupt Request Edge/ Level	Inter -rupt ID	ICDISRn ICDISERn ICDICERn ICDISPRn ICDICPRn ICDABRn	Bit	ICDICFRn	Bit	ICDIPRn ICDIPTRn	Bit
IRQ	-	IRQ0	Level	32	1	0	2	1 to 0	8	7 to 0
		IRQ1	Level	33		1		3 to 2		15 to 8
		IRQ2	Level	34		2		5 to 4		23 to 16
		IRQ3	Level	35		3		7 to 6		31 to 24
		IRQ4	Level	36		4		9 to 8	9	7 to 0
		IRQ5	Level	37		5		11 to 10		15 to 8
		IRQ6	Level	38		6		13 to 12		23 to 16
		IRQ7	Level	39		7		15 to 14		31 to 24
Secondary cache	-	PL310ERR	Level	40		8		17 to 16	10	7 to 0
Direct memory access controller	0	DMAINT0	Edge	41		9		19 to 18		15 to 8
	1	DMAINT1	Edge	42		10		21 to 20		23 to 16
	2	DMAINT2	Edge	43		11		23 to 22		31 to 24
	3	DMAINT3	Edge	44		12		25 to 24	11	7 to 0
	4	DMAINT4	Edge	45		13		27 to 26		15 to 8
	5	DMAINT5	Edge	46		14		29 to 28		23 to 16
	6	DMAINT6	Edge	47		15		31 to 30		31 to 24
	7	DMAINT7	Edge	48		16	3	1 to 0	12	7 to 0
	8	DMAINT8	Edge	49		17		3 to 2		15 to 8
	9	DMAINT9	Edge	50		18		5 to 4		23 to 16
	10	DMAINT10	Edge	51		19		7 to 6		31 to 24
	11	DMAINT11	Edge	52		20		9 to 8	13	7 to 0
	12	DMAINT12	Edge	53		21		11 to 10		15 to 8
	13	DMAINT13	Edge	54		22		13 to 12		23 to 16
	14	DMAINT14	Edge	55		23		15 to 14		31 to 24
	15	DMAINT15	Edge	56		24		17 to 16	14	7 to 0
	-	DMAERR	Edge	57		25		19 to 18		15 to 8

chanhvinh.com

Table 7.3 List of Interrupt IDs

Interrupt Source					Register Allocation					
Module	Channel	Request Source Name	Interrupt Request Edge/ Level	Inter -rupt ID	ICDISRn ICDISERn ICDICERn ICDISPRn ICDICPRn ICDABRn	Bit	ICDICFRn	Bit	ICDIPRn ICDIPTRn	Bit
Reserved				58	1	26	3	21 to 20	14	23 to 16
				59		27		23 to 22		31 to 24
				60		28		25 to 24	15	7 to 0
				61		29		27 to 26		15 to 8
				62		30		29 to 28		23 to 16
				63		31		31 to 30		31 to 24
				64	2	0	4	1 to 0	16	7 to 0
				65		1		3 to 2		15 to 8
				66		2		5 to 4		23 to 16
				67		3		7 to 6		31 to 24
				68		4		9 to 8	17	7 to 0
				69		5		11 to 10		15 to 8
				70		6		13 to 12		23 to 16
				71		7		15 to 14		31 to 24
				72		8		17 to 16	18	7 to 0
USB 2.0 host/ function module	0	USBI0	Level	73		9		19 to 18		15 to 8
	1	USBI1	Level	74		10		21 to 20		23 to 16
Video display controller 5	-	S0_VI_VSYNC0	Level	75		11		23 to 22		31 to 24
		S0_LO_VSYNC0	Level	76		12		25 to 24	19	7 to 0
		S0_VSYNCERR0	Level	77		13		27 to 26		15 to 8
		GR3_VLINE0	Level	78		14		29 to 28		23 to 16
		S0_VFIELD0	Level	79		15		31 to 30		31 to 24
		IV1_VBUFERR0	Level	80		16	5	1 to 0	20	7 to 0
		IV3_VBUFERR0	Level	81		17		3 to 2		15 to 8
		IV5_VBUFERR0	Level	82		18		5 to 4		23 to 16
		IV6_VBUFERR0	Level	83		19		7 to 6		31 to 24
		S0_WLINE0	Level	84		20		9 to 8	21	7 to 0
Reserved				85		21		11 to 10		15 to 8
				86		22		13 to 12		23 to 16
				87		23		15 to 14		31 to 24
				88		24		17 to 16	22	7 to 0
				89		25		19 to 18		15 to 8
				90		26		21 to 20		23 to 16
				91		27		23 to 22		31 to 24
				92		28		25 to 24	23	7 to 0
				93		29		27 to 26		15 to 8
				94		30		29 to 28		23 to 16
				95		31		31 to 30		31 to 24

chanhvinh.com

Table 7.3 List of Interrupt IDs

Interrupt Source					Register Allocation					
Module	Channel	Request Source Name	Interrupt Request Edge/ Level	Inter -rupt ID	ICDISRn ICDISERn ICDICERn ICDISPRn ICDICPRn ICDABRn Bit	ICDICFRn	Bit	ICDIPRn ICDIPTRn	Bit
Reserved				96	3 0	6	1 to 0	24	7 to 0
				97	1		3 to 2		15 to 8
				98	2		5 to 4		23 to 16
				99	3		7 to 6		31 to 24
				100	4		9 to 8	25	7 to 0
				101	5		11 to 10		15 to 8
				102	6		13 to 12		23 to 16
				103	7		15 to 14		31 to 24
				104	8		17 to 16	26	7 to 0
				105	9		19 to 18		15 to 8
				106	10		21 to 20		23 to 16
				107	11		23 to 22		31 to 24
				108	12		25 to 24	27	7 to 0
				109	13		27 to 26		15 to 8
				110	14		29 to 28		23 to 16
				111	15		31 to 30		31 to 24
				112	16	7	1 to 0	28	7 to 0
				113	17		3 to 2		15 to 8
				114	18		5 to 4		23 to 16
				115	19		7 to 6		31 to 24
				116	20		9 to 8	29	7 to 0
				117	21		11 to 10		15 to 8
				118	22		13 to 12		23 to 16
				119	23		15 to 14		31 to 24
				120	24		17 to 16	30	7 to 0
				121	25		19 to 18		15 to 8
				122	26		21 to 20		23 to 16
				123	27		23 to 22		31 to 24
				124	28		25 to 24	31	7 to 0
				125	29		27 to 26		15 to 8
JPEG codec unit (RZ/A1LU only)	-	JEDI	Level	126	30		29 to 28		23 to 16
		JDTI	Level	127	31		31 to 30		31 to 24

chanhvinh.com

Table 7.3 List of Interrupt IDs

Interrupt Source					Register Allocation					
Module	Channel	Request Source Name	Interrupt Request Edge/ Level	Inter -rupt ID	ICDISRn ICDISERn ICDICERn ICDISPRn ICDICPRn ICDABRn	Bit	ICDICFRn	Bit	ICDIPRn ICDIPTRn	Bit
Reserved				128	4	0	8	1 to 0	32	7 to 0
				129		1		3 to 2		15 to 8
				130		2		5 to 4		23 to 16
				131		3		7 to 6		31 to 24
				132		4		9 to 8	33	7 to 0
				133		5		11 to 10		15 to 8
OS timer	0	OSTM0TINT	Edge	134		6		13 to 12		23 to 16
	1	OSTM1TINT	Edge	135		7		15 to 14		31 to 24
Bus state controller	-	CMI	Level	136		8		17 to 16	34	7 to 0
		WTOUT	Level	137		9		19 to 18		15 to 8
Watchdog timer	-	ITI	Level	138		10		21 to 20		23 to 16
Multi-function timer pulse unit 2	0	TGI0A	Level	139	4	11		23 to 22		31 to 24
		TGI0B	Level	140		12		25 to 24	35	7 to 0
		TGI0C	Level	141		13		27 to 26		15 to 8
		TGI0D	Level	142		14		29 to 28		23 to 16
		TGI0V	Level	143		15		31 to 30		31 to 24
		TGI0E	Level	144		16	9	1 to 0	36	7 to 0
		TGI0F	Level	145		17		3 to 2		15 to 8
	1	TGI1A	Level	146		18		5 to 4		23 to 16
		TGI1B	Level	147		19		7 to 6		31 to 24
		TGI1V	Level	148		20		9 to 8	37	7 to 0
		TGI1U	Level	149		21		11 to 10		15 to 8
	2	TGI2A	Level	150		22		13 to 12		23 to 16
		TGI2B	Level	151		23		15 to 14		31 to 24
		TGI2V	Level	152		24		17 to 16	38	7 to 0
		TGI2U	Level	153		25		19 to 18		15 to 8
	3	TGI3A	Level	154		26		21 to 20		23 to 16
		TGI3B	Level	155		27		23 to 22		31 to 24
		TGI3C	Level	156		28		25 to 24	39	7 to 0
		TGI3D	Level	157		29		27 to 26		15 to 8
		TGI3V	Level	158		30		29 to 28		23 to 16
	4	TGI4A	Level	159		31		31 to 30		31 to 24
		TGI4B	Level	160	5	0	10	1 to 0	40	7 to 0
		TGI4C	Level	161		1		3 to 2		15 to 8
		TGI4D	Level	162		2		5 to 4		23 to 16
		TGI4V	Level	163		3		7 to 6		31 to 24

chanhvinh.com

Table 7.3 List of Interrupt IDs

Interrupt Source					Register Allocation					
Module	Channel	Request Source Name	Interrupt Request Edge/ Level	Inter -rupt ID	ICDISRn ICDISERn ICDICERn ICDISPRn ICDICPRn ICDABRn	Bit	ICDICFRn	Bit	ICDIPRn ICDIPTRn	Bit
Reserved				164	5	4	10	9 to 8	41	7 to 0
				165		5		11 to 10		15 to 8
				166		6		13 to 12		23 to 16
				167		7		15 to 14		31 to 24
				168		8		17 to 16	42	7 to 0
				169		9		19 to 18		15 to 8
12-bit A/D converter	-	ADI	Level	170		10		21 to 20		23 to 16
		LMTI	Level	171		11		23 to 22		31 to 24
Serial sound interface	0	SSII0	Level	172		12		25 to 24	43	7 to 0
		SSIRXI0	Level	173		13		27 to 26		15 to 8
		SSITXI0	Level	174		14		29 to 28		23 to 16
	1	SSII1	Level	175		15		31 to 30		31 to 24
		SSIRXI1	Level	176		16	11	1 to 0	44	7 to 0
		SSITXI1	Level	177		17		3 to 2		15 to 8
	2	SSII2	Level	178		18		5 to 4		23 to 16
		SSIRTI2	Level	179		19		7 to 6		31 to 24
	3	SSII3	Level	180		20		9 to 8	45	7 to 0
		SSIRXI3	Level	181		21		11 to 10		15 to 8
		SSITXI3	Level	182		22		13 to 12		23 to 16
Reserved				183		23		15 to 14		31 to 24
				184		24		17 to 16	46	7 to 0
				185		25		19 to 18		15 to 8
				186		26		21 to 20		23 to 16
				187		27		23 to 22		31 to 24
Renesas SPDIF interface	-	SPDIFI	Level	188		28		25 to 24	47	7 to 0

chanhvinh.com

Table 7.3 List of Interrupt IDs

Interrupt Source					Register Allocation					
Module	Channel	Request Source Name	Interrupt Request Edge/ Level	Inter -rupt ID	ICDISRn ICDISERn ICDICERn ICDISPRn ICDICPRn ICDABRn	Bit	ICDICFRn	Bit	ICDIPRn ICDIPTRn	Bit
I2C bus interface	0	INTIICTEI0	Level	189	5	29	11	27 to 26	47	15 to 8
		INTIICRI0	Edge	190		30		29 to 28		23 to 16
		INTIICTI0	Edge	191		31		31 to 30		31 to 24
		INTIICSPI0	Level	192	6	0	12	1 to 0	48	7 to 0
		INTIICSTI0	Level	193		1		3 to 2		15 to 8
		INTIICNAKI0	Level	194		2		5 to 4		23 to 16
		INTIICALI0	Level	195		3		7 to 6		31 to 24
		INTIICTMOI0	Level	196		4		9 to 8	49	7 to 0
	1	INTIICTEI1	Level	197		5		11 to 10		15 to 8
		INTIICRI1	Edge	198		6		13 to 12		23 to 16
		INTIICTI1	Edge	199		7		15 to 14		31 to 24
		INTIICSPI1	Level	200		8		17 to 16	50	7 to 0
		INTIICSTI1	Level	201		9		19 to 18		15 to 8
		INTIICNAKI1	Level	202		10		21 to 20		23 to 16
		INTIICALI1	Level	203		11		23 to 22		31 to 24
		INTIICTMOI1	Level	204		12		25 to 24	51	7 to 0
	2	INTIICTEI2	Level	205		13		27 to 26		15 to 8
		INTIICRI2	Edge	206		14		29 to 28		23 to 16
		INTIICTI2	Edge	207		15		31 to 30		31 to 24
		INTIICSPI2	Level	208		16	13	1 to 0	52	7 to 0
		INTIICSTI2	Level	209		17		3 to 2		15 to 8
		INTIICNAKI2	Level	210		18		5 to 4		23 to 16
		INTIICALI2	Level	211		19		7 to 6		31 to 24
		INTIICTMOI2	Level	212		20		9 to 8	53	7 to 0
	3	INTIICTEI3	Level	213		21		11 to 10		15 to 8
		INTIICRI3	Edge	214		22		13 to 12		23 to 16
		INTIICTI3	Edge	215		23		15 to 14		31 to 24
		INTIICSPI3	Level	216		24		17 to 16	54	7 to 0
		INTIICSTI3	Level	217		25		19 to 18		15 to 8
		INTIICNAKI3	Level	218		26		21 to 20		23 to 16
		INTIICALI3	Level	219		27		23 to 22		31 to 24
		INTIICTMOI3	Level	220		28		25 to 24	55	7 to 0

chanhvinh.com

Table 7.3 List of Interrupt IDs

Interrupt Source					Register Allocation					
Module	Channel	Request Source Name	Interrupt Request Edge/ Level	Inter -rupt ID	ICDISRn ICDISERn ICDICERn ICDISPRn ICDICPRn ICDABRn	Bit	ICDICFRn	Bit	ICDIPRn ICDIPTRn	Bit
Serial communi-cation interface with FIFO	0	BRI0	Level	221	6	29	13	27 to 26	55	15 to 8
		ERI0	Level	222		30		29 to 28		23 to 16
		RXI0	Level	223		31		31 to 30		31 to 24
		TXI0	Level	224	7	0	14	1 to 0	56	7 to 0
	1	BRI1	Level	225		1		3 to 2		15 to 8
		ERI1	Level	226		2		5 to 4		23 to 16
		RXI1	Level	227		3		7 to 6		31 to 24
		TXI1	Level	228		4		9 to 8	57	7 to 0
	2	BRI2	Level	229		5		11 to 10		15 to 8
		ERI2	Level	230		6		13 to 12		23 to 16
		RXI2	Level	231		7		15 to 14		31 to 24
		TXI2	Level	232		8		17 to 16	58	7 to 0
	3	BRI3	Level	233		9		19 to 18		15 to 8
		ERI3	Level	234		10		21 to 20		23 to 16
		RXI3	Level	235		11		23 to 22		31 to 24
		TXI3	Level	236		12		25 to 24	59	7 to 0
	4	BRI4	Level	237		13		27 to 26		15 to 8
		ERI4	Level	238		14		29 to 28		23 to 16
		RXI4	Level	239		15		31 to 30		31 to 24
		TXI4	Level	240		16	15	1 to 0	60	7 to 0
Reserved				241		17		3 to 2		15 to 8
				242		18		5 to 4		23 to 16
				243		19		7 to 6		31 to 24
				244		20		9 to 8	61	7 to 0
				245		21		11 to 10		15 to 8
				246		22		13 to 12		23 to 16
				247		23		15 to 14		31 to 24
				248		24		17 to 16	62	7 to 0
				249		25		19 to 18		15 to 8
				250		26		21 to 20		23 to 16
				251		27		23 to 22		31 to 24
				252		28		25 to 24	63	7 to 0

chanhvinh.com

Table 7.3 List of Interrupt IDs

Interrupt Source					Register Allocation					
Module	Channel	Request Source Name	Interrupt Request Edge/ Level	Inter-rupt ID	ICDISRn ICDISERn ICDICERn ICDISPRn ICDICPRn ICDABRn	Bit	ICDICFRn	Bit	ICDIPRn ICDIPTRn	Bit
CAN interface	Com-mon	INTRCANGERR	Level	253	7	29	15	27 to 26	63	15 to 8
		INTRCANGRECC	Level	254		30		29 to 28		23 to 16
	0	INTRCAN0REC	Level	255		31		31 to 30		31 to 24
		INTRCAN0ERR	Level	256	8	0	16	1 to 0	64	7 to 0
		INTRCAN0TRX	Level	257		1		3 to 2		15 to 8
	1	INTRCAN1REC	Level	258		2		5 to 4		23 to 16
		INTRCAN1ERR	Level	259		3		7 to 6		31 to 24
		INTRCAN1TRX	Level	260		4		9 to 8	65	7 to 0
Reserved				261		5		11 to 10		15 to 8
				262		6		13 to 12		23 to 16
				263		7		15 to 14		31 to 24
				264		8		17 to 16	66	7 to 0
				265		9		19 to 18		15 to 8
				266		10		21 to 20		23 to 16
				267		11		23 to 22		31 to 24
				268		12		25 to 24	67	7 to 0
				269		13		27 to 26		15 to 8
Renesas serial peripheral interface	0	SPEI0	Level	270		14		29 to 28		23 to 16
		SPRI0	Level	271		15		31 to 30		31 to 24
		SPTI0	Level	272		16	17	1 to 0	68	7 to 0
	1	SPEI1	Level	273		17		3 to 2		15 to 8
		SPRI1	Level	274		18		5 to 4		23 to 16
		SPTI1	Level	275		19		7 to 6		31 to 24
	2	SPEI2	Level	276		20		9 to 8	69	7 to 0
		SPRI2	Level	277		21		11 to 10		15 to 8
		SPTI2	Level	278		22		13 to 12		23 to 16
Reserved				279		23		15 to 14		31 to 24
				280		24		17 to 16	70	7 to 0
				281		25		19 to 18		15 to 8
				282		26		21 to 20		23 to 16
				283		27		23 to 22		31 to 24
				284		28		25 to 24	71	7 to 0

chanhvinh.com

Table 7.3 List of Interrupt IDs

Interrupt Source					Register Allocation					
Module	Channel	Request Source Name	Interrupt Request Edge/ Level	Inter -rupt ID	ICDISRn ICDISERn ICDICERn ICDISPRn ICDICPRn ICDABRn	Bit	ICDICFRn	Bit	ICDIPRn ICDIPTRn	Bit
IEBus™ controller (RZ/A1L only)	-	IEBBTD	Edge	285	8	29	17	27 to 26	71	15 to 8
		IEBBTERR	Edge	286		30		29 to 28		23 to 16
		IEBBTSTA	Edge	287		31		31 to 30		31 to 24
		IEBBTV	Edge	288	9	0	18	1 to 0	72	7 to 0
CD-ROM decoder (RZ/A1L only)	-	ISY	Level	289		1		3 to 2		15 to 8
		IERR	Level	290		2		5 to 4		23 to 16
		ITARG	Level	291		3		7 to 6		31 to 24
		ISEC	Level	292		4		9 to 8	73	7 to 0
		IBUF	Level	293		5		11 to 10		15 to 8
		IREADY	Level	294		6		13 to 12		23 to 16
Reserved				295		7		15 to 14		31 to 24
				296		8		17 to 16	74	7 to 0
				297		9		19 to 18		15 to 8
				298		10		21 to 20		23 to 16
MMC host interface	-	MMC0	Level	299		11		23 to 22		31 to 24
		MMC1	Level	300		12		25 to 24	75	7 to 0
		MMC2	Level	301		13		27 to 26		15 to 8
SD host interface	0	SDHI0_3	Level	302		14		29 to 28		23 to 16
		SDHI0_0	Level	303		15		31 to 30		31 to 24
		SDHI0_1	Level	304		16	19	1 to 0	76	7 to 0
	1	SDHI1_3	Level	305		17		3 to 2		15 to 8
		SDHI1_0	Level	306		18		5 to 4		23 to 16
		SDHI1_1	Level	307		19		7 to 6		31 to 24
Realtime clock	-	ARM	Level	308		20		9 to 8	77	7 to 0
		PRD	Level	309		21		11 to 10		15 to 8
		CUP	Level	310		22		13 to 12		23 to 16
SCUX	-	SCUAI0	Level	311		23		15 to 14		31 to 24
		SCUAI1	Level	312		24		17 to 16	78	7 to 0
		SCUFDI0	Level	313		25		19 to 18		15 to 8
		SCUFDI1	Level	314		26		21 to 20		23 to 16
		SCUFDI2	Level	315		27		23 to 22		31 to 24
		SCUFDI3	Level	316		28		25 to 24	79	7 to 0
		SCUFUI0	Level	317		29		27 to 26		15 to 8
		SCUFUI1	Level	318		30		29 to 28		23 to 16
		SCUFUI2	Level	319		31		31 to 30		31 to 24
		SCUFUI3	Level	320	10	0	20	1 to 0	80	7 to 0
		SCUDVI0	Level	321		1		3 to 2		15 to 8
		SCUDVI1	Level	322		2		5 to 4		23 to 16
		SCUDVI2	Level	323		3		7 to 6		31 to 24
		SCUDVI3	Level	324		4		9 to 8	81	7 to 0

chanhvinh.com

Table 7.3 List of Interrupt IDs

Interrupt Source					Register Allocation					
Module	Channel	Request Source Name	Interrupt Request Edge/ Level	Inter -rupt ID	ICDISRn ICDISERn ICDICERn ICDISPRn ICDICPRn ICDABRn	Bit	ICDICFRn	Bit	ICDIPRn ICDIPTRn	Bit
Media local bus (RZ/A1L only)	-	MLB_CINT	Level	325	10	5	20	11 to 10	81	15 to 8
		MLB_SINT	Level	326		6		13 to 12		23 to 16
Reserved				327		7		15 to 14		31 to 24
				328		8		17 to 16	82	7 to 0
				329		9		19 to 18		15 to 8
				330		10		21 to 20		23 to 16
LIN interface (RZ/A1L only)	0	LIN0_INT_T	Edge	331		11		23 to 22		31 to 24
		LIN0_INT_R	Edge	332		12		25 to 24	83	7 to 0
		LIN0_INT_S	Edge	333		13		27 to 26		15 to 8
		LIN0_INT_M	Edge	334		14		29 to 28		23 to 16
Reserved				335		15		31 to 30		31 to 24
				336		16	21	1 to 0	84	7 to 0
				337		17		3 to 2		15 to 8
				338		18		5 to 4		23 to 16
				339		19		7 to 6		31 to 24
				340		20		9 to 8	85	7 to 0
				341		21		11 to 10		15 to 8
				342		22		13 to 12		23 to 16
				343		23		15 to 14		31 to 24
				344		24		17 to 16	86	7 to 0
				345		25		19 to 18		15 to 8
				346		26		21 to 20		23 to 16
Serial communi-cation interface	0	ERI0	Level	347		27		23 to 22		31 to 24
		RXI0	Edge	348		28		25 to 24	87	7 to 0
		TXI0	Edge	349		29		27 to 26		15 to 8
		TEI0	Level	350		30		29 to 28		23 to 16
	1	ERI1	Level	351		31		31 to 30		31 to 24
		RXI1	Edge	352	11	0	22	1 to 0	88	7 to 0
		TXI1	Edge	353		1		3 to 2		15 to 8
		TEI1	Level	354		2		5 to 4		23 to 16
Ethernet AVB (RZ/A1LU only)	-	AVBI_DATA	Level	355		3		7 to 6		31 to 24
		AVBI_ERROR	Level	356		4		9 to 8	89	7 to 0
		AVBI_MANAGE	Level	357		5		11 to 10		15 to 8
		AVBI_MAC	Level	358		6		13 to 12		23 to 16
Ethernet controller	-	ETHERI	Level	359		7		15 to 14		31 to 24
Reserved		AVBI_ERROR		360		8		17 to 16	90	7 to 0
				361		9		19 to 18		15 to 8
				362		10		21 to 20		23 to 16
				363		11		23 to 22		31 to 24

chanhvinh.com

Table 7.3 List of Interrupt IDs

Interrupt Source					Register Allocation					
Module	Channel	Request Source Name	Interrupt Request Edge/ Level	Inter -rupt ID	ICDISRn ICDISERn ICDICERn ICDISPRn ICDICPRn ICDABRn	Bit	ICDICFRn	Bit	ICDIPRn ICDIPTRn	Bit
Capture engine unit	-	CEUI	Level	364	11	12	22	25 to 24	91	7 to 0
Reserved				365		13		27 to 26		15 to 8
				366		14		29 to 28		23 to 16
				367		15		31 to 30		31 to 24
				368		16	23	1 to 0	92	7 to 0
				369		17		3 to 2		15 to 8
				370		18		5 to 4		23 to 16
				371		19		7 to 6		31 to 24
				372		20		9 to 8	93	7 to 0
				373		21		11 to 10		15 to 8
				374		22		13 to 12		23 to 16
				375		23		15 to 14		31 to 24
				376		24		17 to 16	94	7 to 0
				377		25		19 to 18		15 to 8
				378		26		21 to 20		23 to 16
				379		27		23 to 22		31 to 24
				380		28		25 to 24	95	7 to 0
Internal bus		H2XMLB_ERRINT	Level	381		29		27 to 26		15 to 8
		H2XIC1_ERRINT	Level	382		30		29 to 28		23 to 16
		X2HPERI1_ERRINT	Level	383		31		31 to 30		31 to 24
		X2HPERI2_ERRINT	Level	384	12	0	24	1 to 0	96	7 to 0
		X2HPERI34_ERRINT	Level	385		1		3 to 2		15 to 8
		X2HPERI5_ERRINT	Level	386		2		5 to 4		23 to 16
		X2HPERI67_ERRINT	Level	387		3		7 to 6		31 to 24
		X2HDBGR_ERRINT	Level	388		4		9 to 8	97	7 to 0
		X2HBSC_ERRINT	Level	389		5		11 to 10		15 to 8
		X2HSPI1_ERRINT	Level	390		6		13 to 12		23 to 16
		X2HSPI2_ERRINT	Level	391		7		15 to 14		31 to 24
		PRRI	Level	392		8		17 to 16	98	7 to 0
Reserved				393		9		19 to 18		15 to 8
				394		10		21 to 20		23 to 16
				395		11		23 to 22		31 to 24
				396		12		25 to 24	99	7 to 0
				397		13		27 to 26		15 to 8
				398		14		29 to 28		23 to 16
				399		15		31 to 30		31 to 24

chanhvinh.com

Table 7.3 List of Interrupt IDs

Interrupt Source					Register Allocation					
Module	Channel	Request Source Name	Interrupt Request Edge/ Level	Inter -rupt ID	ICDISRn ICDISERn ICDICERn ICDISPRn ICDICPRn ICDABRn	Bit	ICDICFRn	Bit	ICDIPRn ICDIPTRn	Bit
Reserved				400	12	16	25	1 to 0	100	7 to 0
				401		17		3 to 2		15 to 8
				402		18		5 to 4		23 to 16
				403		19		7 to 6		31 to 24
				404		20		9 to 8	101	7 to 0
				405		21		11 to 10		15 to 8
				406		22		13 to 12		23 to 16
				407		23		15 to 14		31 to 24
				408		24		17 to 16	102	7 to 0
				409		25		19 to 18		15 to 8
				410		26		21 to 20		23 to 16
				411		27		23 to 22		31 to 24
				412		28		25 to 24	103	7 to 0
				413		29		27 to 26		15 to 8
				414		30		29 to 28		23 to 16
				415		31		31 to 30		31 to 24
Pin interrupts Note: The port name is given in the Channel column for the pin interrupts.	JP0_0	TINT0	Edge/Level	416	13	0	26	1 to 0	104	7 to 0
	JP0_1	TINT1	Edge/Level	417		1		3 to 2		15 to 8
	P0_0	TINT2	Edge/Level	418		2		5 to 4		23 to 16
	P0_1	TINT3	Edge/Level	419		3		7 to 6		31 to 24
	P0_2	TINT4	Edge/Level	420		4		9 to 8	105	7 to 0
	P0_3	TINT5	Edge/Level	421		5		11 to 10		15 to 8
	P1_0	TINT6	Edge/Level	422		6		13 to 12		23 to 16
	P1_1	TINT7	Edge/Level	423		7		15 to 14		31 to 24
	P1_2	TINT8	Edge/Level	424		8		17 to 16	106	7 to 0
	P1_3	TINT9	Edge/Level	425		9		19 to 18		15 to 8
	P1_4	TINT10	Edge/Level	426		10		21 to 20		23 to 16
	P1_5	TINT11	Edge/Level	427		11		23 to 22		31 to 24

chanhvinh.com

Table 7.3 List of Interrupt IDs

Interrupt Source					Register Allocation					
Module	Channel	Request Source Name	Interrupt Request Edge/ Level	Inter -rupt ID	ICDISRn ICDISERn ICDICERn ICDISPRn ICDICPRn ICDABRn	Bit	ICDICFRn	Bit	ICDIPRn ICDIPTRn	Bit
Pin interrupts Note: The port name is given in the Channel column for the pin interrupts.	P1_6	TINT12	Edge/Level	428	13	12	26	25 to 24	107	7 to 0
	P1_7	TINT13	Edge/Level	429		13		27 to 26		15 to 8
	P1_8	TINT14	Edge/Level	430		14		29 to 28		23 to 16
	P1_9	TINT15	Edge/Level	431		15		31 to 30		31 to 24
	P1_10	TINT16	Edge/Level	432		16	27	1 to 0	108	7 to 0
	P1_11	TINT17	Edge/Level	433		17		3 to 2		15 to 8
	P1_12	TINT18	Edge/Level	434		18		5 to 4		23 to 16
	P1_13	TINT19	Edge/Level	435		19		7 to 6		31 to 24
	P1_14	TINT20	Edge/Level	436		20		9 to 8	109	7 to 0
	P1_15	TINT21	Edge/Level	437		21		11 to 10		15 to 8
	P2_0	TINT22	Edge/Level	438		22		13 to 12		23 to 16
	P2_1	TINT23	Edge/Level	439		23		15 to 14		31 to 24
	P2_2	TINT24	Edge/Level	440		24		17 to 16	110	7 to 0
	P2_3	TINT25	Edge/Level	441		25		19 to 18		15 to 8
	P2_4	TINT26	Edge/Level	442		26		21 to 20		23 to 16
	P2_5	TINT27	Edge/Level	443		27		23 to 22		31 to 24
	P2_6	TINT28	Edge/Level	444		28		25 to 24	111	7 to 0
	P2_7	TINT29	Edge/Level	445		29		27 to 26		15 to 8
	P2_8	TINT30	Edge/Level	446		30		29 to 28		23 to 16
	P2_9	TINT31	Edge/Level	447		31		31 to 30		31 to 24
	P3_0	TINT32	Edge/Level	448	14	0	28	1 to 0	112	7 to 0
	P3_1	TINT33	Edge/Level	449		1		3 to 2		15 to 8
	P3_2	TINT34	Edge/Level	450		2		5 to 4		23 to 16
	P3_3	TINT35	Edge/Level	451		3		7 to 6		31 to 24
	P3_4	TINT36	Edge/Level	452		4		9 to 8	113	7 to 0
	P3_5	TINT37	Edge/Level	453		5		11 to 10		15 to 8
	P3_6	TINT38	Edge/Level	454		6		13 to 12		23 to 16
	P3_7	TINT39	Edge/Level	455		7		15 to 14		31 to 24
	P3_8	TINT40	Edge/Level	456		8		17 to 16	114	7 to 0
	P3_9	TINT41	Edge/Level	457		9		19 to 18		15 to 8
	P3_10	TINT42	Edge/Level	458		10		21 to 20		23 to 16
	P3_11	TINT43	Edge/Level	459		11		23 to 22		31 to 24
	P3_12	TINT44	Edge/Level	460		12		25 to 24	115	7 to 0
	P3_13	TINT45	Edge/Level	461		13		27 to 26		15 to 8
	P3_14	TINT46	Edge/Level	462		14		29 to 28		23 to 16
	P3_15	TINT47	Edge/Level	463		15		31 to 30		31 to 24

chanhvinh.com

Table 7.3 List of Interrupt IDs

Interrupt Source					Register Allocation					
Module	Channel	Request Source Name	Interrupt Request Edge/ Level	Inter -rupt ID	ICDISRn ICDISERn ICDICERn ICDISPRn ICDICPRn ICDABRn	Bit	ICDICFRn	Bit	ICDIPRn ICDIPTRn	Bit
Pin interrupts Note: The port name is given in the Channel column for the pin interrupts.	P4_0	TINT48	Edge/Level	464	14	16	29	1 to 0	116	7 to 0
	P4_1	TINT49	Edge/Level	465		17		3 to 2		15 to 8
	P4_2	TINT50	Edge/Level	466		18		5 to 4		23 to 16
	P4_3	TINT51	Edge/Level	467		19		7 to 6		31 to 24
	P4_4	TINT52	Edge/Level	468		20		9 to 8	117	7 to 0
	P4_5	TINT53	Edge/Level	469		21		11 to 10		15 to 8
	P4_6	TINT54	Edge/Level	470		22		13 to 12		23 to 16
	P4_7	TINT55	Edge/Level	471		23		15 to 14		31 to 24
	P5_0	TINT56	Edge/Level	472		24		17 to 16	118	7 to 0
	P5_1	TINT57	Edge/Level	473		25		19 to 18		15 to 8
	P5_2	TINT58	Edge/Level	474		26		21 to 20		23 to 16
	P5_3	TINT59	Edge/Level	475		27		23 to 22		31 to 24
	P5_4	TINT60	Edge/Level	476		28		25 to 24	119	7 to 0
	P5_5	TINT61	Edge/Level	477		29		27 to 26		15 to 8
	P5_6	TINT62	Edge/Level	478		30		29 to 28		23 to 16
	P5_7	TINT63	Edge/Level	479		31		31 to 30		31 to 24
	P5_8	TINT64	Edge/Level	480	15	0	30	1 to 0	120	7 to 0
	P5_9	TINT65	Edge/Level	481		1		3 to 2		15 to 8
	P5_10	TINT66	Edge/Level	482		2		5 to 4		23 to 16
	P5_11	TINT67	Edge/Level	483		3		7 to 6		31 to 24
	P5_12	TINT68	Edge/Level	484		4		9 to 8	121	7 to 0
	P5_13	TINT69	Edge/Level	485		5		11 to 10		15 to 8
	P5_14	TINT70	Edge/Level	486		6		13 to 12		23 to 16
	P5_15	TINT71	Edge/Level	487		7		15 to 14		31 to 24
	P6_0	TINT72	Edge/Level	488		8		17 to 16	122	7 to 0
	P6_1	TINT73	Edge/Level	489		9		19 to 18		15 to 8
	P6_2	TINT74	Edge/Level	490		10		21 to 20		23 to 16
	P6_3	TINT75	Edge/Level	491		11		23 to 22		31 to 24
	P6_4	TINT76	Edge/Level	492		12		25 to 24	123	7 to 0
	P6_5	TINT77	Edge/Level	493		13		27 to 26		15 to 8
	P6_6	TINT78	Edge/Level	494		14		29 to 28		23 to 16
	P6_7	TINT79	Edge/Level	495		15		31 to 30		31 to 24

chanhvinh.com

Table 7.3 List of Interrupt IDs

Interrupt Source					Register Allocation					
Module	Channel	Request Source Name	Interrupt Request Edge/ Level	Inter -rupt ID	ICDISRn ICDISERn ICDICERn ICDISPRn ICDICPRn ICDABRn	Bit	ICDICFRn	Bit	ICDIPRn ICDIPTRn	Bit
Pin interrupts Note: The port name is given in the Channel column for the pin interrupts.	P6_8	TINT80	Edge/Level	496	15	16	31	1 to 0	124	7 to 0
	P6_9	TINT81	Edge/Level	497		17		3 to 2		15 to 8
	P6_10	TINT82	Edge/Level	498		18		5 to 4		23 to 16
	P6_11	TINT83	Edge/Level	499		19		7 to 6		31 to 24
	P6_12	TINT84	Edge/Level	500		20		9 to 8	125	7 to 0
	P6_13	TINT85	Edge/Level	501		21		11 to 10		15 to 8
	P6_14	TINT86	Edge/Level	502		22		13 to 12		23 to 16
	P6_15	TINT87	Edge/Level	503		23		15 to 14		31 to 24
	P7_0	TINT88	Edge/Level	504		24		17 to 16	126	7 to 0
	P7_1	TINT89	Edge/Level	505		25		19 to 18		15 to 8
	P7_2	TINT90	Edge/Level	506		26		21 to 20		23 to 16
	P7_3	TINT91	Edge/Level	507		27		23 to 22		31 to 24
	P7_4	TINT92	Edge/Level	508		28		25 to 24	127	7 to 0
	P7_5	TINT93	Edge/Level	509		29		27 to 26		15 to 8
	P7_6	TINT94	Edge/Level	510		30		29 to 28		23 to 16
	P7_7	TINT95	Edge/Level	511		31		31 to 30		31 to 24
	P7_8	TINT96	Edge/Level	512	16	0	32	1 to 0	128	7 to 0
	P7_9	TINT97	Edge/Level	513		1		3 to 2		15 to 8
	P7_10	TINT98	Edge/Level	514		2		5 to 4		23 to 16
	P7_11	TINT99	Edge/Level	515		3		7 to 6		31 to 24
	P8_0	TINT100	Edge/Level	516		4		9 to 8	129	7 to 0
	P8_1	TINT101	Edge/Level	517		5		11 to 10		15 to 8
	P8_2	TINT102	Edge/Level	518		6		13 to 12		23 to 16
	P8_3	TINT103	Edge/Level	519		7		15 to 14		31 to 24

chanhvinh.com

Table 7.3 List of Interrupt IDs

Interrupt Source					Register Allocation					
Module	Channel	Request Source Name	Interrupt Request Edge/ Level	Inter -rupt ID	ICDISRn ICDISERn ICDICERn ICDISPRn ICDICPRn ICDABRn	Bit	ICDICFRn	Bit	ICDIPRn ICDIPTRn	Bit
Pin interrupts Note: The port name is given in the Channel column for the pin interrupts.	P8_4	TINT104	Edge/Level	520	16	8	32	17 to 16	130	7 to 0
	P8_5	TINT105	Edge/Level	521		9		19 to 18		15 to 8
	P8_6	TINT106	Edge/Level	522		10		21 to 20		23 to 16
	P8_7	TINT107	Edge/Level	523		11		23 to 22		31 to 24
	P8_8	TINT108	Edge/Level	524		12		25 to 24	131	7 to 0
	P8_9	TINT109	Edge/Level	525		13		27 to 26		15 to 8
	P8_10	TINT110	Edge/Level	526		14		29 to 28		23 to 16
	P8_11	TINT111	Edge/Level	527		15		31 to 30		31 to 24
	P8_12	TINT112	Edge/Level	528		16	33	1 to 0	132	7 to 0
	P8_13	TINT113	Edge/Level	529		17		3 to 2		15 to 8
	P8_14	TINT114	Edge/Level	530		18		5 to 4		23 to 16
	P8_15	TINT115	Edge/Level	531		19		7 to 6		31 to 24
	P9_0	TINT116	Edge/Level	532		20		9 to 8	133	7 to 0
	P9_1	TINT117	Edge/Level	533		21		11 to 10		15 to 8
	P9_2	TINT118	Edge/Level	534		22		13 to 12		23 to 16
	P9_3	TINT119	Edge/Level	535		23		15 to 14		31 to 24
	P9_4	TINT120	Edge/Level	536		24		17 to 16	134	7 to 0
	P9_5	TINT121	Edge/Level	537		25		19 to 18		15 to 8

chanhvinh.com

7.6 Operation

7.6.1 Initial Settings

For details on the registers for making initial settings and the procedures for settings in general, see the Arm Generic Interrupt Controller Architecture Specification and the PrimeCell® Generic Interrupt Controller (PL390) Technical Reference Manual from Arm. Figure 7.2 illustrates the flow of initial settings.

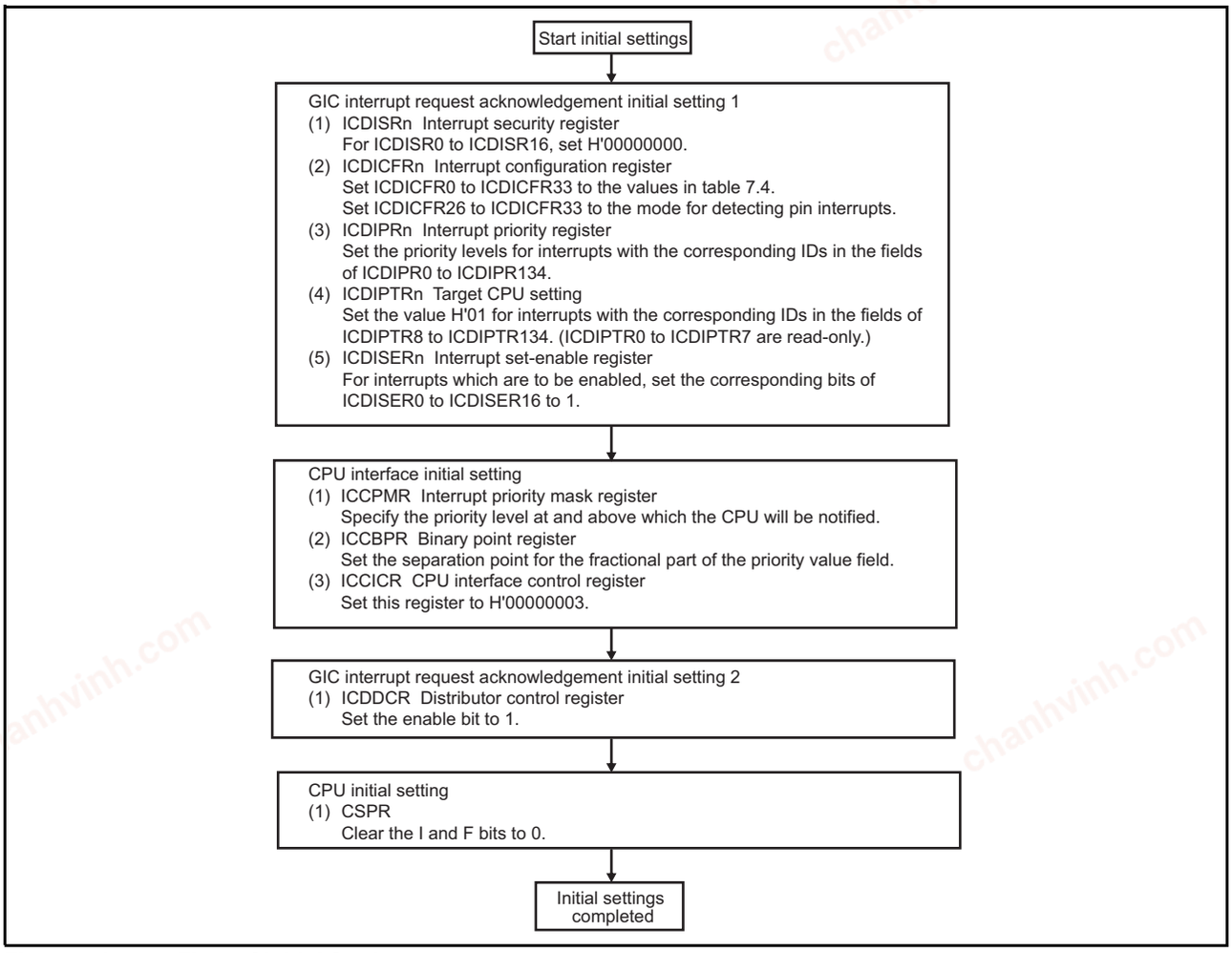


Figure 7.2 Flow of Initial Settings

chanhvinh.com

Table 7.4 ICDICFRn Interrupt Configuration Register Settings

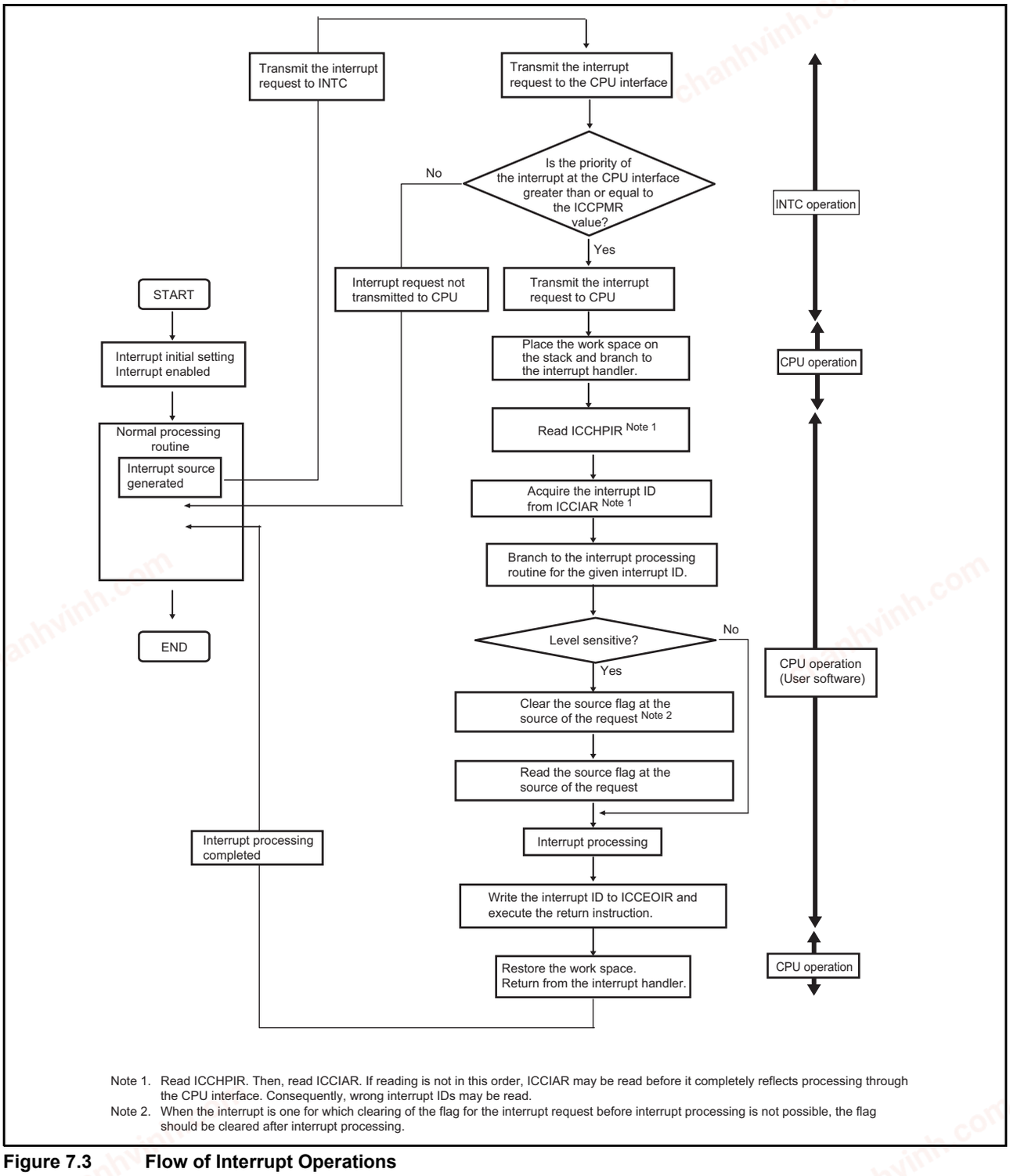
Register Name	Setting	Interrupt ID
ICDICFR0	H'AAAAAAAA	15 to 0
ICDICFR1	H'00000055	31 to 16
ICDICFR2	H'FFFD5555	47 to 32
ICDICFR3	H'555FFFFF	63 to 48
ICDICFR4	H'55555555	79 to 64
ICDICFR5	H'55555555	95 to 80
ICDICFR6	H'55555555	111 to 96
ICDICFR7	H'55555555	127 to 112
ICDICFR8	H'5555F555	143 to 128
ICDICFR9	H'55555555	159 to 144
ICDICFR10	H'55555555	175 to 160
ICDICFR11	H'F5555555	191 to 176
ICDICFR12	H'F555F555	207 to 192
ICDICFR13	H'5555F555	223 to 208
ICDICFR14	H'55555555	239 to 224
ICDICFR15	H'55555555	255 to 240
ICDICFR16	H'55555555	271 to 256
ICDICFR17	H'FD555555	287 to 272
ICDICFR18	H'55555557	303 to 288
ICDICFR19	H'55555555	319 to 304
ICDICFR20	H'7FD55555	335 to 320
ICDICFR21	H'5F555555	351 to 336
ICDICFR22	H'FD55555F	367 to 352
ICDICFR23	H'55555557	383 to 368
ICDICFR24	H'55555555	399 to 384
ICDICFR25	H'55555555	415 to 400
ICDICFR26*1	H'55555555	431 to 416
ICDICFR27*1	H'55555555	447 to 432
ICDICFR28*1	H'55555555	463 to 448
ICDICFR29*1	H'55555555	479 to 464
ICDICFR30*1	H'55555555	495 to 480
ICDICFR31*1	H'55555555	511 to 496
ICDICFR32*1	H'55555555	527 to 512
ICDICFR33*1	H'55555555	537 to 528

Note 1. Edge or level detection can be selected for IDs corresponding to pin interrupts. The settings in the above table select level detection.

chanhvinh.com

7.6.2 Flow of Interrupt Operations

For details on operation involved in interrupt generation, see the ArmGeneric Interrupt Controller Architecture Specification and the PrimeCell® Generic Interrupt Controller (PL390) Technical Reference Manual from Arm. Figure 7.3 shows the flow of interrupt operations.



Note 1. Read ICCHPIR. Then, read ICCIAR. If reading is not in this order, ICCIAR may be read before it completely reflects processing through the CPU interface. Consequently, wrong interrupt IDs may be read.
Note 2. When the interrupt is one for which clearing of the flag for the interrupt request before interrupt processing is not possible, the flag should be cleared after interrupt processing.

Figure 7.3 Flow of Interrupt Operations

chanhvinh.com

7.7 Data Transfer with Interrupt Request Signals

Interrupt request signals can be used to activate the direct memory access controller and transfer data.
Interrupt sources for which the direct memory access controller is designated as the destination by DMA extension resource selectors 0 to 7 are masked and requests from them are not input to the interrupt controller.
Figure 7.4 shows a block diagram of interrupt control. For details, see section 9, Direct Memory Access Controller.

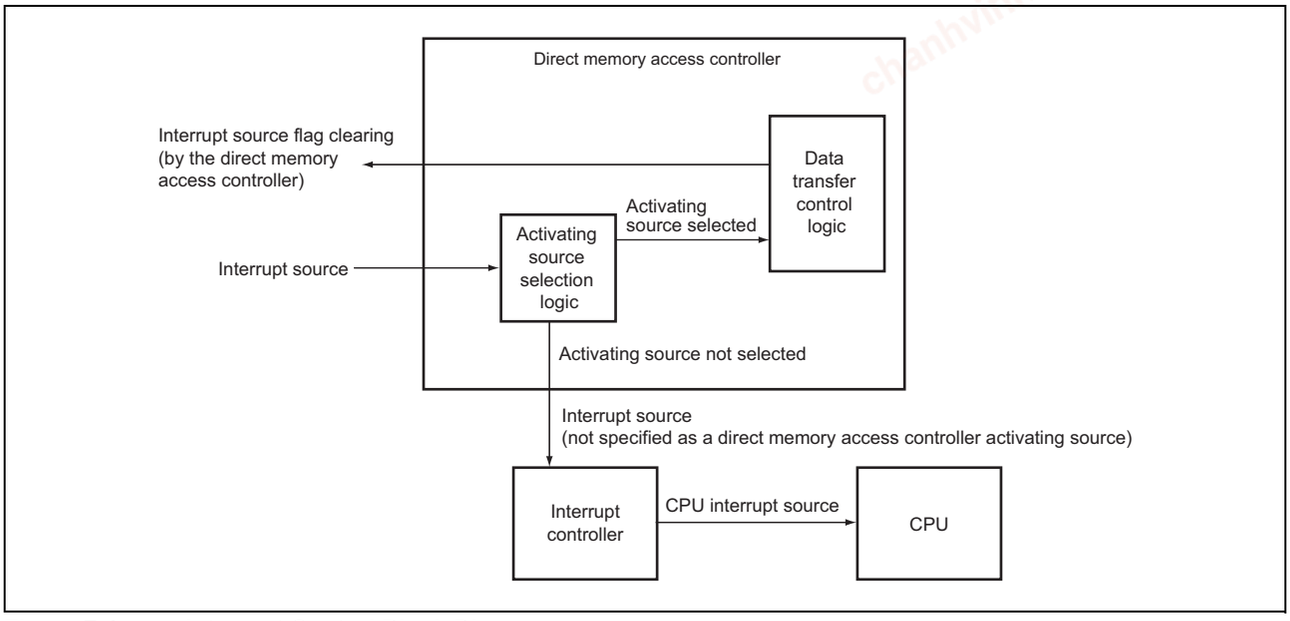


Figure 7.4 Interrupt Control Block Diagram

7.7.1 Handling Interrupt Request Signals as Sources for CPU Interrupt but Not Direct Memory Access Controller Activating

1. Do not select direct memory access controller activating sources.
2. When interrupts occur, interrupt requests are sent to the CPU.
3. The CPU clears the interrupt source and performs the necessary processing in the interrupt exception service routine.

7.7.2 Handling Interrupt Request Signals as Sources for Activating Direct Memory Access Controller but Not CPU Interrupt

1. Select direct memory access controller activating sources.
2. Activating sources are applied to the direct memory access controller when interrupts occur.
3. The direct memory access controller clears the interrupt sources when starting transfer.

chanhvinh.com

7.8 Usage Note

7.8.1 Timing to Clear Interrupt Source

Clear the interrupt source flag to 0 in the interrupt exception handler. It takes some time to clear an interrupt in the CPU after clearing the interrupt source flag to 0. Read the interrupt source flag after clearing it to ensure that the interrupt request that should have been cleared is not received again erroneously. After that, execute the return instruction.

7.8.2 Notes on Selecting IRQ Interrupt Pin Functions

While the corresponding setting in interrupt control register 1 (ICR1) is for interrupt requests to be detected on falling edges or both edges of an IRQn input and the current input on the pin is at the low level, this will be detected as a falling edge and thus an interrupt when the pin function is switched to the IRQ interrupt function.

7.8.3 Notes on Reading Interrupt ID Values from Interrupt Acknowledge Register (ICCIAR)

If an interrupt ID value read from the interrupt acknowledge register (ICCIAR) is 0, the interrupt notice may be wrong. At that time, confirm the interrupt state before proceeding with interrupt processing.
When the interrupt ID is read as 0, the interrupt state can be confirmed by using bit 0 in the active bit register 0 (ICDABR0).

If the interrupt state is inactive, the interrupt notice is wrong and no interrupt processing is required.
Return from interrupt processing after writing the same value as the setting value to the interrupt priority register 0 (ICDIPR0).
If the interrupt state is active, the interrupt notice is correct. Proceed with interrupt processing.
If an interrupt ID value read from the interrupt acknowledge register (ICCIAR) is 1022 or 1023, return from interrupt processing after writing the same value as the setting value to the interrupt priority register 0 (ICDIPR0).

chanhvinh.com

7.8.4 Notes on Using IRQ Pins as Triggers for Release from Standby when Software Standby is in Use

To use an IRQ pin as the trigger for release from standby when software standby is in use, execute the following processing.

(1) When the mode setting for an IRQ pin has been switched from the alternative mode to the port mode

Set the IRQ sense select bits corresponding to the given IRQ pin in interrupt control register 1 (ICR1) to the initial value, 00 (interrupt requests being detected as the low level of the IRQn input).

(2) When the mode setting for an IRQ pin has been switched from the port mode to the alternative mode

After switching the pin mode setting, make the setting of the IRQ sense select bits corresponding to the given IRQ pin in the following sequence. To start with, set the IRQ sense select bits in interrupt control register 1 (ICR1) to the initial value, 00 (interrupt requests being detected as the low level of the IRQn input), and then set them to the desired value, as described below.

- When the IRQ pin is to be used with the IRQ sense select bits set to 01 (interrupt requests being detected on falling edges of the IRQn input)
  Set these bits to 00 (detection as the low level) *1 and then to 01 (detection on falling edges).
- When the IRQ pin is to be used with the IRQ sense select bits set to 10 (interrupt requests being detected on rising edges of the IRQn input)
  Set these bits to 00 (detection as the low level) *1 and then to 10 (detection on rising edges).
- When the IRQ pin is to be used with the IRQ sense select bits set to 11 (interrupt requests being detected on the both edges of the IRQn input)
  Set these bits to 00 (detection as the low level) *1 and then to 11 (detection on the both edges).

Note 1. This setting is for the detection of interrupt requests when the IRQ pin is at the low level. Make the appropriate settings so that the interrupt function does not operate, i.e. the interrupt is ignored, during the period from setting of the IRQ pin mode to setting of the IRQ sense select bits.

chanhvinh.com

8. Bus State Controller

The bus state controller outputs control signals for various types of memory and external devices that are connected to the external address space. The functions of this module enable this LSI to connect directly with SRAM, SDRAM, and other memory storage devices, and external devices.

8.1 Features

1. External address space
   —A maximum of 64 Mbytes for each of areas CS0 to CS5.
   —Can specify the normal space interface, SRAM interface with byte selection, burst ROM (clocked synchronous or asynchronous), MPX-I/O, and SDRAM memory type for each address space.
   —Can select the data bus width (8, 16, or 32 bits) for each of address spaces.
   —Controls insertion of wait cycles for each address space.
   —Controls insertion of wait cycles for each read access and write access.
   —Can set independent idle cycles during the continuous access for five cases: read-write (in same space/different spaces), read-read (in same space/different spaces), the first cycle is a write access.
2. Normal space interface
   —Supports the interface that can directly connect to the SRAM.
3. Burst ROM interface (clocked asynchronous)
   —High-speed access to the ROM that has the page mode function.
4. MPX-I/O interface
   —Can directly connect to a peripheral LSI that needs an address/data multiplexing.
5. SDRAM interface
   —Can set the SDRAM in up to two areas.
   —Multiplex output for row address/column address.
   —Efficient access by single read/single write.
   —High-speed access in bank-active mode.
   —Supports an auto-refresh and self-refresh.
   —Supports a power-down mode.
   —Issues MRS and EMRS commands.
6. SRAM interface with byte selection
   —Can connect directly to a SRAM with byte selection.
7. Burst ROM interface (clocked synchronous)
   —Can connect directly to a burst ROM of the clocked synchronous type.
8. Refresh function
   —Supports the auto-refresh and self-refresh functions.
   —Specifies the refresh interval using the refresh counter and clock selection.
   —Can execute concentrated refresh by specifying the refresh counts (1, 2, 4, 6, or 8).
9. Usage as interval timer for refresh counter
   —Generates an interrupt request at compare match.
10. Detection of long wait state for access by the signal on the external $\overline{\text{WAIT}}$ pin.
    —A timeout detection condition is specifiable per CS space.
    —Once timeout is detected, the external WAIT function is disabled and a timeout detection interrupt request is issued.

chanhvinh.com

Figure 8.1 shows a block diagram of this module.

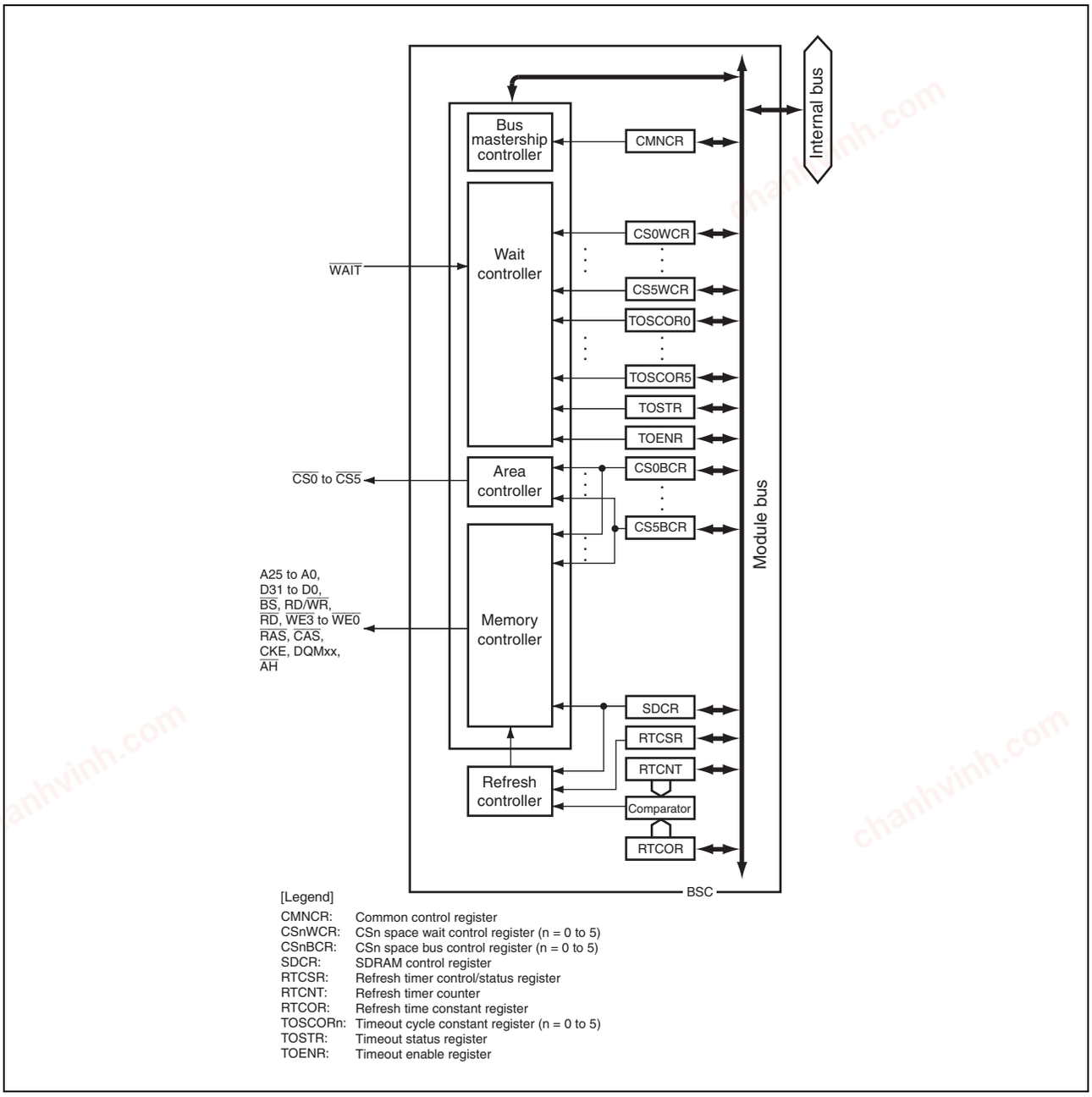


Figure 8.1 Block Diagram of Bus State Controller

chanhvinh.com

8.2 Input/Output Pins

Table 8.1 shows the pin configuration.

Table 8.1 Pin Configuration

Name	I/O	Function
A25 to A0	Output	Address bus
D31 to D0	I/O	Data bus
$\overline{BS}$	Output	Bus cycle start
$\overline{CS0}$ to $\overline{CS5}$	Output	Chip select
RD/$\overline{WR}$	Output	Read/write Connects to $\overline{WE}$ pins when SDRAM or SRAM with byte selection is connected.
$\overline{RD}$	Output	Read pulse signal (read data output enable signal)
$\overline{WE3}$/DQMUU/$\overline{AH}$	Output	Indicates that D31 to D24 are being written to. Connected to the byte select signal when a SRAM with byte selection is connected. Functions as the select signals for D31 to D24 when SDRAM is connected. Functions as the address hold signal when the MPX-I/O is used.
$\overline{WE2}$/DQMUL	Output	Indicates that D23 to D16 are being written to. Connected to the byte select signal when a SRAM with byte selection is connected. Functions as the select signals for D23 to D16 when SDRAM is connected.
$\overline{WE1}$/DQMLU	Output	Indicates that D15 to D8 are being written to. Connected to the byte select signal when a SRAM with byte selection is connected. Functions as the select signals for D15 to D8 when SDRAM is connected.
$\overline{WE0}$/DQMLL	Output	Indicates that D7 to D0 are being written to. Connected to the byte select signal when a SRAM with byte selection is connected. Functions as the select signals for D7 to D0 when SDRAM is connected.
$\overline{RAS}$	Output	Connects to $\overline{RAS}$ pin when SDRAM is connected.
$\overline{CAS}$	Output	Connects to $\overline{CAS}$ pin when SDRAM is connected.
CKE	Output	Connects to CKE pin when SDRAM is connected.
$\overline{WAIT}$	Input	External wait input

chanhvinh.com

8.3 Area Overview

8.3.1 Address Map

In the architecture, this LSI has a 32-bit address space, which is divided into external memory spaces (SPI multi I/O bus space, large-capacity on-chip RAM, hold on-chip RAM, on-chip peripheral modules, and reserved areas) according to the upper bits of the address.

See section 5, LSI Internal Bus (including Secondary Cache) for how to enable or disable caching for the CS0 to CS5 external address spaces.

The kind of memory to be connected and the data bus width are specified in each partial space. The address map for the external address space is listed below.

Table 8.2 Address Map

Internal Address	Space	Memory to be Connected
H'00000000 to H'03FFFFFF	CS0	Normal space, SRAM with byte selection, burst ROM (asynchronous or synchronous)
H'04000000 to H'07FFFFFF	CS1	Normal space, SRAM with byte selection
H'08000000 to H'0BFFFFFF	CS2	Normal space, SRAM with byte selection, SDRAM
H'0C000000 to H'0FFFFFFF	CS3	Normal space, SRAM with byte selection, SDRAM
H'10000000 to H'13FFFFFF	CS4	Normal space, SRAM with byte selection, burst ROM (asynchronous)
H'14000000 to H'17FFFFFF	CS5	Normal space, SRAM with byte selection, MPX-I/O
H'18000000 to H'3FFFFFFF	Other	SPI multi I/O bus space, large-capacity on-chip RAM, hold on-chip RAM, on-chip peripheral modules, reserved area*1
H'40000000 to H'407FFFFF	CS0 mirror	Access from the north main bus is not possible, but access from the south main bus is possible.*2
H'40800000 to H'43FFFFFF	CS0 mirror	Normal space, SRAM with byte selection, burst ROM (asynchronous or synchronous)
H'44000000 to H'47FFFFFF	CS1 mirror	Normal space, SRAM with byte selection
H'48000000 to H'4BFFFFFF	CS2 mirror	Normal space, SRAM with byte selection, SDRAM
H'4C000000 to H'4FFFFFFF	CS3 mirror	Normal space, SRAM with byte selection, SDRAM
H'50000000 to H'53FFFFFF	CS4 mirror	Normal space, SRAM with byte selection, burst ROM (asynchronous)
H'54000000 to H'57FFFFFF	CS5 mirror	Normal space, SRAM with byte selection, MPX-I/O
H'58000000 to H'FFFFFFFF	Other	SPI multi I/O bus space (mirror), large-capacity on-chip RAM (mirror), hold on-chip RAM (mirror), on-chip peripheral modules, reserved area*1

Note 1. For the large-capacity on-chip RAM space and hold on-chip RAM space, access the addresses shown in section 40, On-Chip RAM. For the on-chip peripheral module space, access the addresses shown in section 46, List of Registers. Do not access addresses which are not described in these sections. Otherwise, the correct operation cannot be guaranteed.

Note 2. For details, see section 5, LSI Internal Bus.

chanhvinh.com

8.3.2 Data Bus Width and Related Pin Setting for Each Area Depending on Boot Mode

The initial state of data bus width and settings of the pins related to this module depends on boot mode. For boot mode, refer to section 3, Boot Mode.

In boot mode 0, the state of area 0 is fixed to the state with bus width of 16 bits, because this LSI is started up by the program stored in the ROM connected to area 0. The initial states of areas 1 to 5 are the same as that of area 0, but the bus width can be changed by the program. Immediately after a power-on reset in these modes, some of the address and data-bus signals and the $\overline{CS0}$, $\overline{RD}$, and RD/$\overline{WR}$ signals are automatically selected by default as the functions of the corresponding pins, since these signals are required to read ROM data from area 0. With the exception of these pins, the general purpose pin function is selected by default, and other required pin functions must be specified by the program. Read access to area 0 is only permitted before the pin settings are completed.

In boot modes 1 to 3, the state of areas 0 to 5 can be changed from the initial state by the program, because the LSI is started by the program stored in the SPI serial memory, the NAND flash memory with the SD controller, or the NAND flash memory with the MMC controller. Since pin functions related to this module are not set automatically, they need to be set by the program. Do not access external address spaces before the pin settings are completed.

Table 8.3 shows the initial state by areas 0 to 5 in boot modes 0 and 1 to 3.

The sample access waveforms shown in this section include the pins such as $\overline{BS}$ and $\overline{WEn}$. They are the waveforms when pin functions are assigned to the general I/O ports. For example, when 16-bit bus width is used, setting for pin A1 is needed. When 8-bit bus width is used, setting for pins A1 and A0 is also needed.

For details on pin function settings, see section 41, Ports.

Table 8.3 Initial States by Areas in Boot Modes 0 and 1 to 3

Boot Mode	Item	Area 0	Areas 1 to 5
0	Data bus width	Fixed to 16 bits. Not changeable.	16 bits. Can be changed by program.
	Settings of pins related to this module	Pins A20 to A1, D15 to D0, $\overline{CS0}$, RD/$\overline{WR}$, and $\overline{RD}$ are set automatically. Other pins need to be set by program.	
1 to 3	Data bus width	32 bits. Can be changed by program.	
	Settings of pins related to this module	General I/O function. For external bus access, all the necessary pins need to be set by program.	

Note 1. If operation is to start in boot mode 0 and the connected boot ROM includes address bit A21 or higher-order address bits, the circuit board must include pull-down resistors for the corresponding address lines.

Note 2. The data-bus width may be limited by the type of memory in use. For details, see section 8.4.2, CSn Space Bus Control Register (CSnBCR) (n = 0 to 5).

chanhvinh.com

8.4 Register Descriptions

Table 8.4 shows the register configuration of this module.

Do not access the areas until settings of the connected memory interface are completed.

Table 8.4 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Common control register	CMNCR	R/W	H'00001018	H'3FFFC000	32
CS0 space bus control register	CS0BCR	R/W	H'36DB0C00*1	H'3FFFC004	32
CS1 space bus control register	CS1BCR	R/W	H'36DB0C00*1	H'3FFFC008	32
CS2 space bus control register	CS2BCR	R/W	H'36DB0C00*1	H'3FFFC00C	32
CS3 space bus control register	CS3BCR	R/W	H'36DB0C00*1	H'3FFFC010	32
CS4 space bus control register	CS4BCR	R/W	H'36DB0C00*1	H'3FFFC014	32
CS5 space bus control register	CS5BCR	R/W	H'36DB0C00*1	H'3FFFC018	32
CS0 space wait control register	CS0WCR	R/W	H'00000500	H'3FFFC028	32
CS1 space wait control register	CS1WCR	R/W	H'00000500	H'3FFFC02C	32
CS2 space wait control register	CS2WCR	R/W	H'00000500	H'3FFFC030	32
CS3 space wait control register	CS3WCR	R/W	H'00000500	H'3FFFC034	32
CS4 space wait control register	CS4WCR	R/W	H'00000500	H'3FFFC038	32
CS5 space wait control register	CS5WCR	R/W	H'00000500	H'3FFFC03C	32
SDRAM control register	SDCR	R/W	H'00000000	H'3FFFC04C	32
Refresh timer control/status register	RTCSR	R/W	H'00000000	H'3FFFC050	32
Refresh timer counter	RTCNT	R/W	H'00000000	H'3FFFC054	32
Refresh time constant register	RTCOR	R/W	H'00000000	H'3FFFC058	32
Timeout cycle constant register 0	TOSCOR0	R/W	H'00000000	H'3FFFC060	32
Timeout cycle constant register 1	TOSCOR1	R/W	H'00000000	H'3FFFC064	32
Timeout cycle constant register 2	TOSCOR2	R/W	H'00000000	H'3FFFC068	32
Timeout cycle constant register 3	TOSCOR3	R/W	H'00000000	H'3FFFC06C	32
Timeout cycle constant register 4	TOSCOR4	R/W	H'00000000	H'3FFFC070	32
Timeout cycle constant register 5	TOSCOR5	R/W	H'00000000	H'3FFFC074	32
Timeout status register	TOSTR	R/W	H'00000000	H'3FFFC080	32
Timeout enable register	TOENR	R/W	H'00000000	H'3FFFC084	32

Note 1. H'36DB0C00 in boot mode 0; H'36DB0E00 in boot modes 1 to 3

chanhvinh.com

8.4.1 Common Control Register (CMNCR)

CMNCR is a 32-bit register that controls the common items for each area.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	TL0	-	-	-	AL0	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	DPRTY[1:0]		-	-	-	-	-	-	-	HIZ MEM	HIZ CNT*
Initial value:	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28	TL0	0	R/W	Transfer End Level Specifies the TEND0 signal output is high active or low active. 0: Low-active output from TEND0 1: High-active output from TEND0
27 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	AL0	0	R/W	Specifies the DACK0 (acknowledge) signal output is high active or low active. 0: Low-active output from DACK0 1: High-active output from DACK0
23 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10, 9	DPRTY[1:0]	00	R/W	DMA Burst Transfer Priority Specify the priority for a refresh request during DMA burst transfer. 0*: Accepts a refresh request during DMA burst transfer. 10: Does not accept a refresh request during DMA burst transfer. 11: Reserved (setting prohibited)
8 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4, 3	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	HIZMEM	0	R/W	High-Z Memory Control Specifies the pin state in software standby mode or deep standby mode for A25 to A0, $\overline{BS}$, $\overline{CSn}$, RD/$\overline{WR}$, $\overline{WEn}$/DQMxx/$\overline{AH}$, and $\overline{RD}$. 0: High impedance in software standby mode or deep standby mode. 1: Driven in software standby mode or deep standby mode
0	HIZCNT*	0	R/W	High-Z Control Specifies the state in software standby mode or deep standby mode for CKE, $\overline{RAS}$, and $\overline{CAS}$. 0: High impedance in software standby mode or deep standby mode for CKE, $\overline{RAS}$, and $\overline{CAS}$. 1: Driven in software standby mode or deep standby mode for CKE, $\overline{RAS}$, and $\overline{CAS}$.

Note: * For High-Z control of CKIO, see section 6, Clock Pulse Generator.

chanhvinh.com

8.4.2 CSn Space Bus Control Register (CSnBCR) (n = 0 to 5)

CSnBCR is a 32-bit readable/writable register that specifies the memory connected to each space, the number of idle cycles between bus cycles, and the bus width.

Do not access external memory for the corresponding area until CSnBCR initial setting and pin setting are completed. Idle cycles may be inserted even when they are not specified. For details, see section 8.5.10, Wait between Access Cycles.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	IWW[2:0]			IWRWD[2:0]			IWRWS[2:0]			IWRRD[2:0]			IWRRS[2:0]		
Initial value:	0	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	TYPE[2:0]			-	BSZ[1:0]		-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	1	1*	0*	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R	R	R	R	R	R	R	R	R

Note: * B'10 in boot mode 0; B'11 in boot modes 1 to 3.

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 28	IWW[2:0]	011	R/W	Idle Cycles between Write-Read Cycles and Write-Write Cycles These bits specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target access cycles are the write-read cycle and write-write cycle. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
27 to 25	IWRWD[2:0]	011	R/W	Idle Cycles for Another Space Read-Write Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target access cycle is a read-write one in which continuous access cycles switch between different spaces. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
24 to 22	IWRWS[2:0]	011	R/W	Idle Cycles for Read-Write in the Same Space Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target cycle is a read-write cycle of which continuous access cycles are for the same space. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
21 to 19	IWRRD[2:0]	011	R/W	Idle Cycles for Read-Read in Another Space Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target cycle is a read-read cycle of which continuous access cycles switch between different space. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
18 to 16	IWRRS[2:0]	011	R/W	Idle Cycles for Read-Read in the Same Space Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target cycle is a read-read cycle of which continuous access cycles are for the same space. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	TYPE[2:0]	000	R/W	Specify the type of memory connected to a space. 000: Normal space 001: Burst ROM (clock asynchronous) 010: MPX-I/O 011: SRAM with byte selection 100: SDRAM 101: Reserved (setting prohibited) 110: Reserved (setting prohibited) 111: Burst ROM (clock synchronous) For details for memory type in each area, see Table 8.2. Note: When connecting the burst ROM to the CS0 space in boot mode 0, change the CS0WCR register to the settings by the burst ROM CS0WCR uses and then set TYPE[2:0] to the burst ROM setting. In boot modes 1 to 3, memory access should be performed after setting CS0BCR and CS0WCR.
11	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
10, 9	BSZ[1:0]	10*	R/W	Data Bus Width Specification Specify the data bus widths of spaces. 00: Reserved (setting prohibited) 01: 8-bit size 10: 16-bit size 11: 32-bit size For MPX-I/O, selects bus width by address Notes: 1. If area 5 is specified as MPX-I/O, the bus width can be specified as 8 bits or 16 bits by the address according to the SZSEL bit in CS5WCR by specifying the BSZ[1:0] bits to 11. The fixed bus width can be specified as 8 bits or 16 bits 2. In boot modes 0, the BSZ[1:0] bits settings in CS0BCR are ignored. 3. If area 2 or area 3 is specified as SDRAM space, the bus width can be specified as either 16 bits or 32 bits. 4. If area 0 is specified as clocked synchronous burst ROM space, the bus width can be specified as either 16 bits or 32 bits.
8 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * B'10 in boot mode 0; B'11 in boot modes 1 to 3.

chanhvinh.com

8.4.3 CSn Space Wait Control Register (CSnWCR) (n = 0 to 5)

CSnWCR specifies various wait cycles for memory access. The bit configuration of this register varies as shown below according to the memory type (TYPE2 to TYPE0) specified by the CSn space bus control register (CSnBCR). Specify CSnWCR before accessing the target area. Specify CSnBCR first, then specify CSnWCR.

(1) Normal Space, SRAM with Byte Selection, and MPX-I/O

- CS0WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-*	BAS	-	-	-*	-*
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SW[1:0]		WR[3:0]				WM	-	-	-	-	HW[1:0]	
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21	—*	0	R/W	Reserved Set this bit to 0 when the interfaces for normal space or for SRAM with byte selection are used.
20	BAS	0	R/W	SRAM with Byte Selection Byte Access Select Specifies the WEn and RD/WR signal timing when the SRAM interface with byte selection is used. 0: Asserts the WEn signal at the read/write timing and asserts the RD/WR signal during the write access cycle. 1: Asserts the WEn signal during the read/write access cycle and asserts the RD/WR signal at the write timing.
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	—*	All 0	R/W	Reserved Set these bits to 0 when the interfaces for normal space or for SRAM with byte selection are used.
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address, CS0 Assertion to RD, WEn Assertion Specify the number of delay cycles from address and CS0 assertion to RD and WEn assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
10 to 7	WR[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of cycles that are necessary for read/write access. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	HW[1:0]	00	R/W	Delay Cycles from $\overline{RD}$, $\overline{WEn}$ Negation to Address, $\overline{CS0}$ Negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ negation to address and $\overline{CS0}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

Note: * In boot mode 0, to connect the burst ROM to the CS0 space and switch to burst ROM interface after activation, set the TYPE[2:0] bits in CS0BCR after setting the burst number by the bits 20 and 21 and the burst wait cycle number by the bits 16 and 17. Do not write 1 to the reserved bits other than above bits.

chanhvinh.com

- CS1WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	BAS	-	WW[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SW[1:0]		WR[3:0]				WM	-	-	-	-	HW[1:0]	
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BAS	0	R/W	SRAM with Byte Selection Byte Access Select Specifies the WEn and RD/WR signal timing when the SRAM interface with byte selection is used. 0: Asserts the WEn signal at the read/write timing and asserts the RD/WR signal during the write access cycle. 1: Asserts the WEn signal during the read/write access cycle and asserts the RD/WR signal at the write timing.
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18 to 16	WW[2:0]	000	R/W	Number of Write Access Wait Cycles Specify the number of cycles that are necessary for write access. 000: The same cycles as WR[3:0] setting (number of read access wait cycles) 001: No cycle 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address, CSn Assertion to RD, WEn Assertion Specify the number of delay cycles from address and CSn assertion to RD and WEn assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
10 to 7	WR[3:0]	1010	R/W	Number of Read Access Wait Cycles Specify the number of cycles that are necessary for read access. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	HW[1:0]	00	R/W	Delay Cycles from $\overline{RD}$, $\overline{WEn}$ Negation to Address, $\overline{CSn}$ Negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ negation to address and $\overline{CSn}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

chanhvinh.com

- CS2WCR, CS3WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	BAS	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	WR[3:0]				WM	-	-	-	-	-	-
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BAS	0	R/W	SRAM with Byte Selection Byte Access Select Specifies the $\overline{\text{WEn}}$ and RD/$\overline{\text{WR}}$ signal timing when the SRAM interface with byte selection is used. 0: Asserts the $\overline{\text{WEn}}$ signal at the read timing and asserts the RD/$\overline{\text{WR}}$ signal during the write access cycle. 1: Asserts the $\overline{\text{WEn}}$ signal during the read access cycle and asserts the RD/$\overline{\text{WR}}$ signal at the write timing.
19 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 7	WR[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of cycles that are necessary for read/write access. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

- CS4WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	BAS	-	WW[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SW[1:0]		WR[3:0]				WM	-	-	-	-	HW[1:0]	
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BAS	0	R/W	SRAM with Byte Selection Byte Access Select Specifies the $\overline{WEn}$ and RD/$\overline{WR}$ signal timing when the SRAM interface with byte selection is used. 0: Asserts the $\overline{WEn}$ signal at the read timing and asserts the RD/$\overline{WR}$ signal during the write access cycle. 1: Asserts the $\overline{WEn}$ signal during the read access cycle and asserts the RD/$\overline{WR}$ signal at the write timing.
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18 to 16	WW[2:0]	000	R/W	Number of Write Access Wait Cycles Specify the number of cycles that are necessary for write access. 000: The same cycles as WR[3:0] setting (number of read access wait cycles) 001: No cycle 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address, $\overline{CS4}$ Assertion to $\overline{RD}$, $\overline{WE}$ Assertion Specify the number of delay cycles from address and $\overline{CS4}$ assertion to $\overline{RD}$ and $\overline{WE}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
10 to 7	WR[3:0]	1010	R/W	Number of Read Access Wait Cycles Specify the number of cycles that are necessary for read access. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	HW[1:0]	00	R/W	Delay Cycles from $\overline{RD}$, $\overline{WEn}$ Negation to Address, $\overline{CS4}$ Negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ negation to address and $\overline{CS4}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

chanhvinh.com

- CS5WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	SZSEL	MPXW/ BAS	-	WW[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SW[1:0]		WR[3:0]				WM	-	-	-	-	HW[1:0]	
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21	SZSEL	0	R/W	MPX-I/O Interface Bus Width Specification Specifies an address to select the bus width when the BSZ[1:0] of CS5BCR are specified as 11. This bit is valid only when area 5 is specified as MPX-I/O. 0: Selects the bus width by address A14 1: Selects the bus width by address A21 The relationship between the SZSEL bit and bus width selected by A14 or A21 are summarized below.
20	MPXW	0	R/W	MPX-I/O Interface Address Wait This bit setting is valid only when area 5 is specified as MPX-I/O. Specifies the address cycle insertion wait for MPX-I/O interface. 0: Inserts no wait cycle 1: Inserts 1 wait cycle
20	BAS	0	R/W	SRAM with Byte Selection Byte Access Select This bit setting is valid only when area 5 is specified as SRAM with byte selection. Specifies the $\overline{WEn}$ and RD/$\overline{WR}$ signal timing when the SRAM interface with byte selection is used. 0: Asserts the $\overline{WEn}$ signal at the read timing and asserts the RD/$\overline{WR}$ signal during the write access cycle. 1: Asserts the $\overline{WEn}$ signal during the read access cycle and asserts the RD/$\overline{WR}$ signal at the write timing.
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18 to 16	WW[2:0]	000	R/W	Number of Write Access Wait Cycles Specify the number of cycles that are necessary for write access. 000: The same cycles as WR[3:0] setting (number of read access wait cycles) 001: No cycle 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

SZSEL	A14	A21	Bus Width
0	0	Not affected	8 bits
0	1	Not affected	16 bits
1	Not affected	0	8 bits
1	Not affected	1	16 bits

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address, $\overline{CS5}$ Assertion to $\overline{RD}$, $\overline{WE}$ Assertion Specify the number of delay cycles from address and $\overline{CS5}$ assertion to $\overline{RD}$ and $\overline{WEn}$ assertion when area 5 is specified as normal space or SRAM with byte selection. Specify the number of delay cycles from the end of address cycle (Ta3) to $\overline{RD}$ and $\overline{WEn}$ assertion when area 5 is specified as MPx-I/O. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
10 to 7	WR[3:0]	1010	R/W	Number of Read Access Wait Cycles Specify the number of cycles that are necessary for read access. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	HW[1:0]	00	R/W	Delay Cycles from $\overline{RD}$, $\overline{WEn}$ Negation to Address, $\overline{CS5}$ Negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ negation to address and $\overline{CS5}$ negation when area 5 is specified as normal space or SRAM with byte selection. Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ negation to $\overline{CS5}$ negation when area 5 is specified as MPx-I/O. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

chanhvinh.com

(2) Burst ROM (Clocked Asynchronous)

- CS0WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	BST[1:0]		-	-	BW[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	W[3:0]				WM	-	-	-	-	-	-
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21, 20	BST[1:0]	00	R/W	Burst Count Specification Specify the burst count for 16-byte or more access. These bits must not be set to B'11, because B'11 setting is reserved.
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	BW[1:0]	00	R/W	Number of Burst Wait Cycles Specify the number of wait cycles to be inserted between the second or subsequent access cycles in burst access. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 7	W[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of wait cycles to be inserted in the first access cycle. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)

Bus Width	BST[1:0]	Burst count (16-byte access)
8 bits	00	16 burst × one time
	01	4 burst × four times
16 bits	00	8 burst × one time
	01	2 burst × four times
	10	4-4 or 2-4-2 burst
32 bits	xx	4 burst × one time

Note: • For details, see Table 8.17, Relationship between Bus Width, Access Size, and Number of Bursts.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

- CS4WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	BST[1:0]		-	-	BW[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SW[1:0]		W[3:0]				WM	-	-	-	-	HW[1:0]	
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21, 20	BST[1:0]	00	R/W	Burst Count Specification Specify the burst count for 16-byte or more access. These bits must not be set to B'11, because B'11 setting is reserved.

Bus Width	BST[1:0]	Burst count (16-byte access)
8 bits	00	16 burst × one time
	01	4 burst × four times
16 bits	00	8 burst × one time
	01	2 burst × four times
	10	4-4 or 2-4-2 burst
32 bits	xx	4 burst × one time

Note: • For details, see Table 8.17, Relationship between Bus Width, Access Size, and Number of Bursts.

19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	BW[1:0]	00	R/W	Number of Burst Wait Cycles Specify the number of wait cycles to be inserted between the second or subsequent access cycles in burst access. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address, $\overline{CS4}$ Assertion to $\overline{RD}$, $\overline{WEn}$ Assertion Specify the number of delay cycles from address and $\overline{CS4}$ assertion to $\overline{RD}$ and $\overline{WEn}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
10 to 7	W[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of wait cycles to be inserted in the first access cycle. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	HW[1:0]	00	R/W	Delay Cycles from $\overline{RD}$, $\overline{WEn}$ Negation to Address, $\overline{CS4}$ Negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ negation to address and $\overline{CS4}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

chanhvinh.com

(3) SDRAM*

- CS2WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	A2CL[1:0]		-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8, 7	A2CL[1:0]	10	R/W	CAS Latency for Area 2 Specify the CAS latency for area 2. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * If only one area is connected to the SDRAM, specify area 3. In this case, specify area 2 as normal space or SRAM with byte selection.

chanhvinh.com

- CS3WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	WTRP[1:0]*		-	WTRCD[1:0]*		-	A3CL[1:0]		-	-	TRWL[1:0]*		-	WTRC[1:0]*	
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R	R/W	R/W	R	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W

Note: * If both areas 2 and 3 are specified as SDRAM, WTRP[1:0], WTRCD[1:0], TRWL[1:0], and WTRC[1:0] bit settings are used in both areas in common.

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
14, 13	WTRP[1:0]*	00	R/W	Number of Auto-Precharge Completion Wait Cycles Specify the number of minimum precharge completion wait cycles as shown below. • From the start of auto-precharge and issuing of ACTV command for the same bank • From issuing of the PRE/PALL command to issuing of the ACTV command for the same bank • Till entering the power-down mode or deep power-down mode • From the issuing of PALL command to issuing REF command in auto refresh mode • From the issuing of PALL command to issuing SELF command in self refresh mode The setting for areas 2 and 3 is common. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
12	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
11, 10	WTRCD [1:0]*	01	R/W	Number of Wait Cycles between ACTV Command and READ(A)/WRIT(A) Command Specify the minimum number of wait cycles from issuing the ACTV command to issuing the READ(A)/WRIT(A) command. The setting for areas 2 and 3 is common. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8, 7	A3CL[1:0]	10	R/W	CAS Latency for Area 3 Specify the CAS latency for area 3. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
4, 3	TRWL[1:0]*	00	R/W	Number of Auto-Precharge Startup Wait Cycles Specify the number of minimum auto-precharge startup wait cycles as shown below. • Cycle number from the issuance of the WRITA command by this LSI until the completion of auto-precharge in the SDRAM. Equivalent to the cycle number from the issuance of the WRITA command until the issuance of the ACTV command. Confirm that how many cycles are required between the WRITA command receive in the SDRAM and the auto-precharge activation, referring to each SDRAM data sheet. And set the cycle number so as not to exceed the cycle number specified by this bit. • Cycle number from the issuance of the WRIT command until the issuance of the PRE command. This is the case when accessing another low address in the same bank in bank active mode. The setting for areas 2 and 3 is common. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1, 0	WTRC[1:0]*	00	R/W	Number of Idle Cycles from REF Command/Self-Refresh Release to ACTV/REF/MRS Command Specify the number of minimum idle cycles in the periods shown below. • From the issuance of the REF command until the issuance of the ACTV/REF/MRS command • From releasing self-refresh until the issuance of the ACTV/REF/MRS command. The setting for areas 2 and 3 is common. 00: 2 cycles 01: 3 cycles 10: 5 cycles 11: 8 cycles

Note: * If both areas 2 and 3 are specified as SDRAM, WTRP[1:0], WTRCD[1:0], TRWL[1:0], and WTRC[1:0] bit settings are used in both areas in common.

If only one area is connected to the SDRAM, specify area 3. In this case, specify area 2 as normal space or SRAM with byte selection.

chanhvinh.com

(4) Burst ROM (Clocked Synchronous)

- CS0WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	BW[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	W[3:0]				WM	-	-	-	-	-	-
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	BW[1:0]	00	R/W	Number of Burst Wait Cycles Specify the number of wait cycles to be inserted between the second or subsequent access cycles in burst access. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 7	W[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of wait cycles to be inserted in the first access cycle. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

8.4.4 SDRAM Control Register (SDCR)

SDCR specifies the method to refresh and access SDRAM, and the types of SDRAMs to be connected.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	A2ROW[1:0]		-	A2COL[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	DEEP	-	RFSH	RMODE	PDOWN	BACTV	-	-	-	A3ROW[1:0]		-	A3COL[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20, 19	A2ROW[1:0]	00	R/W	Number of Bits of Row Address for Area 2 Specify the number of bits of row address for area 2. 00: 11 bits 01: 12 bits 10: 13 bits 11: Reserved (setting prohibited)
18	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
17, 16	A2COL[1:0]	00	R/W	Number of Bits of Column Address for Area 2 Specify the number of bits of column address for area 2. 00: 8 bits 01: 9 bits 10: 10 bits 11: Reserved (setting prohibited)
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	DEEP	0	R/W	Deep Power-Down Mode This bit is valid for low-power SDRAM. If the RFSH or RMODE bit is set to 1 while this bit is set to 1, the deep power-down entry command is issued and the low-power SDRAM enters the deep power-down mode. 0: Self-refresh mode 1: Deep power-down mode
12	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
11	RFSH	0	R/W	Refresh Control Specifies whether or not the refresh operation of the SDRAM is performed. 0: No refresh 1: Refresh
10	RMODE	0	R/W	Refresh Mode Specifies whether to perform auto-refresh or self-refresh when the RFSH bit is 1. When the RFSH bit is 1 and this bit is 1, self-refresh starts immediately. When the RFSH bit is 1 and this bit is 0, auto-refresh starts according to the contents that are set in registers RTCSR, RTCNT, and RTCOR. 0: Auto-refresh is performed 1: Self-refresh is performed

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
9	PDOWN	0	R/W	Power-Down Mode Specifies whether the SDRAM will enter the power-down mode after the access to the SDRAM. With this bit being set to 1, after the SDRAM is accessed, the CKE signal is driven low and the SDRAM enters the power-down mode. 0: The SDRAM does not enter the power-down mode after being accessed. 1: The SDRAM enters the power-down mode after being accessed.
8	BACTV	0	R/W	Bank Active Mode Specifies to access whether in auto-precharge mode (using READA and WRITA commands) or in bank active mode (using READ and WRIT commands). 0: Auto-precharge mode (using READA and WRITA commands) 1: Bank active mode (using READ and WRIT commands) Note: Bank active mode can be set only for area 3. When both areas 2 and 3 are set to SDRAM, specify the auto-precharge mode.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4, 3	A3ROW[1:0]	00	R/W	Number of Bits of Row Address for Area 3 Specify the number of bits of the row address for area 3. 00: 11 bits 01: 12 bits 10: 13 bits 11: Reserved (setting prohibited)
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1, 0	A3COL[1:0]	00	R/W	Number of Bits of Column Address for Area 3 Specify the number of bits of the column address for area 3. 00: 8 bits 01: 9 bits 10: 10 bits 11: Reserved (setting prohibited)

chanhvinh.com

8.4.5 Refresh Timer Control/Status Register (RTCSR)

RTCSR specifies various items about refresh for SDRAM.

When RTCSR is written, the upper 16 bits of the write data must be H'A55A to cancel write protection.

The phase of the clock for incrementing the count in the refresh timer counter (RTCNT) is adjusted only by a power-on reset. Note that there is an error in the time until the compare match flag is set for the first time after the timer is started with the CKS[2:0] bits being set to a value other than B'000.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	CMF	CMIE	CKS[2:0]			RRC[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0.
7	CMF	0	R/W	Compare Match Flag Indicates that a compare match occurs between the refresh timer counter (RTCNT) and refresh time constant register (RTCOR). This bit is set or cleared in the following conditions. 0: Clearing condition: When 0 is written in CMF after reading out RTCSR during CMF = 1. 1: Setting condition: When the condition RTCNT = RTCOR is satisfied.
6	CMIE	0	R/W	Compare Match Interrupt Enable Enables or disables CMF interrupt requests when the CMF bit in RTCSR is set to 1. 0: Disables CMF interrupt requests. 1: Enables CMF interrupt requests.
5 to 3	CKS[2:0]	000	R/W	Clock Select Select the clock input to count-up the refresh timer counter (RTCNT). 000: Stop the counting-up 001: CKIOφ/4 010: CKIOφ/16 011: CKIOφ/64 100: CKIOφ/256 101: CKIOφ/1024 110: CKIOφ/2048 111: CKIOφ/4096
2 to 0	RRC[2:0]	000	R/W	Refresh Count Specify the number of continuous refresh cycles, when the refresh request occurs after the coincidence of the values of the refresh timer counter (RTCNT) and the refresh time constant register (RTCOR). These bits can make the period of occurrence of refresh long. 000: 1 time 001: 2 times 010: 4 times 011: 6 times 100: 8 times 101: Reserved (setting prohibited) 110: Reserved (setting prohibited) 111: Reserved (setting prohibited)

chanhvinh.com

8.4.6 Refresh Timer Counter (RTCNT)

RTCNT is an 8-bit counter that increments using the clock selected by bits CKS[2:0] in RTCSR. When RTCNT matches RTCOR, RTCNT is cleared to 0. The value in RTCNT returns to 0 after counting up to 255. When the RTCNT is written, the upper 16 bits of the write data must be H'A55A to cancel write protection.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-								
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0.
7 to 0		All 0	R/W	8-Bit Counter

8.4.7 Refresh Time Constant Register (RTCOR)

RTCOR is an 8-bit register. When RTCOR matches RTCNT, the CMF bit in RTCSR is set to 1 and RTCNT is cleared to 0.

When the RFSH bit in SDCR is 1, a memory refresh request is issued by this matching signal. This request is maintained until the refresh operation is performed. If the request is not processed when the next matching occurs, the previous request is ignored.

When the CMIE bit in RTCSR is set to 1, an interrupt request is issued by this matching signal. The request continues to be output until the CMF bit in RTCSR is cleared. Clearing the CMF bit only affects the interrupt request and does not clear the refresh request. Therefore, a combination of refresh request and interval timer interrupt can be specified so that the number of refresh requests are counted by using timer interrupts while refresh is performed periodically.

When RTCOR is written, the upper 16 bits of the write data must be H'A55A to cancel write protection.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-								
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0.
7 to 0		All 0	R/W	8-Bit Counter

chanhvinh.com

8.4.8 Timeout Cycle Constant Register (TOSCORn) (n = 0 to 5)

TOSCORn is a 16-bit register the value of which is effective when the WM bit in the CSn space wait control register (CSnWCR) is 0 and the corresponding bit in the timeout enable register (TOENR) is 1. When the number of cycles of waiting due to the signal on the eternal wait input pin matches the setting of TOSCORn, external wait input is disabled to end the cycle of access, the timeout status flag for the corresponding space in the timeout status register (TOSTR) is set, and a timeout detection interrupt request is generated. The timeout detection interrupt request is retained until the corresponding bit in TOENR is set to 0 or 0 is written to the timeout status flag for the corresponding space. Note that timeout detection is enabled even while the timeout status flag for the corresponding space in TOSTR is 1, and external wait input is disabled in response to a further timeout.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0.
15 to 0		All 0	R/W	16-Bit Register H'0000: 65536 cycles H'0001: 1 cycle : H'FFFF: 65535 cycles

chanhvinh.com

8.4.9 Timeout Status Register (TOSTR)

TOSTOR is an 8-bit register that holds the timeout status flags for the CS spaces. When the WM bit in the CSn space wait control register (CSnWCR) is 0 and the corresponding bit in the timeout enable register (TOENR) is 1 and the number of cycles of waiting in response to the signal on the eternal wait input matches the setting of TOSCORn, the timeout status flag for the corresponding space is set and a timeout detection interrupt request is generated. The only writable value for the timeout status flags is 0, which clears the flag. Writing 1 to a flag is ignored.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	CS5T OSTF	CS4T OSTF	CS3T OSTF	CS2T OSTF	CS1T OSTF	CS0T OSTF
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	—	All 0	R	Reserved These bits are always read as 0.
5	CS5TOSTF	0	R/W	CS5 Space Timeout Status Flag Status flag that indicates that the number of cycles of waiting due to the input on the external wait pin during access to the CS5 space has matched the setting of the CS5 space timeout cycle constant register (TOSCOR5).This bit is set or cleared in the following conditions. 0: Clearing condition When 0 is written in CS5TOSTF. 1: Setting condition When the WM bit in the CS5 space wait control register (CS5WCR) is 0 and the CS5TOEN bit in the timeout enable register (TOENR) is 1, the number of cycles of waiting due to the input on the external wait pin during access to the CS5 space has matched the setting of TOSCOR5.
4	CS4TOSTF	0	R/W	CS4 Space Timeout Status Flag Status flag that indicates that the number of cycles of waiting due to the input on the external wait pin during access to the CS4 space has matched the setting of the CS4 space timeout cycle constant register (TOSCOR4). For the condition to set or clear this bit, refer to the description of CS5TOSTF.
3	CS3TOSTF	0	R/W	CS3 Space Timeout Status Flag Status flag that indicates that the number of cycles of waiting due to the input on the external wait pin during access to the CS3 space has matched the setting of the CS3 space timeout cycle constant register (TOSCOR3). For the condition to set or clear this bit, refer to the description of CS5TOSTF.
2	CS2TOSTF	0	R/W	CS2 Space Timeout Status Flag Status flag that indicates that the number of cycles of waiting due to the input on the external wait pin during access to the CS2 space has matched the setting of the CS2 space timeout cycle constant register (TOSCOR2). For the condition to set or clear this bit, refer to the description of CS5TOSTF.
1	CS1TOSTF	0	R/W	CS1 Space Timeout Status Flag Status flag that indicates that the number of cycles of waiting due to the input on the external wait pin during access to the CS1 space has matched the setting of the CS1 space timeout cycle constant register (TOSCOR1). For the condition to set or clear this bit, refer to the description of CS5TOSTF.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
0	CS0TOSTF	0	R/W	CS0 Space Timeout Status Flag Status flag that indicates that the number of cycles of waiting due to the input on the external wait pin during access to the CS0 space has matched the setting of the CS0 space timeout cycle constant register (TOSCOR0). For the condition to set or clear this bit, refer to the description of CS5TOSTF.

chanhvinh.com

8.4.10 Timeout Enable Register (TOENR)

TOENR is an 8-bit register that specifies enabling or disabling the detection of timeout for waiting in each of the CS spaces.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	CS5 TOEN	CS4 TOEN	CS3 TOEN	CS2 TOEN	CS1 TOEN	CS0 TOEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	—	All 0	R	Reserved These bits are always read as 0.
5	CS5TOEN	0	R/W	CS5 Space Timeout Detection Enable Specifies enabling or disabling the detection of timeout for waiting in the CS5 space. 0: The timeout detection is disabled. 1: The timeout detection is enabled.
4	CS4TOEN	0	R/W	CS4 Space Timeout Detection Enable Specifies enabling or disabling the detection of timeout for waiting in the CS4 space. 0: The timeout detection is disabled. 1: The timeout detection is enabled.
3	CS3TOEN	0	R/W	CS3 Space Timeout Detection Enable Specifies enabling or disabling the detection of timeout for waiting in the CS3 space. 0: The timeout detection is disabled. 1: The timeout detection is enabled.
2	CS2TOEN	0	R/W	CS2 Space Timeout Detection Enable Specifies enabling or disabling the detection of timeout for waiting in the CS2 space. 0: The timeout detection is disabled. 1: The timeout detection is enabled.
1	CS1TOEN	0	R/W	CS1 Space Timeout Detection Enable Specifies enabling or disabling the detection of timeout for waiting in the CS1 space. 0: The timeout detection is disabled. 1: The timeout detection is enabled.
0	CS0TOEN	0	R/W	CS0 Space Timeout Detection Enable Specifies enabling or disabling the detection of timeout for waiting in the CS0 space. 0: The timeout detection is disabled. 1: The timeout detection is enabled.

chanhvinh.com

8.5 Operation

8.5.1 Access Size and Data Alignment

This LSI supports little endian, in which the least significant byte (LSB) is that in the direction of the 0th address.
Data bus width can be selected from 8 bits, 16 bits, and 32 bits for the normal memory and SRAM with byte selection.
Data bus width can be selected from 16 bits and 32 bits for SDRAM. For MPX-I/O, the data bus width is fixed to either 8 or 16 bits, or made selectable as 8 bits or 16 bits by one of the address lines.
Data bus width varies depending on boot mode. For details, refer to section 8.3.2, Data Bus Width and Related Pin Setting for Each Area Depending on Boot Mode.
Data alignment is performed in accordance with the data bus width selected for the device. This also means that four read operations are required to read 32-bit data from a byte-width device. In this LSI, data alignment and conversion of data length is performed automatically between the respective interfaces.
Table 8.5 to Table 8.7 show the relationship between device data width and access unit.

Table 8.5 32-Bit External Device Access and Data Alignment in Little Endian

	Data Bus				Strobe Signals			
Operation	**D31 to D24**	**D23 to D16**	**D15 to D8**	**D7 to D0**	**$\overline{WE3}$, DQMUU**	**$\overline{WE2}$, DQMUL**	**$\overline{WE1}$, DQMLU**	**$\overline{WE0}$, DQMLL**
8-bit access at address 0	—	—	—	Data 7 to 0	—	—	—	Assert
8-bit access at address 1	—	—	Data 7 to 0	—	—	—	Assert	—
8-bit access at address 2	—	Data 7 to 0	—	—	—	Assert	—	—
8-bit access at address 3	Data 7 to 0	—	—	—	Assert	—	—	—
16-bit access at address 0	—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert
16-bit access at address 2	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert	—	—
32-bit access at address 0	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Assert	Assert	Assert	Assert

Table 8.6 16-Bit External Device Access and Data Alignment in Little Endian

	Data Bus				Strobe Signals			
Operation	**D31 to D24**	**D23 to D16**	**D15 to D8**	**D7 to D0**	**$\overline{WE3}$, DQMUU**	**$\overline{WE2}$, DQMUL**	**$\overline{WE1}$, DQMLU**	**$\overline{WE0}$, DQMLL**
8-bit access at address 0	—	—	—	Data 7 to 0	—	—	—	Assert
8-bit access at address 1	—	—	Data 7 to 0	—	—	—	Assert	—
8-bit access at address 2	—	—	—	Data 7 to 0	—	—	—	Assert
8-bit access at address 3	—	—	Data 7 to 0	—	—	—	Assert	—
16-bit access at address 0	—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert
16-bit access at address 2	—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert

chanhvinh.com

Table 8.6 16-Bit External Device Access and Data Alignment in Little Endian

Operation		Data Bus				Strobe Signals			
		D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$, DQMUU	$\overline{WE2}$, DQMUL	$\overline{WE1}$, DQMLU	$\overline{WE0}$, DQMLL
32-bit access at address 0	1st access at address 0	—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert
	2nd access at address 2	—	—	Data 31 to 24	Data 23 to 16	—	—	Assert	Assert

Table 8.7 8-Bit External Device Access and Data Alignment in Little Endian

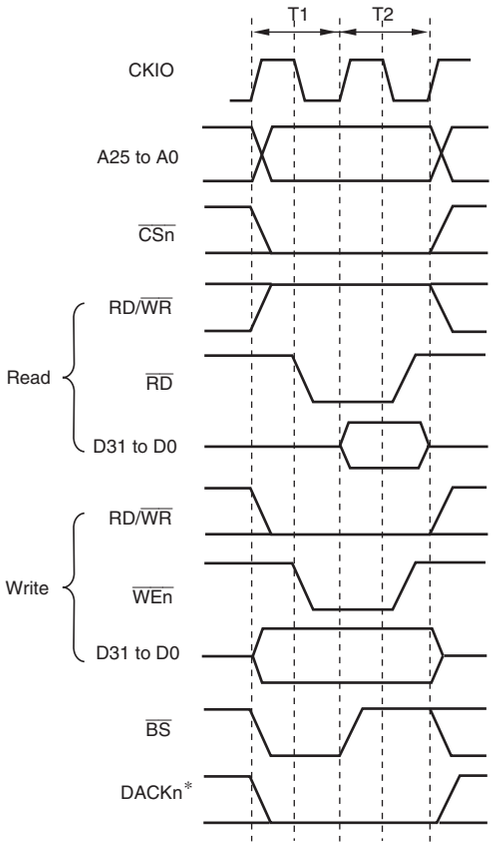
Operation		Data Bus				Strobe Signals			
		D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$, DQMUU	$\overline{WE2}$, DQMUL	$\overline{WE1}$, DQMLU	$\overline{WE0}$, DQMLL
8-bit access at address 0		—	—	—	Data 7 to 0	—	—	—	Assert
8-bit access at address 1		—	—	—	Data 7 to 0	—	—	—	Assert
8-bit access at address 2		—	—	—	Data 7 to 0	—	—	—	Assert
8-bit access at address 3		—	—	—	Data 7 to 0	—	—	—	Assert
16-bit access at address 0	1st access at address 0	—	—	—	Data 7 to 0	—	—	—	Assert
	2nd access at address 1	—	—	—	Data 15 to 8	—	—	—	Assert
16-bit access at address 2	1st access at address 0	—	—	—	Data 7 to 0	—	—	—	Assert
	2nd access at address 1	—	—	—	Data 15 to 8	—	—	—	Assert
32-bit access at address 0	1st access at address 0	—	—	—	Data 7 to 0	—	—	—	Assert
	2nd access at address 1	—	—	—	Data 15 to 8	—	—	—	Assert
	3rd access at address 2	—	—	—	Data 23 to 16	—	—	—	Assert
	4th access at address 3	—	—	—	Data 31 to 24	—	—	—	Assert

chanhvinh.com

8.5.2 Normal Space Interface

(1) Basic Timing

For access to a normal space, this LSI uses strobe signal output in consideration of the fact that mainly static RAM will be directly connected. When using SRAM with a byte-selection pin, see section 8.5.8, SRAM Interface with Byte Selection. Figure 8.2 shows the basic timings of normal space access. A no-wait normal access is completed in two cycles. The $\overline{BS}$ signal is asserted for one cycle to indicate the start of a bus cycle.



Note: * The waveform for DACKn is when active low is specified.

Figure 8.2 Normal Space Basic Access Timing (Access Wait 0)

chanhvinh.com

There is no access size specification when reading. The correct access start address is output in the least significant bit of the address, but since there is no access size specification, 32 bits are always read in case of a 32-bit device. 16 bits are always read in case of a 16-bit device. When writing, only the $\overline{WEn}$ signal for the byte to be written is asserted.
It is necessary to output the data that has been read using $\overline{RD}$ when a buffer is established in the data bus. The RD/$\overline{WR}$ signal is in a read state (high output) when no access has been carried out. Therefore, care must be taken when controlling the external data buffer with this signal, to avoid output collision.
Figure 8.3 and Figure 8.4 show the basic timings in continuous access to normal space. If the WM bit in CSnWCR is cleared to 0, a Tnop cycle is inserted after the CSn space access to evaluate the external wait (Figure 8.3). If the WM bit in CSnWCR is set to 1, external waits are ignored and no Tnop cycle is inserted (Figure 8.4).

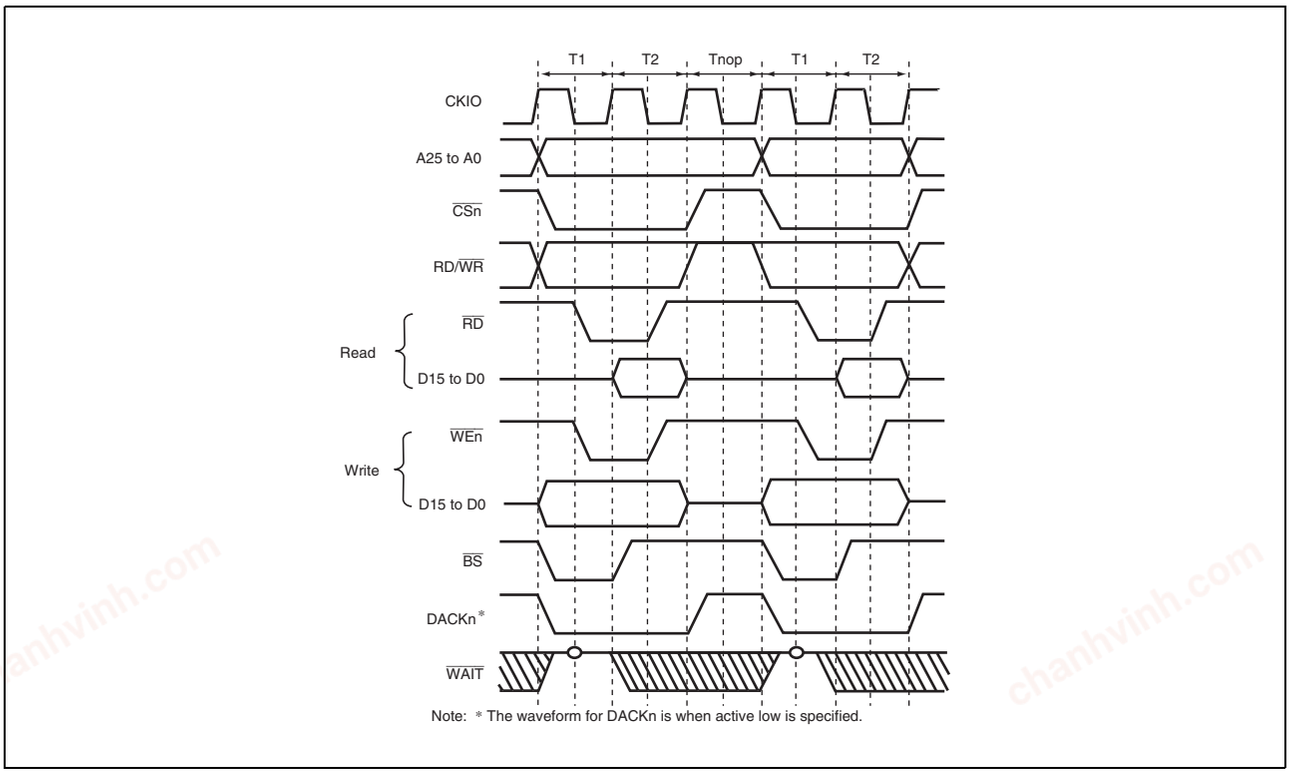


Figure 8.3 Continuous Access to Normal Space (1) Bus Width = 16 Bits, 32-Bit Access, CSnWCR.WM Bit = 0 (Access Wait = 0, Cycle Wait = 0)

chanhvinh.com

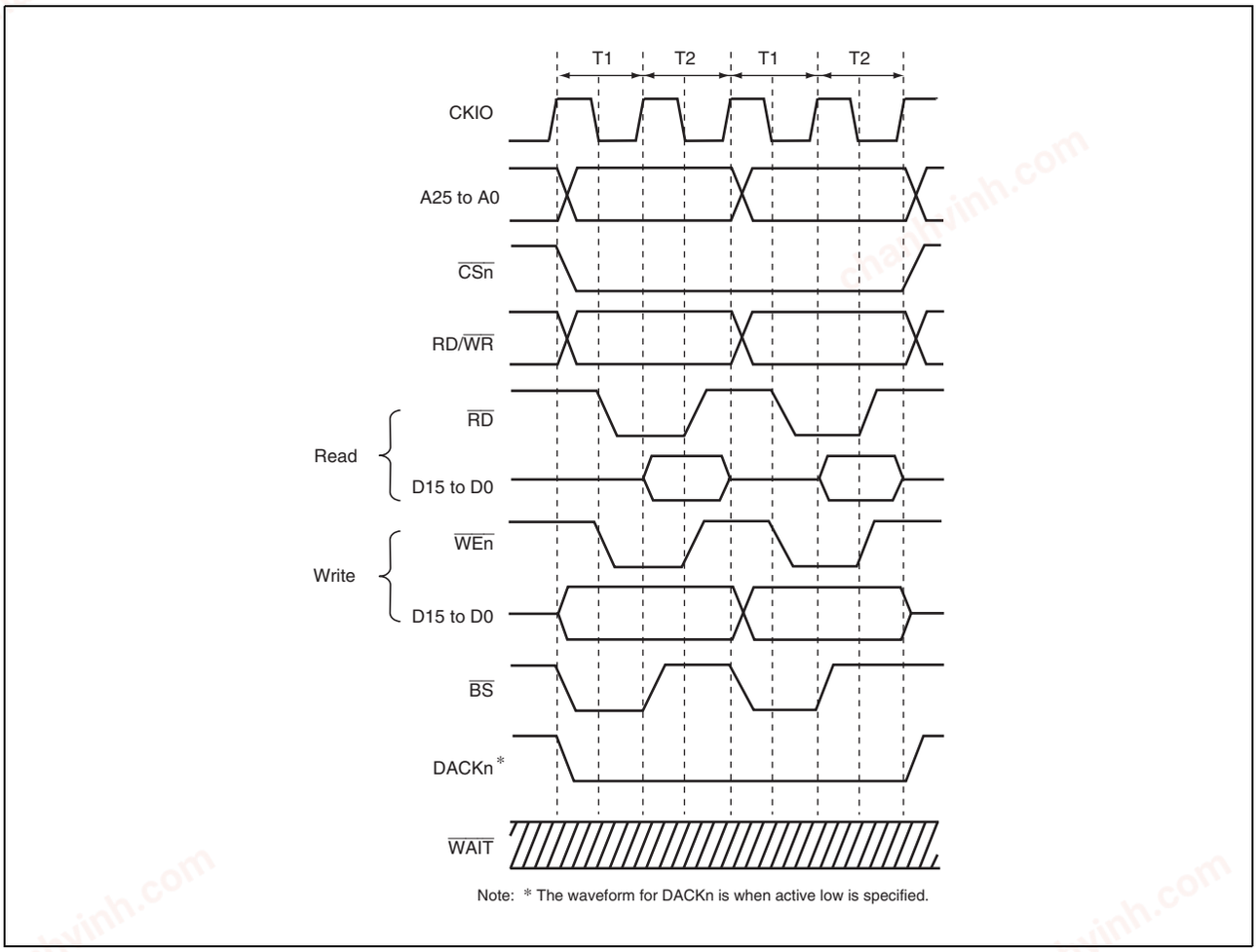


Figure 8.4 Continuous Access to Normal Space (2) Bus Width = 16 Bits, 32-Bit Access, CSnWCR.WM Bit = 1 (Access Wait = 0, Cycle Wait = 0)

chanhvinh.com

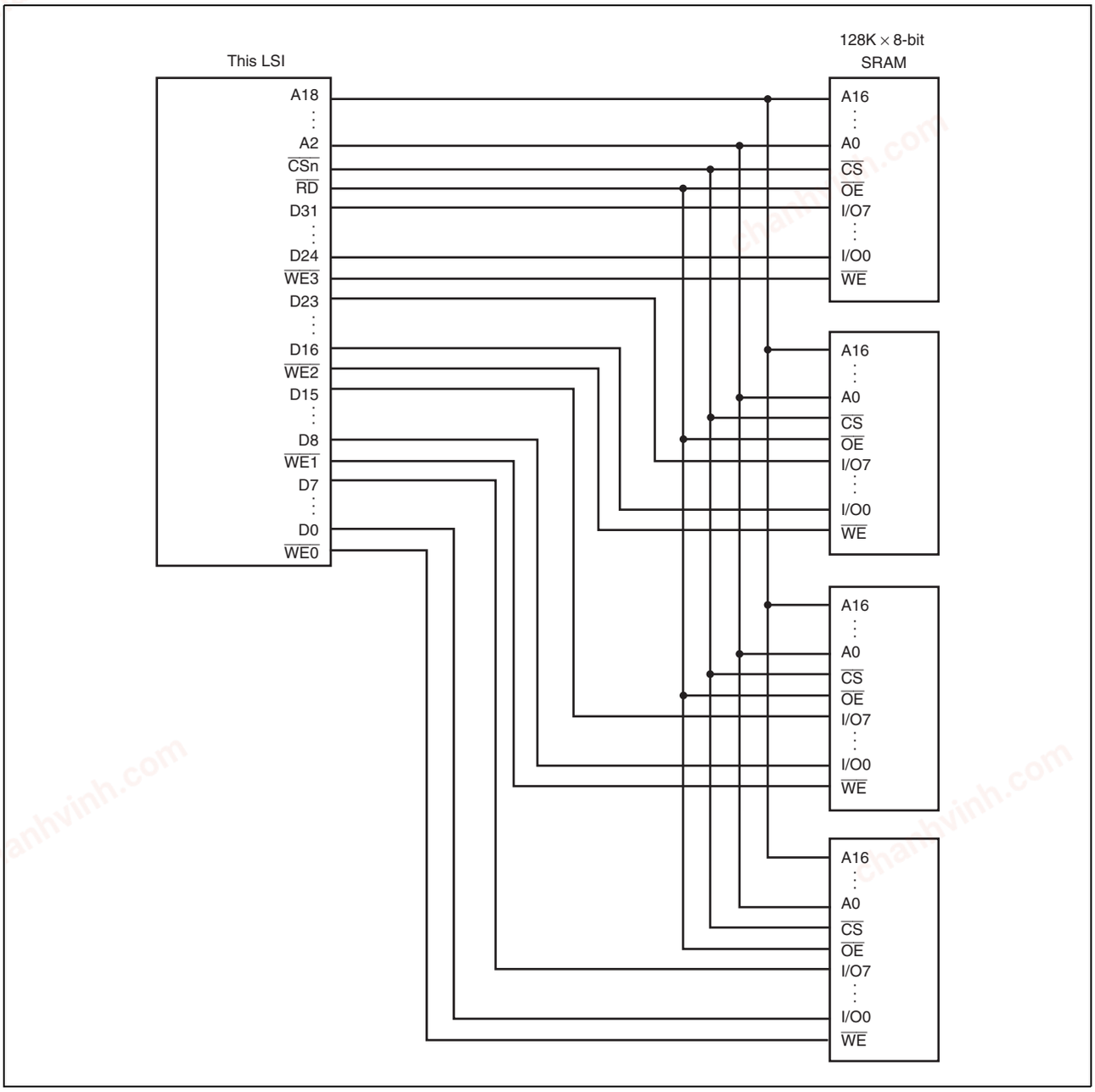


Figure 8.5 Example of 32-Bit Data-Width SRAM Connection

chanhvinh.com

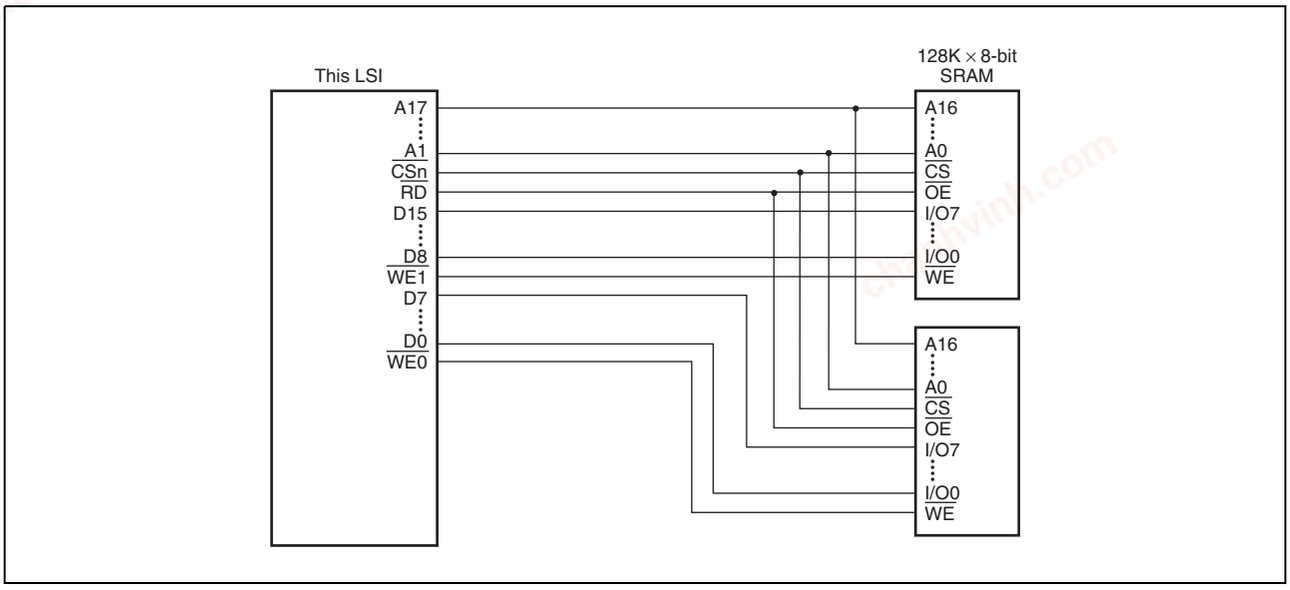


Figure 8.6 Example of 16-Bit Data-Width SRAM Connection

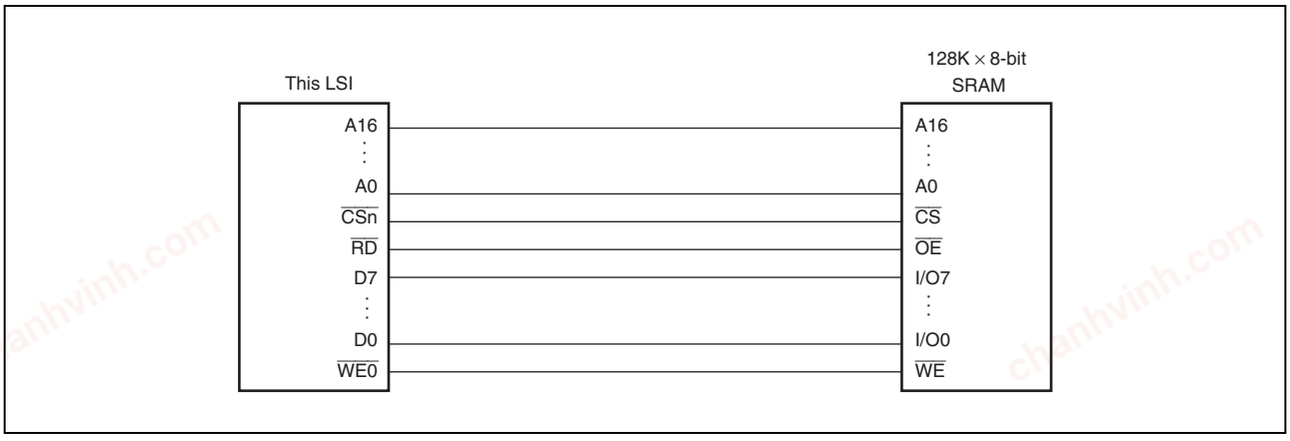


Figure 8.7 Example of 8-Bit Data-Width SRAM Connection

chanhvinh.com

8.5.3 Access Wait Control

Wait cycle insertion on a normal space access can be controlled by the settings of bits WR3 to WR0 in CSnWCR. It is possible for areas 1, 4, and 5 to insert wait cycles independently in read access and in write access. Areas 0, 2, and 3 have common access wait for read cycle and write cycle. The specified number of Tw cycles are inserted as wait cycles in a normal space access shown in Figure 8.8.

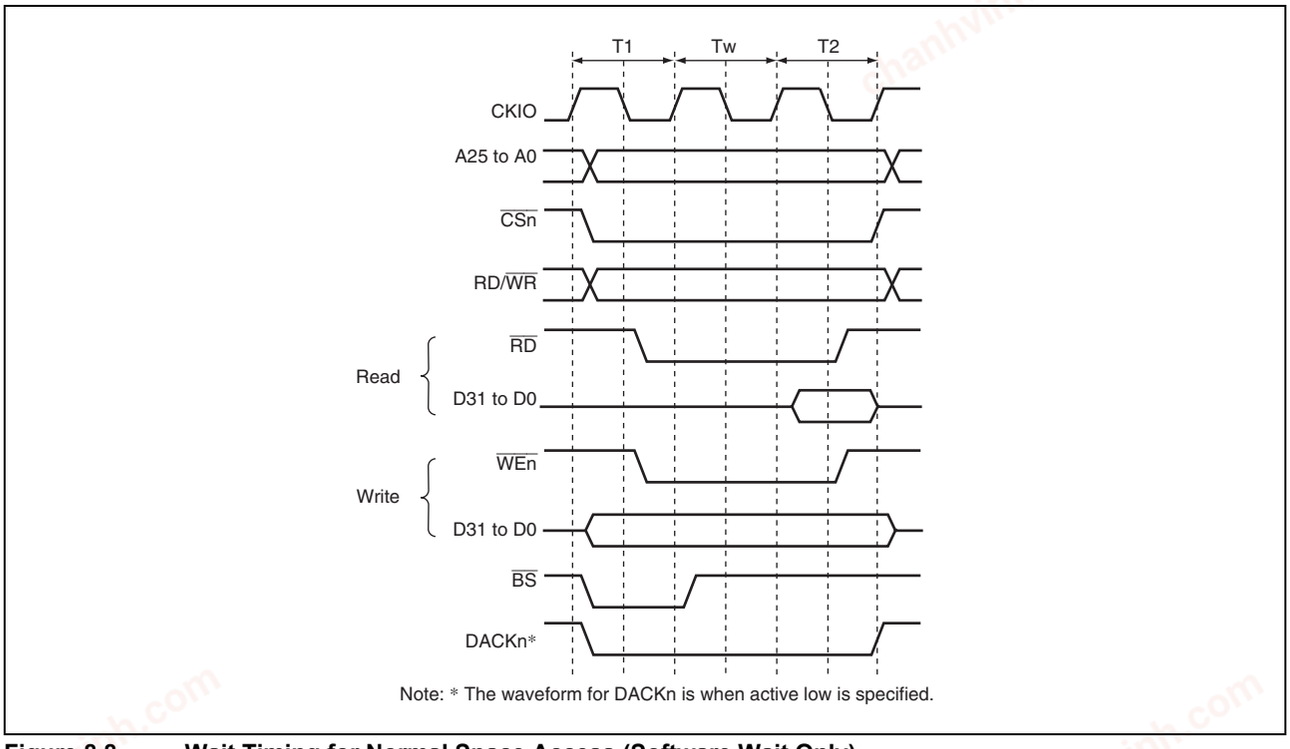


Figure 8.8 Wait Timing for Normal Space Access (Software Wait Only)

chanhvinh.com

When the WM bit in CSnWCR is cleared to 0, the external wait input $\overline{\text{WAIT}}$ signal is also sampled. $\overline{\text{WAIT}}$ pin sampling is shown in Figure 8.9. A 2-cycle wait is specified as a software wait. The $\overline{\text{WAIT}}$ signal is sampled on the falling edge of CKIO at the transition from the T1 or Tw cycle to the T2 cycle.

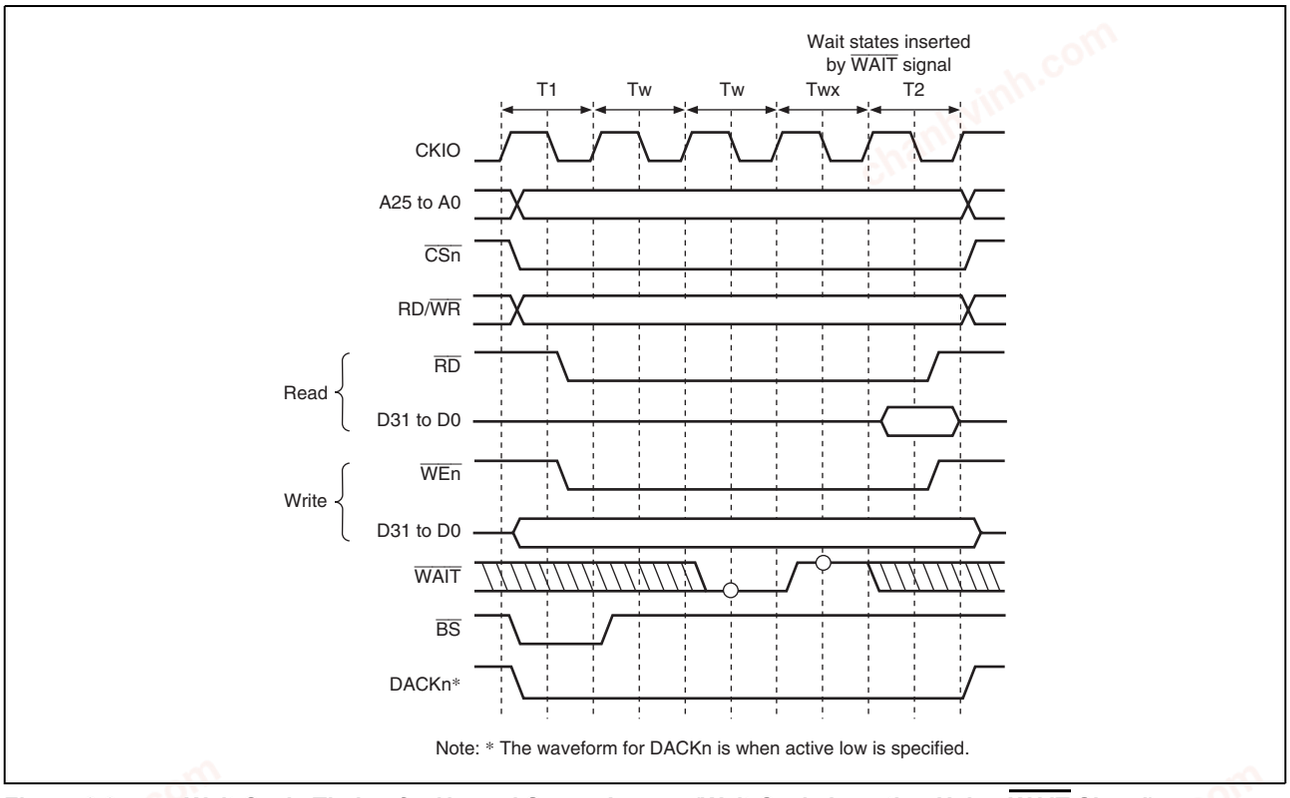


Figure 8.9 Wait Cycle Timing for Normal Space Access (Wait Cycle Insertion Using $\overline{\text{WAIT}}$ Signal)

chanhvinh.com

8.5.4 $\overline{\text{CSn}}$ Assert Period Expansion

The number of cycles from $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$, $\overline{\text{WEn}}$ assertion can be specified by setting bits SW1 and SW0 in CSnWCR. The number of cycles from $\overline{\text{RD}}$, $\overline{\text{WEn}}$ negation to $\overline{\text{CSn}}$ negation can be specified by setting bits HW1 and HW0. Therefore, a flexible interface to an external device can be obtained. Figure 8.10 shows an example. A Th cycle and a Tf cycle are added before and after an ordinary cycle, respectively. In these cycles, $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ are not asserted, while other signals are asserted. The data output is prolonged to the Tf cycle, and this prolongation is useful for devices with slow writing operations.

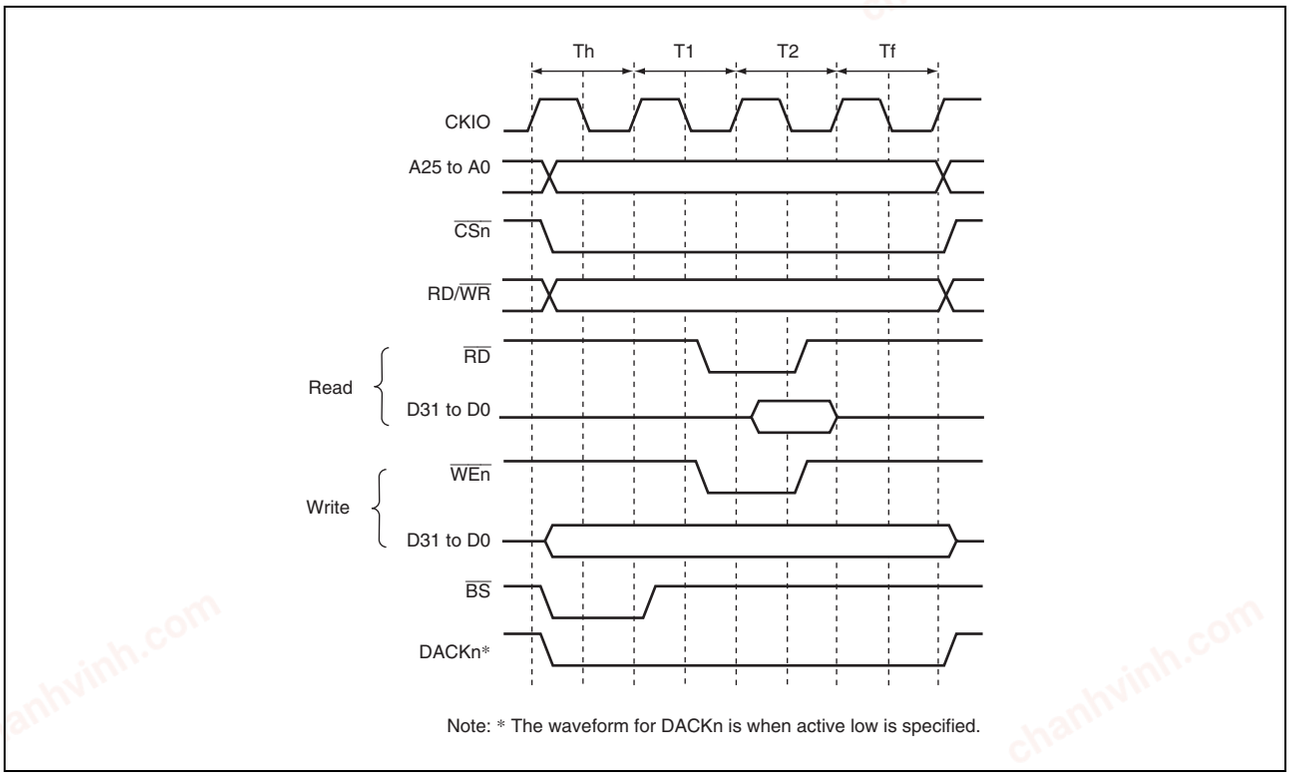


Figure 8.10 $\overline{\text{CSn}}$ Assert Period Expansion

chanhvinh.com

8.5.5 MPX-I/O Interface

Access timing for the MPX space is shown below. In the MPX space, $\overline{CS5}$, $\overline{AH}$, $\overline{RD}$, and $\overline{WEn}$ signals control the accessing. The basic access for the MPX space consists of 2 cycles of address output followed by an access to a normal space. The bus width for the address output cycle or the data input/output cycle is fixed to 8 bits or 16 bits. Alternatively, it can be 8 bits or 16 bits depending on the address to be accessed.

Output of the addresses D15 to D0 or D7 to D0 is performed from cycle Ta2 to cycle Ta3. Because cycle Ta1 has a high-impedance state, collisions of addresses and data can be avoided without inserting idle cycles, even in continuous access cycles. Address output is increased to 3 cycles by setting the MPXW bit in CS5WCR to 1.

The RD/$\overline{WR}$ signal is output at the same time as the $\overline{CS5}$ signal; it is high in the read cycle and low in the write cycle. The data cycle is the same as that in a normal space access.

The delay cycles the number of which is specified by SW[1:0] are inserted between cycle Ta3 and cycle T1. The delay cycles the number of which is specified by HW[1:0] are added after cycle T2.

Timing charts are shown in Figure 8.11 to Figure 8.13.

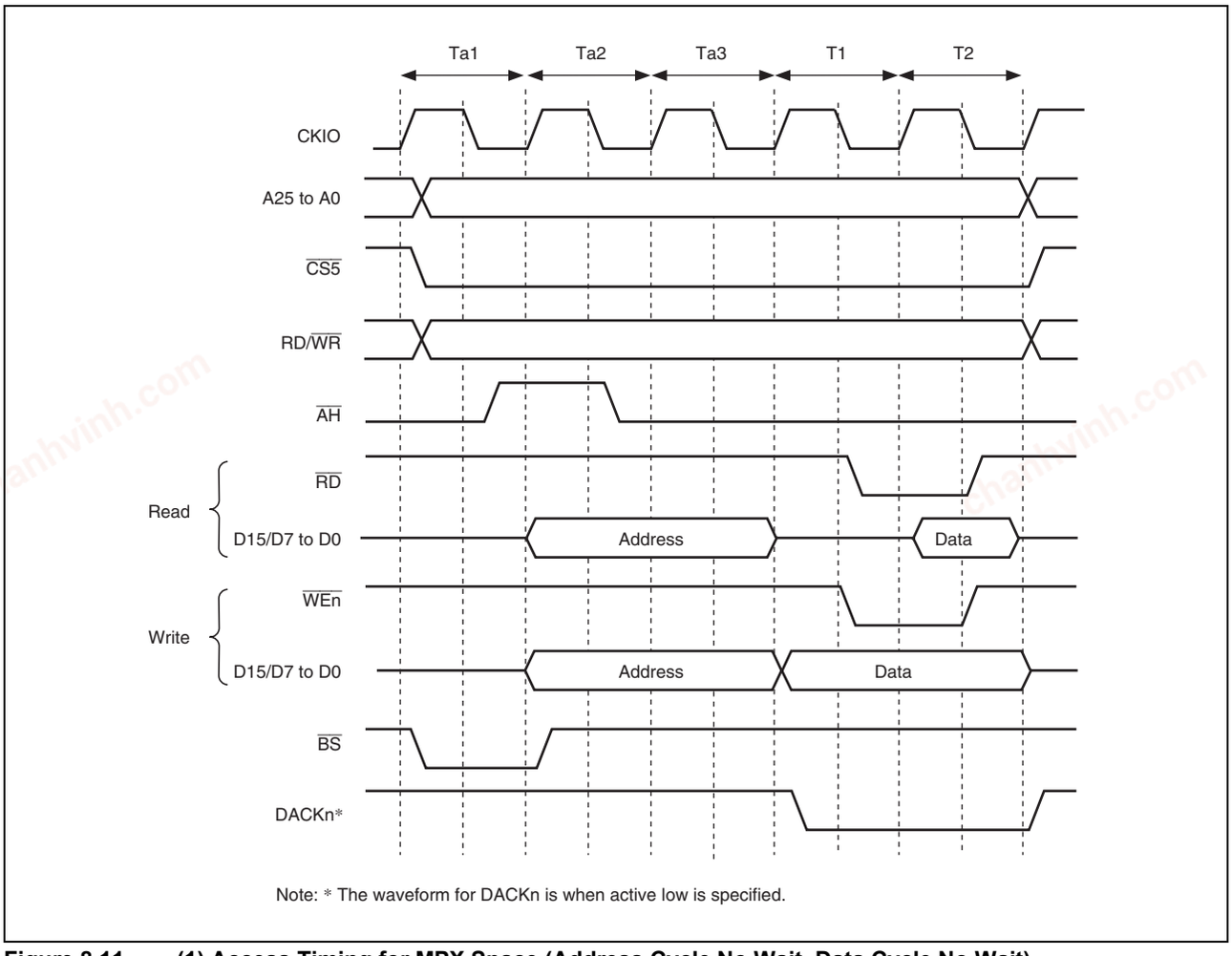


Figure 8.11 (1) Access Timing for MPX Space (Address Cycle No Wait, Data Cycle No Wait)

chanhvinh.com

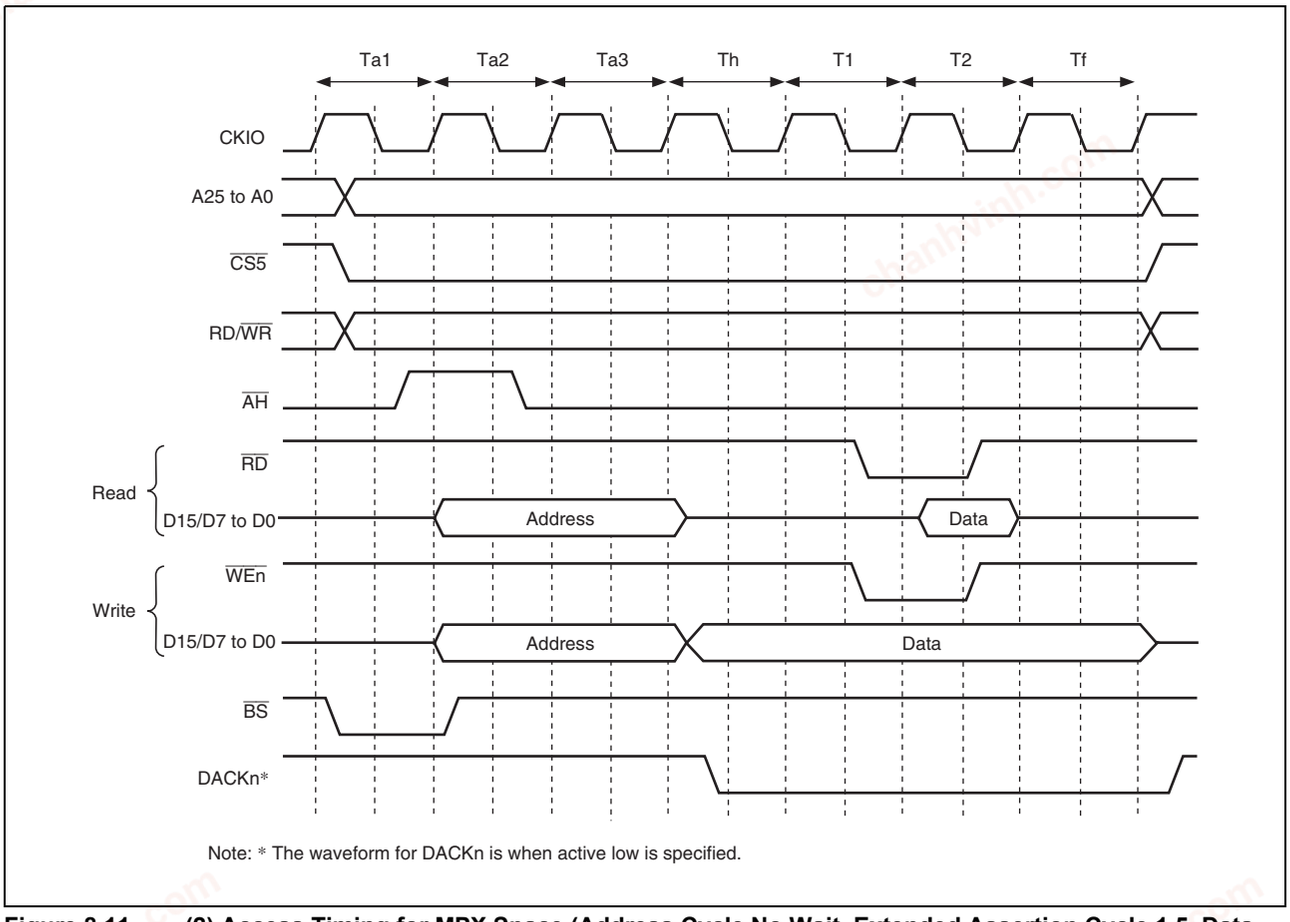


Figure 8.11 (2) Access Timing for MPX Space (Address Cycle No Wait, Extended Assertion Cycle 1.5, Data Cycle No Wait, Extended Negation Cycle 1.5)

chanhvinh.com

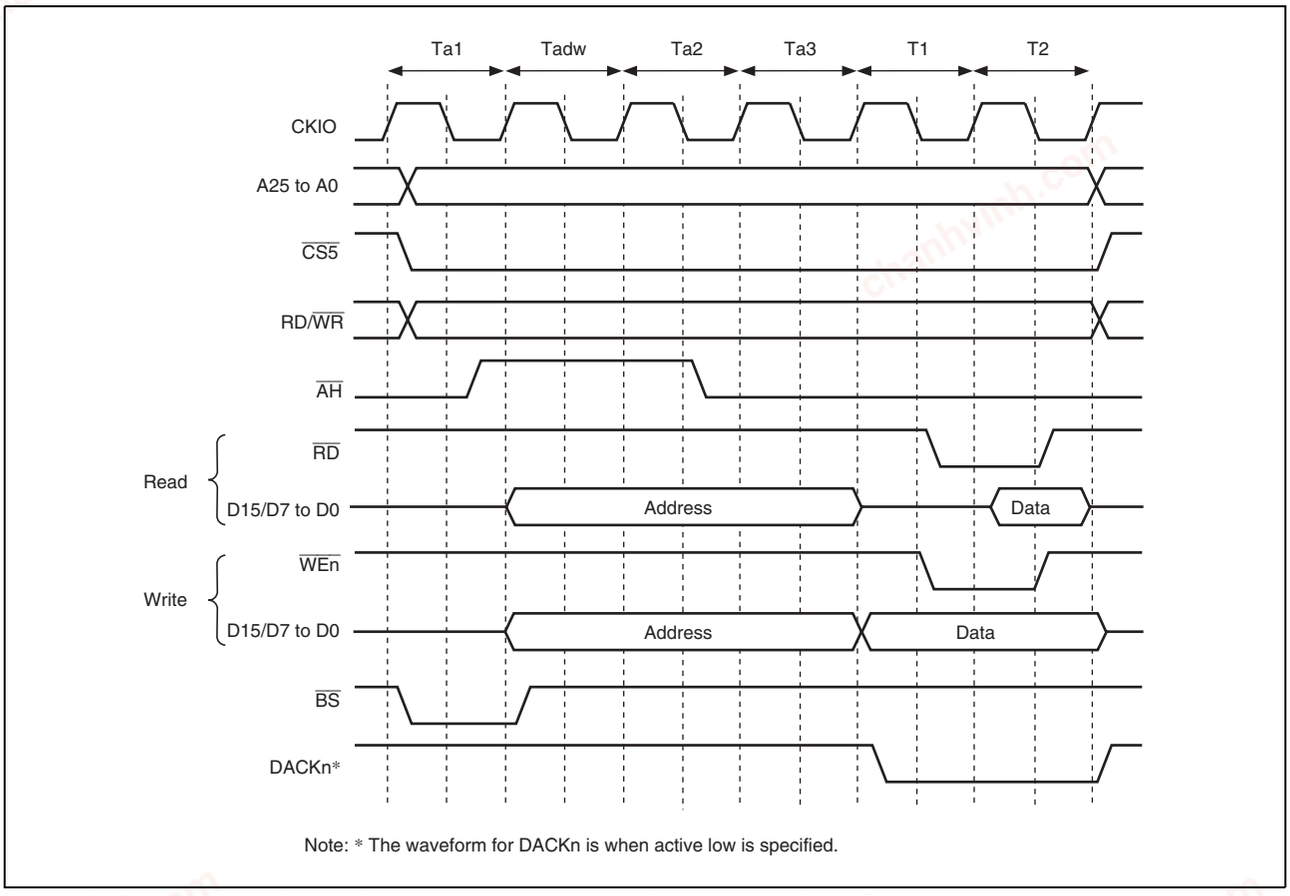


Figure 8.12 Access Timing for MPX Space (Address Cycle Wait 1, Data Cycle No Wait)

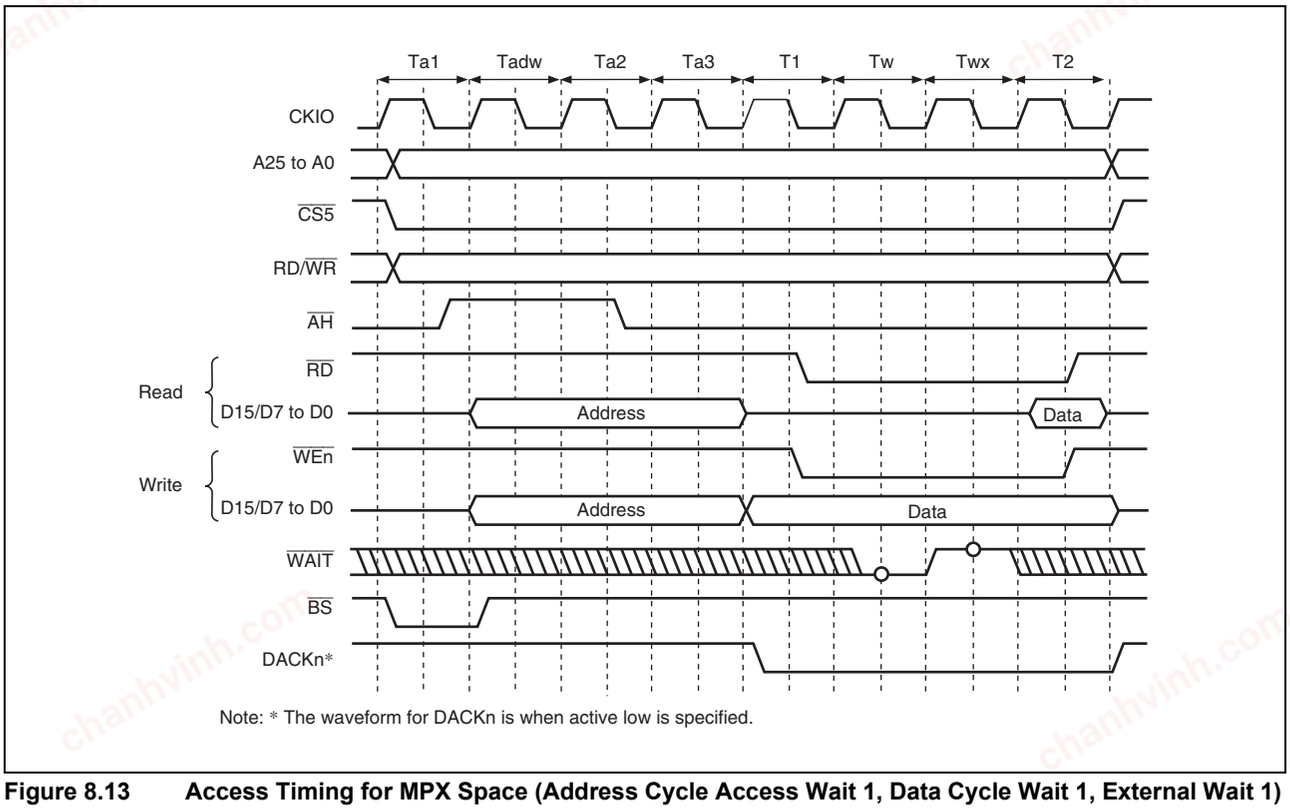


Figure 8.13 Access Timing for MPX Space (Address Cycle Access Wait 1, Data Cycle Wait 1, External Wait 1)

chanhvinh.com

8.5.6 SDRAM Interface

(1) SDRAM Direct Connection

The SDRAM that can be connected to this LSI is a product that has 11/12/13 bits of row address, 8/9/10 bits of column address, 4 or less banks, and uses the A10 pin for setting precharge mode in read and write command cycles.
The control signals for direct connection of SDRAM are $\overline{RAS}$, $\overline{CAS}$, RD/$\overline{WR}$, DQMUU, DQMUL, DQMLU, DQMLL, CKE, $\overline{CS2}$, and $\overline{CS3}$. All the signals other than $\overline{CS2}$ and $\overline{CS3}$ are common to all areas, and signals other than CKE are valid only when $\overline{CS2}$ or $\overline{CS3}$ is asserted. SDRAM can be connected to up to 2 spaces. The data bus width of the area that is connected to SDRAM is 16 bits or 32 bits.
Burst read/single write (burst length 1) and burst read/burst write (burst length 1) are supported as the SDRAM operating mode.
Commands for SDRAM can be specified by $\overline{RAS}$, $\overline{CAS}$, RD/$\overline{WR}$, and specific address signals. These commands supports:

- NOP
- Auto-refresh (REF)
- Self-refresh (SELF)
- All banks pre-charge (PALL)
- Specified bank pre-charge (PRE)
- Bank active (ACTV)
- Read (READ)
- Read with pre-charge (READA)
- Write (WRIT)
- Write with pre-charge (WRITA)
- Write mode register (MRS, EMRS)

The byte to be accessed is specified by DQMUU, DQMUL, DQMLU, and DQMLL. Reading or writing is performed for a byte whose corresponding DQMxx is low. For details on the relationship between DQMxx and the byte to be accessed, see section 8.5.1, Access Size and Data Alignment.

chanhvinh.com

Figure 8.14 and Figure 8.15 show examples of the connection of the SDRAM with the LSI.

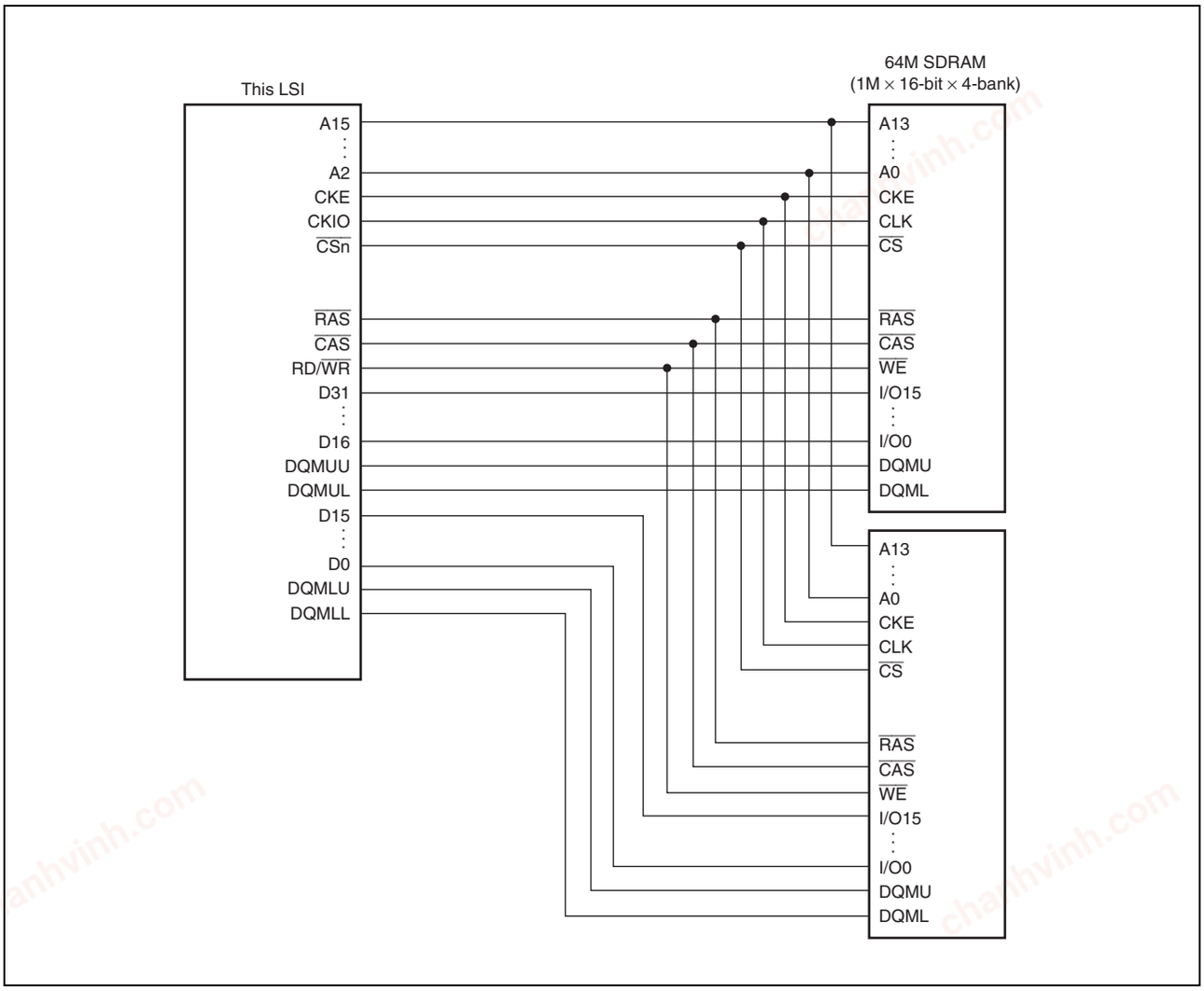


Figure 8.14 Example of 32-Bit Data Width SDRAM Connection

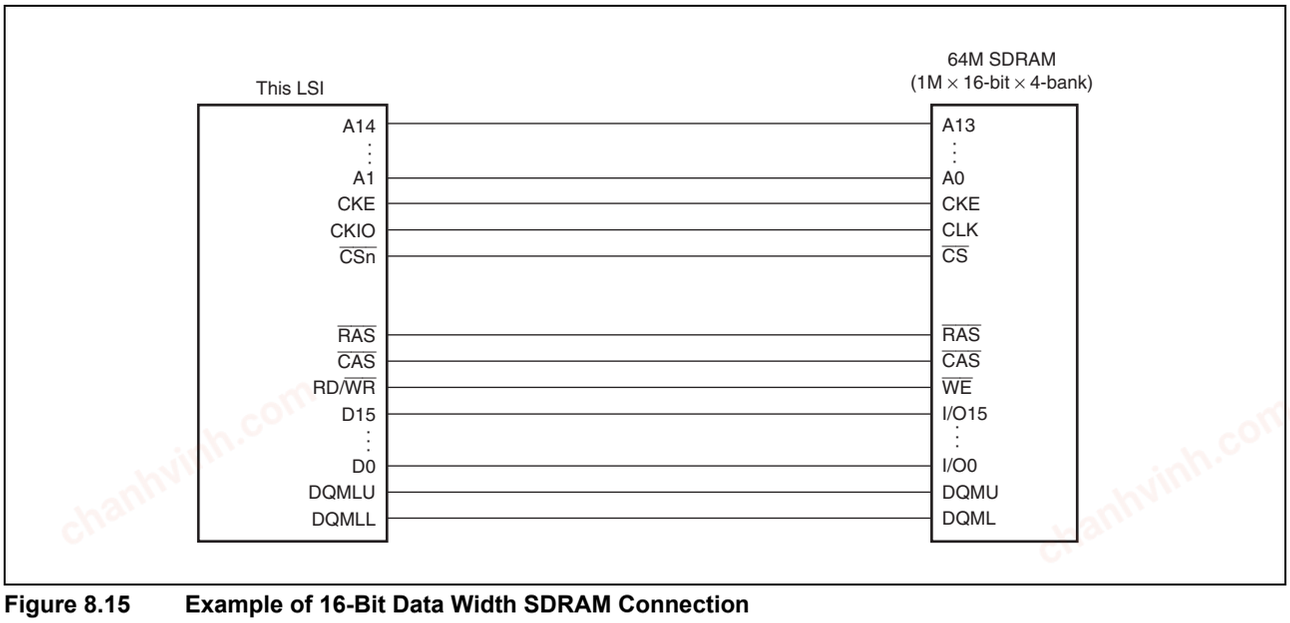


Figure 8.15 Example of 16-Bit Data Width SDRAM Connection

chanhvinh.com

(2) Address Multiplexing

An address multiplexing is specified so that SDRAM can be connected without external multiplexing circuitry according to the setting of bits BSZ[1:0] in CSnBCR and bits A2ROW[1:0], A2COL[1:0], A3ROW[1:0], and A3COL[1:0] in SDCR. Table 8.8 to Table 8.13 show the relationship between the settings of bits BSZ[1:0], A2ROW[1:0], A2COL[1:0], A3ROW[1:0], and A3COL[1:0] and the bits output at the address pins. Do not specify those bits in the manner other than this table, otherwise the operation of this LSI is not guaranteed. A25 to A18 are not multiplexed and the original values of address are always output at these pins.

When the data bus width is 16 bits (BSZ1 and BSZ0 = B'10), A0 of SDRAM specifies a 16-bit address. Therefore, connect this A0 pin of SDRAM to the A1 pin of the LSI; the A1 pin of SDRAM to the A2 pin of the LSI, and so on.

When the data bus width is 32 bits (BSZ1 and BSZ0 = B'11), A0 of SDRAM specifies a 32-bit address. Therefore, connect this A0 pin of SDRAM to the A2 pin of the LSI; the A1 pin of SDRAM to the A3 pin of the LSI, and so on.

Table 8.8 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (1)-1

	Setting			
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 bits)	**00 (11 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22*2	A22*2	A12 (BA1)	Specifies bank
A13	A21*2	A21*2	A11 (BA0)	
A12	A20	L/H*1	A10/AP	Specifies address/ precharge
A11	A19	A11	A9	Address
A10	A18	A10	A8	
A9	A17	A9	A7	
A8	A16	A8	A6	
A7	A15	A7	A5	
A6	A14	A6	A4	
A5	A13	A5	A3	
A4	A12	A4	A2	
A3	A11	A3	A1	
A2	A10	A2	A0	
A1	A9	A1		Unused
A0	A8	A0		
Example of connected memory				
64-Mbit product (512 Kwords × 32 bits × 4 banks, column 8 bits product): 1 16-Mbit product (512 Kwords × 16 bits × 2 banks, column 8 bits product): 2				

Note 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
Note 2. Bank address specification

chanhvinh.com

Table 8.8 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (1)-2

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 bits)	**01 (12 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23*2	A23*2	A13 (BA1)	Specifies bank
A14	A22*2	A22*2	A12 (BA0)	
A13	A21	A13	A11	Address
A12	A20	L/H*1	A10/AP	Specifies address/ precharge
A11	A19	A11	A9	Address
A10	A18	A10	A8	
A9	A17	A9	A7	
A8	A16	A8	A6	
A7	A15	A7	A5	
A6	A14	A6	A4	
A5	A13	A5	A3	
A4	A12	A4	A2	
A3	A11	A3	A1	
A2	A10	A2	A0	
A1	A9	A1		Unused
A0	A8	A0		
Example of connected memory				
128-Mbit product (1 Mwords × 32 bits × 4 banks, column 8 bits product): 1 64-Mbit product (1 Mwords × 16 bits × 4 banks, column 8 bits product): 2				

Note 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
Note 2. Bank address specification

chanhvinh.com

Table 8.9 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (2)-1

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 bits)	**01 (12 bits)**	**01 (9 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A26	A17		Unused
A16	A25	A16		
A15	A24*2	A24*2	A13 (BA1)	Specifies bank
A14	A23*2	A23*2	A12 (BA0)	
A13	A22	A13	A11	Address
A12	A21	L/H*1	A10/AP	Specifies address/ precharge
A11	A20	A11	A9	Address
A10	A19	A10	A8	
A9	A18	A9	A7	
A8	A17	A8	A6	
A7	A16	A7	A5	
A6	A15	A6	A4	
A5	A14	A5	A3	
A4	A13	A4	A2	
A3	A12	A3	A1	
A2	A11	A2	A0	
A1	A10	A1		Unused
A0	A9	A0		
Example of connected memory				
256-Mbit product (2 Mwords × 32 bits × 4 banks, column 9 bits product): 1 128-Mbit product (2 Mwords × 16 bits × 4 banks, column 9 bits product): 2				

Note 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
Note 2. Bank address specification

chanhvinh.com

Table 8.9 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (2)-2

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 bits)	**01 (12 bits)**	**10 (10 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A27	A17		Unused
A16	A26	A16		
A15	A25*2	A25*2	A13 (BA1)	Specifies bank
A14	A24*2	A24*2	A12 (BA0)	
A13	A23	A13	A11	Address
A12	A22	L/H*1	A10/AP	Specifies address/ precharge
A11	A21	A11	A9	Address
A10	A20	A10	A8	
A9	A19	A9	A7	
A8	A18	A8	A6	
A7	A17	A7	A5	
A6	A16	A6	A4	
A5	A15	A5	A3	
A4	A14	A4	A2	
A3	A13	A3	A1	
A2	A12	A2	A0	
A1	A11	A1		Unused
A0	A10	A0		
Example of connected memory				
512-Mbit product (4 Mwords × 32 bits × 4 banks, column 10 bits product): 1 256-Mbit product (4 Mwords × 16 bits × 4 banks, column 10 bits product): 2				

Note 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
Note 2. Bank address specification

chanhvinh.com

Table 8.10 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (3)

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 bits)	**10 (13 bits)**	**01 (9 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A26	A17		Unused
A16	A25*2	A25*2	A14 (BA1)	Specifies bank
A15	A24*2	A24*2	A13 (BA0)	
A14	A23	A14	A12	Address
A13	A22	A13	A11	
A12	A21	L/H*1	A10/AP	Specifies address/ precharge
A11	A20	A11	A9	Address
A10	A19	A10	A8	
A9	A18	A9	A7	
A8	A17	A8	A6	
A7	A16	A7	A5	
A6	A15	A6	A4	
A5	A14	A5	A3	
A4	A13	A4	A2	
A3	A12	A3	A1	
A2	A11	A2	A0	
A1	A10	A1		Unused
A0	A9	A0		
Example of connected memory				
512-Mbit product (4 Mwords × 32 bits × 4 banks, column 9 bits product): 1 256-Mbit product (4 Mwords × 16 bits × 4 banks, column 9 bits product): 2				

Note 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
Note 2. Bank address specification

chanhvinh.com

Table 8.11 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (4)-1

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**00 (11 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22	A14		
A13	A21	A21		
A12	A20*2	A20*2	A11 (BA0)	Specifies bank
A11	A19	L/H*1	A10/AP	Specifies address/ precharge
A10	A18	A10	A9	Address
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		Unused
Example of connected memory				
16-Mbit product (512 Kwords × 16 bits × 2 banks, column 8 bits product): 1				

Note 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
Note 2. Bank address specification

chanhvinh.com

Table 8.11 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (4)-2

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**01 (12 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22*2	A22*2	A13 (BA1)	Specifies bank
A13	A21*2	A21*2	A12 (BA0)	
A12	A20	A12	A11	Address
A11	A19	L/H*1	A10/AP	Specifies address/ precharge
A10	A18	A10	A9	Address
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		Unused
Example of connected memory				
64-Mbit product (1 Mwords × 16 bits × 4 banks, column 8 bits product): 1				

Note 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
Note 2. Bank address specification

chanhvinh.com

Table 8.12 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (5)-1

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**01 (12 bits)**	**01 (9 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A26	A17		Unused
A16	A25	A16		
A15	A24	A15		
A14	A23*2	A23*2	A13 (BA1)	Specifies bank
A13	A22*2	A22*2	A12 (BA0)	
A12	A21	A12	A11	Address
A11	A20	L/H*1	A10/AP	Specifies address/ precharge
A10	A19	A10	A9	Address
A9	A18	A9	A8	
A8	A17	A8	A7	
A7	A16	A7	A6	
A6	A15	A6	A5	
A5	A14	A5	A4	
A4	A13	A4	A3	
A3	A12	A3	A2	
A2	A11	A2	A1	
A1	A10	A1	A0	
A0	A9	A0		Unused
Example of connected memory				
128-Mbit product (2 Mwords × 16 bits × 4 banks, column 9 bits product): 1				

Note 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
Note 2. Bank address specification

chanhvinh.com

Table 8.12 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (5)-2

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**01 (12 bits)**	**10 (10 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A27	A17		Unused
A16	A26	A16		
A15	A25	A15		
A14	A24*2	A24*2	A13 (BA1)	Specifies bank
A13	A23*2	A23*2	A12 (BA0)	
A12	A22	A12	A11	Address
A11	A21	L/H*1	A10/AP	Specifies address/ precharge
A10	A20	A10	A9	Address
A9	A19	A9	A8	
A8	A18	A8	A7	
A7	A17	A7	A6	
A6	A16	A6	A5	
A5	A15	A5	A4	
A4	A14	A4	A3	
A3	A13	A3	A2	
A2	A12	A2	A1	
A1	A11	A1	A0	
A0	A10	A0		Unused
Example of connected memory				
256-Mbit product (4 Mwords × 16 bits × 4 banks, column 10 bits product): 1				

Note 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
Note 2. Bank address specification

chanhvinh.com

Table 8.13 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (6)-1

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**10 (13 bits)**	**01 (9 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A26	A17		Unused
A16	A25	A16		
A15	A24*2	A24*2	A14 (BA1)	Specifies bank
A14	A23*2	A23*2	A13 (BA0)	
A13	A22	A13	A12	Address
A12	A21	A12	A11	
A11	A20	L/H*1	A10/AP	Specifies address/ precharge
A10	A19	A10	A9	Address
A9	A18	A9	A8	
A8	A17	A8	A7	
A7	A16	A7	A6	
A6	A15	A6	A5	
A5	A14	A5	A4	
A4	A13	A4	A3	
A3	A12	A3	A2	
A2	A11	A2	A1	
A1	A10	A1	A0	
A0	A9	A0		Unused
Example of connected memory				
256-Mbit product (4 Mwords × 16 bits × 4 banks, column 9 bits product): 1				

Note 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
Note 2. Bank address specification

chanhvinh.com

Table 8.13 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (6)-2

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**10 (13 bits)**	**10 (10 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A27	A17		Unused
A16	A26	A16		
A15	A25*2	A25*2	A14 (BA1)	Specifies bank
A14	A24*2	A24*2	A13 (BA0)	
A13	A23	A13	A12	Address
A12	A22	A12	A11	
A11	A21	L/H*1	A10/AP	Specifies address/ precharge
A10	A20	A10	A9	Address
A9	A19	A9	A8	
A8	A18	A8	A7	
A7	A17	A7	A6	
A6	A16	A6	A5	
A5	A15	A5	A4	
A4	A14	A4	A3	
A3	A13	A3	A2	
A2	A12	A2	A1	
A1	A11	A1	A0	
A0	A10	A0		Unused
Example of connected memory				
512-Mbit product (8 Mwords × 16 bits × 4 banks, column 10 bits product): 1				

Note 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
Note 2. Bank address specification

chanhvinh.com

(3) Burst Read

A burst read occurs in the following cases with this LSI.

- Access size in reading is larger than data bus width.
- 16-, 32- or 64-byte transfer

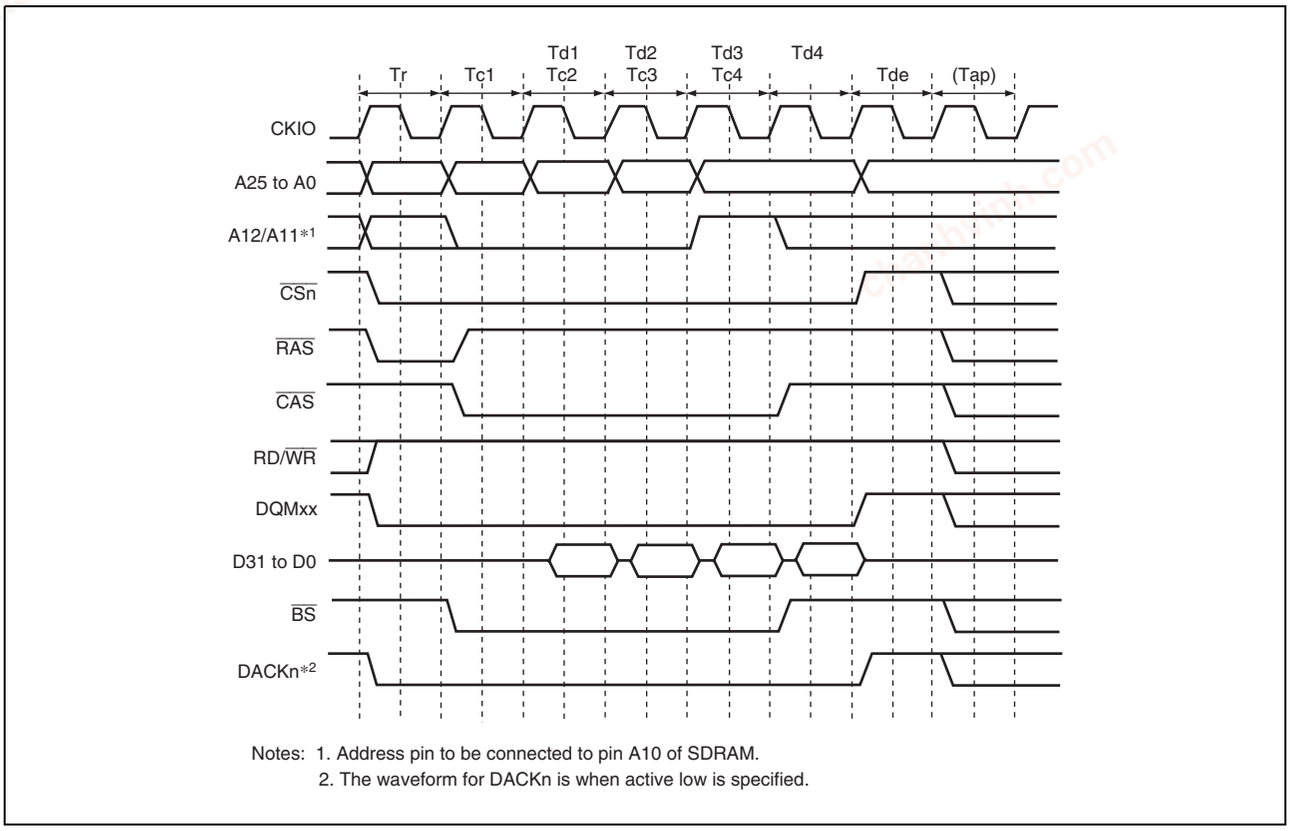
This LSI always accesses the SDRAM with burst length 1. For example, read access of burst length 1 is performed consecutively 8 times to read 16-byte continuous data from the SDRAM that is connected to a 16-bit data bus. This access is called the burst read with the burst number 8. Table 8.14 shows the relationship between the access size and the number of bursts.

Table 8.14 Relationship between Access Size and Number of Bursts

Bus Width	Access Size	Number of Bursts
16 bits	8 bits	1
	16 bits	1
	32 bits	2
	16 bytes	8
	32 bytes	16
	64 bytes	32
32 bits	8 bits	1
	16 bits	1
	32 bits	1
	16 bytes	4
	32 bytes	8
	64 bytes	16

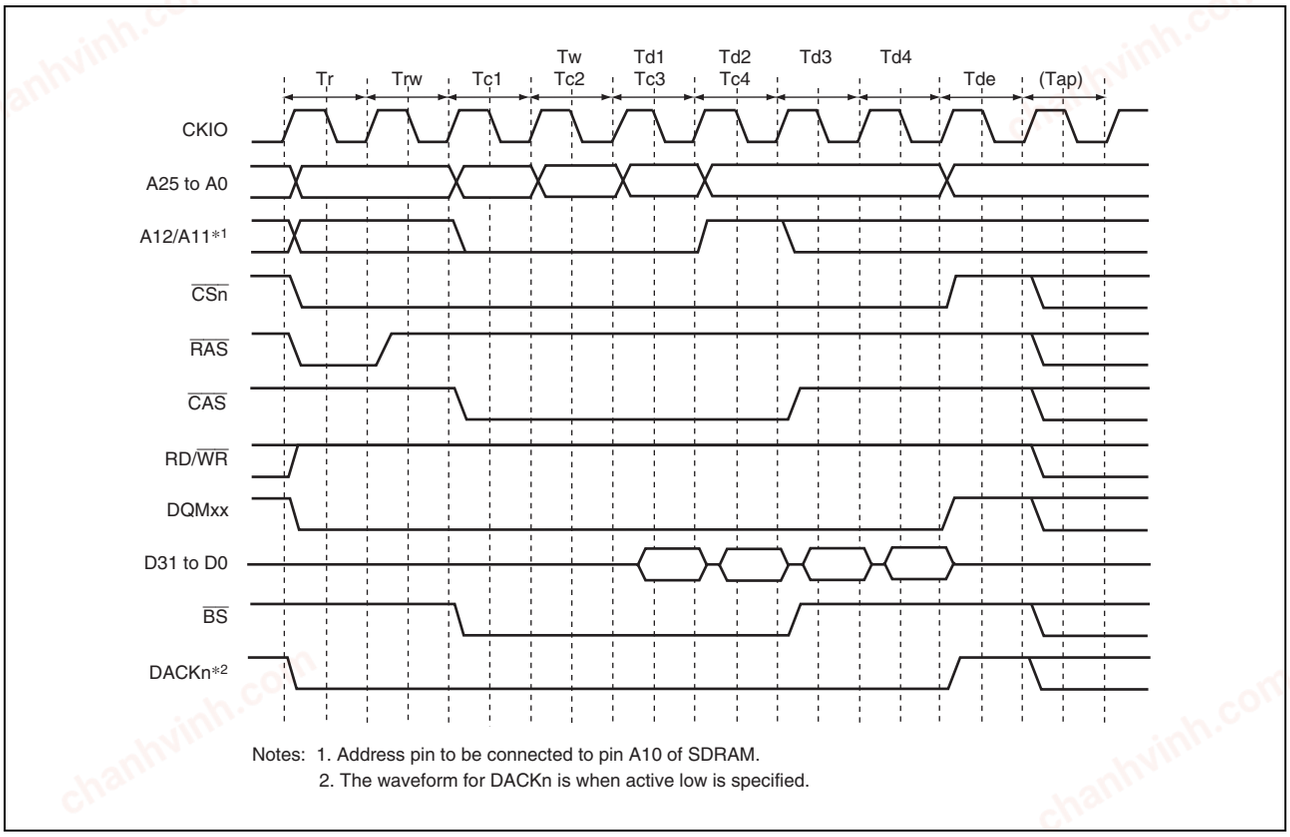
Figure 8.16 and Figure 8.17 show timing charts in burst read. In burst read, an ACTV command is output in the Tr cycle, the READ command is issued in the Tc1, Tc2, and Tc3 cycles, the READA command is issued in the Tc4 cycle, and the read data is received at the rising edge of the external clock (CKIO) in the Td1 to Td4 cycles. The Tap cycle is used to wait for the completion of an auto-precharge induced by the READA command in the SDRAM. In the Tap cycle, a new command will not be issued to the same bank. However, access to another CS space or another bank in the same SDRAM space is enabled. The number of Tap cycles is specified by the WTRP1 and WTRP0 bits in CS3WCR.
In this LSI, wait cycles can be inserted by specifying each bit in CS3WCR to connect the SDRAM in variable frequencies. Figure 8.17 shows an example in which wait cycles are inserted. The number of cycles from the Tr cycle where the ACTV command is output to the Tc1 cycle where the READ command is output can be specified using the WTRCD1 and WTRCD0 bits in CS3WCR. If the WTRCD1 and WTRCD0 bits specify one cycle or more, a Trw cycle where the NOP command is issued is inserted between the Tr cycle and Tc1 cycle. The number of cycles from the Tc1 cycle where the READ command is output to the Td1 cycle where the read data is latched can be specified for the CS2 and CS3 spaces independently, using the A2CL1 and A2CL0 bits in CS2WCR or the A3CL1 and A3CL0 bits in CS3WCR. The number of cycles from Tc1 to Td1 corresponds to the SDRAM CAS latency. The CAS latency for the SDRAM is normally defined as up to three cycles. However, the CAS latency in this LSI can be specified as 1 to 4 cycles. This CAS latency can be achieved by connecting a latch circuit between this LSI and the SDRAM.
A Tde cycle is an idle cycle required to transfer the read data into this LSI and occurs once for every burst read or every single read.

chanhvinh.com



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.16 Burst Read Basic Timing (CAS Latency 1, Auto Pre-Charge)



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

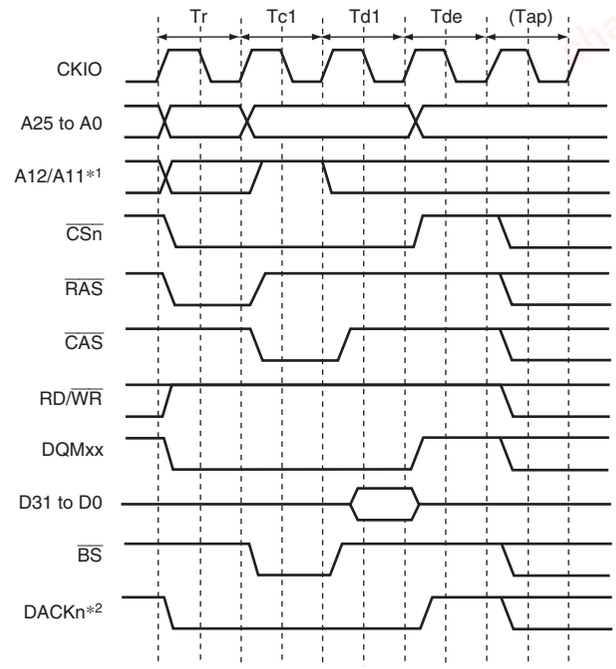
Figure 8.17 Burst Read Wait Specification Timing (CAS Latency 2, WTRCD[1:0] = 1 Cycle, Auto Pre-Charge)

chanhvinh.com

(4) Single Read

A read access ends in one cycle when the data bus width is larger than or equal to the access size. As the SDRAM is set to the burst read with the burst length 1, only the required data is output. A read access that ends in one cycle is called single read.

Figure 8.18 shows the single read basic timing.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.18 Basic Timing for Single Read (CAS Latency 1, Auto Pre-Charge)

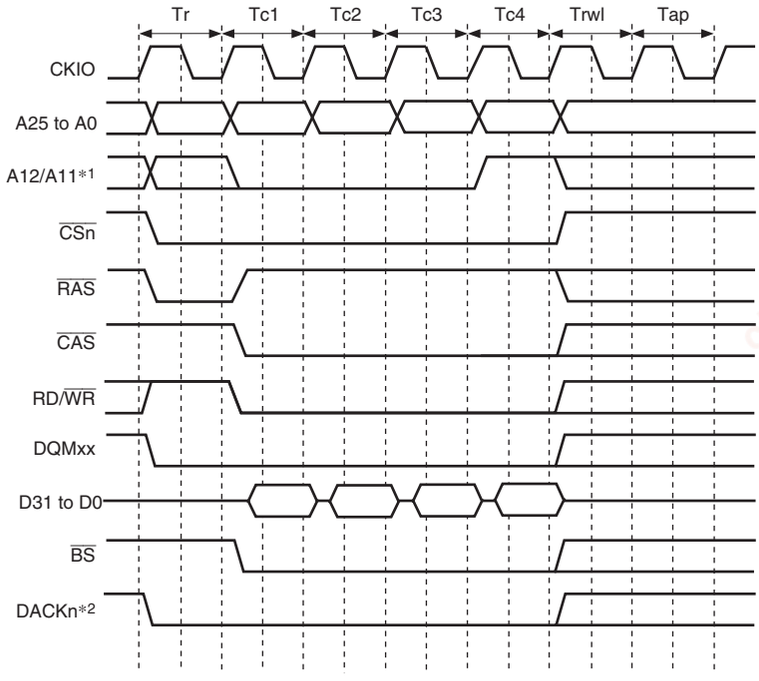
chanhvinh.com

(5) Burst Write

A burst write occurs in the following cases in this LSI.

- Access size in writing is larger than data bus width.
- 16-, 32- or 64-byte transfer

This LSI always accesses SDRAM with burst length 1. For example, write access of burst length 1 is performed continuously 8 times to write 16-byte continuous data to the SDRAM that is connected to a 16-bit data bus. This access is called burst write with the burst number 8. The relationship between the access size and the number of bursts is shown in Table 8.14. Figure 8.19 shows a timing chart for burst writes. In burst write, an ACTV command is output in the Tr cycle, the WRIT command is issued in the Tc1, Tc2, and Tc3 cycles, and the WRITA command is issued to execute an auto-precharge in the Tc4 cycle. In the write cycle, the write data is output simultaneously with the write command. After the write command with the auto-precharge is output, the Trw1 cycle that waits for the auto-precharge initiation is followed by the Tap cycle that waits for completion of the auto-precharge induced by the WRITA command in the SDRAM. Between the Trwl and the Tap cycle, a new command will not be issued to the same bank. However, access to another CS space or another bank in the same SDRAM space is enabled. The number of Trw1 cycles is specified by the TRWL1 and TRWL0 bits in CS3WCR. The number of Tap cycles is specified by the WTRP1 and WTRP0 bits in CS3WCR.



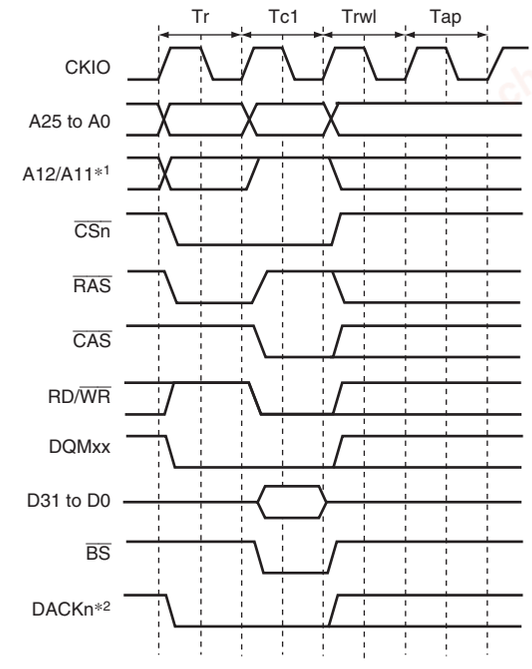
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.19 Basic Timing for Burst Write (Auto Pre-Charge)

chanhvinh.com

(6) Single Write

A write access ends in one cycle when the data bus width is larger than or equal to access size. As a single write or burst write with burst length 1 is set in SDRAM, only the required data is output. The write access that ends in one cycle is called single write. Figure 8.20 shows the single write basic timing.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.20 Single Write Basic Timing (Auto-Precharge)

chanhvinh.com

(7) Bank Active

The SDRAM bank function can be used to support high-speed access to the same row address. When the BACTV bit in SDCR is 1, access is performed using commands without auto-precharge (READ or WRIT). This function is called bank-active function. This function is valid only for area 3. When area 3 is set to bank-active mode, area 2 should be set to normal space or SRAM with byte selection. When areas 2 and 3 are both set to SDRAM, auto precharge mode must be set.

When the bank-active function is used, precharging is not performed when the access ends. When accessing the same row address in the same bank, it is possible to issue the READ or WRIT command immediately, without issuing an ACTV command. As SDRAM is internally divided into several banks, it is possible to activate one row address in each bank. If the next access is to a different row address, a PRE command is first issued to precharge the relevant bank, then when precharging is completed, the access is performed by issuing an ACTV command followed by a READ or WRIT command. If this is followed by an access to a different row address, the access time will be longer because of the precharging performed after the access request is issued. The number of cycles between issuance of the PRE command and the ACTV command is determined by the WTRP1 and WTPR0 bits in CS3WCR.

In a write, when an auto-precharge is performed, a command cannot be issued to the same bank for a period of Trwl + Tap cycles after issuance of the WRITA command. When bank active mode is used, READ or WRIT commands can be issued successively if the row address is the same. The number of cycles can thus be reduced by Trwl + Tap cycles for each write.

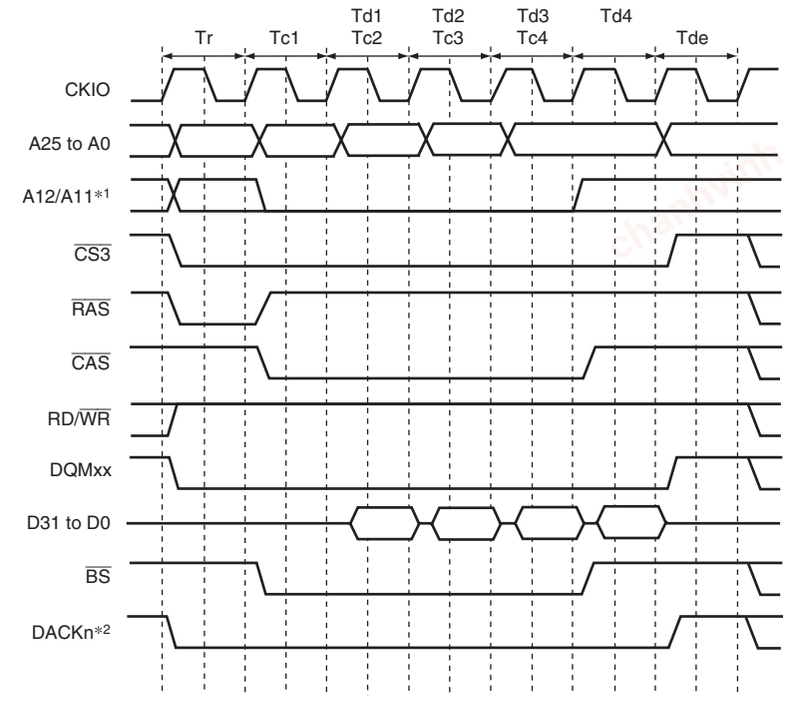
There is a limit on tRAS, the time for placing each bank in the active state. If there is no guarantee that there will not be a cache hit and another row address will be accessed within the period in which this value is maintained by program execution, it is necessary to set auto-refresh and set the refresh cycle to no more than the maximum value of tRAS.

A burst read cycle without auto-precharge is shown in Figure 8.21, a burst read cycle for the same row address in Figure 8.22, and a burst read cycle for different row addresses in Figure 8.23. Similarly, a single write cycle without auto-precharge is shown in Figure 8.24, a single write cycle for the same row address in Figure 8.25, and a single write cycle for different row addresses in Figure 8.26.

In Figure 8.22, a Tnop cycle in which no operation is performed is inserted before the Tc cycle that issues the READ command. The Tnop cycle is inserted to acquire two cycles of CAS latency for the DQMxx signal that specifies the read byte in the data read from the SDRAM. If the CAS latency is specified as two cycles or more, the Tnop cycle is not inserted because the two cycles of latency can be acquired even if the DQMxx signal is asserted after the Tc cycle.

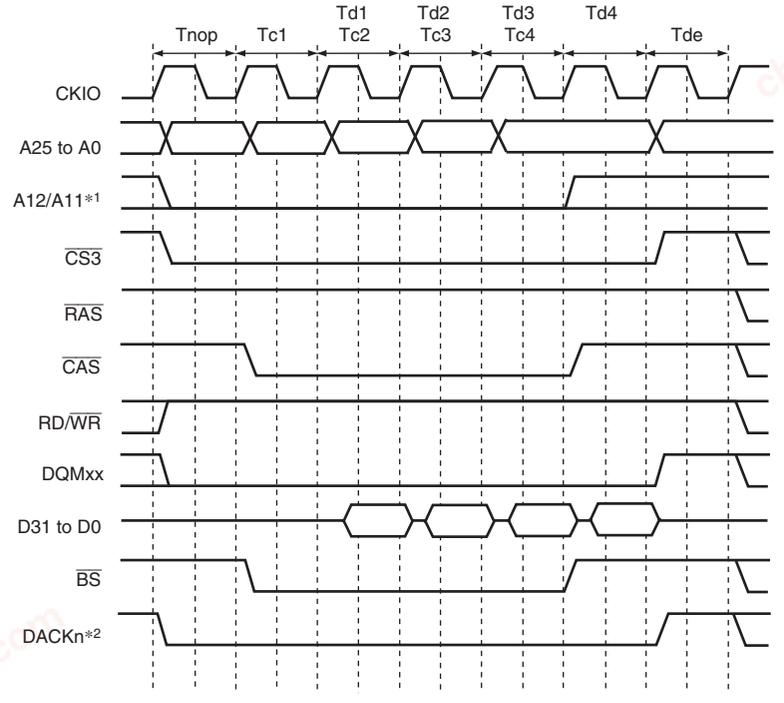
When bank active mode is set, if only access cycles to the respective banks in the area 3 space are considered, as long as access cycles to the same row address continue, the operation starts with the cycle in Figure 8.21 or Figure 8.24, followed by repetition of the cycle in Figure 8.22 or Figure 8.25. An access to a different area during this time has no effect. If there is an access to a different row address in the bank active state, the bus cycle in Figure 8.23 or Figure 8.26 is executed instead of that in Figure 8.22 or Figure 8.25. In bank active mode, too, all banks become inactive after a refresh cycle.

chanhvinh.com



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

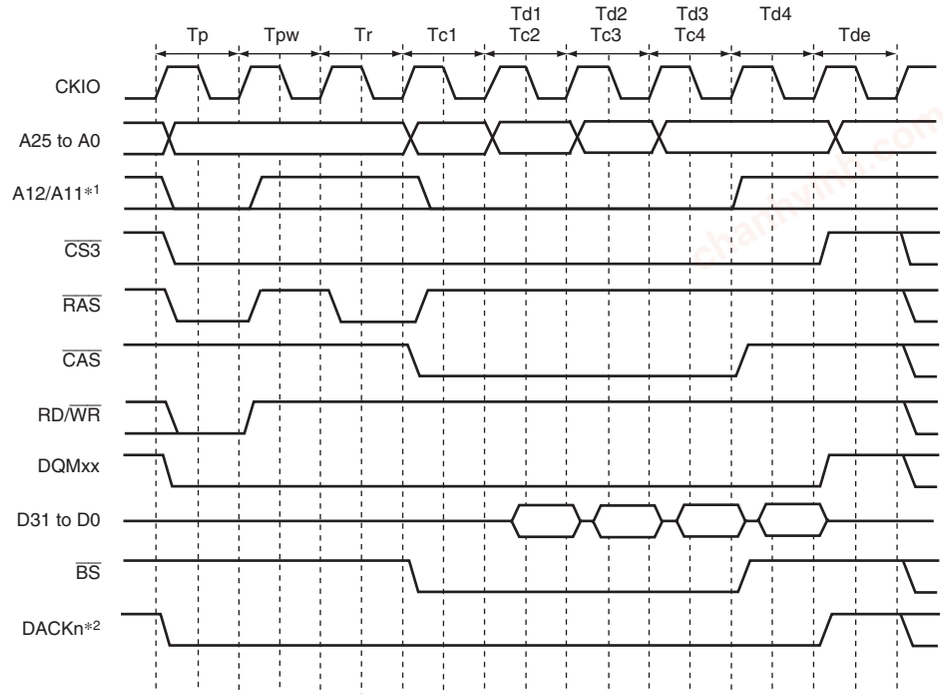
Figure 8.21 Burst Read Timing (Bank Active, Different Bank, CAS Latency 1)



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

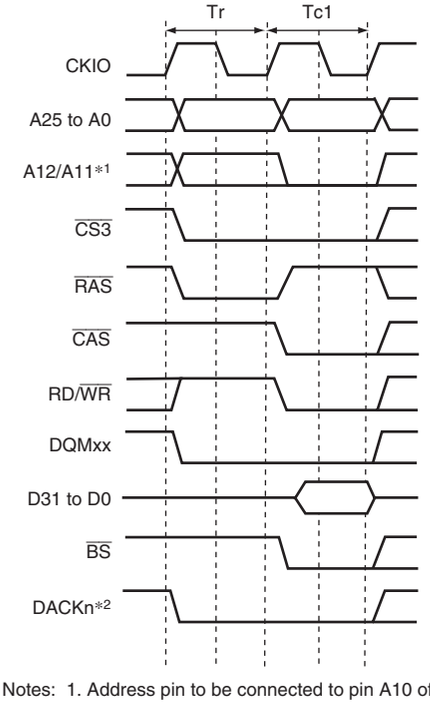
Figure 8.22 Burst Read Timing (Bank Active, Same Row Addresses in the Same Bank, CAS Latency 1)

chanhvinh.com



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

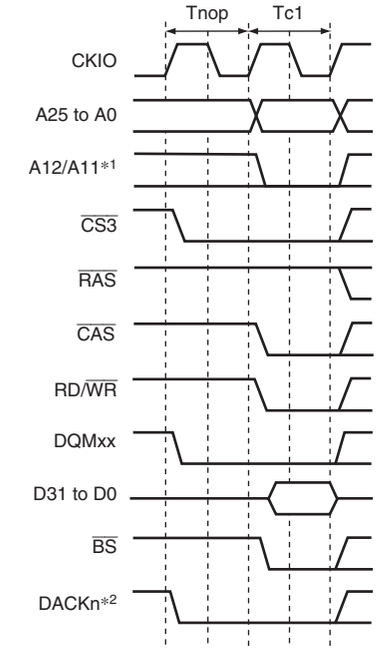
Figure 8.23 Burst Read Timing (Bank Active, Different Row Addresses in the Same Bank, CAS Latency 1)



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

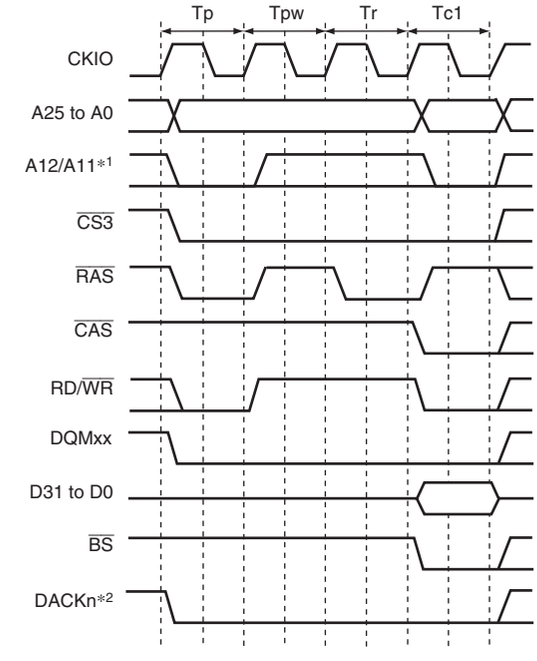
Figure 8.24 Single Write Timing (Bank Active, Different Bank)

chanhvinh.com



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.25 Single Write Timing (Bank Active, Same Row Addresses in the Same Bank)



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.26 Single Write Timing (Bank Active, Different Row Addresses in the Same Bank)

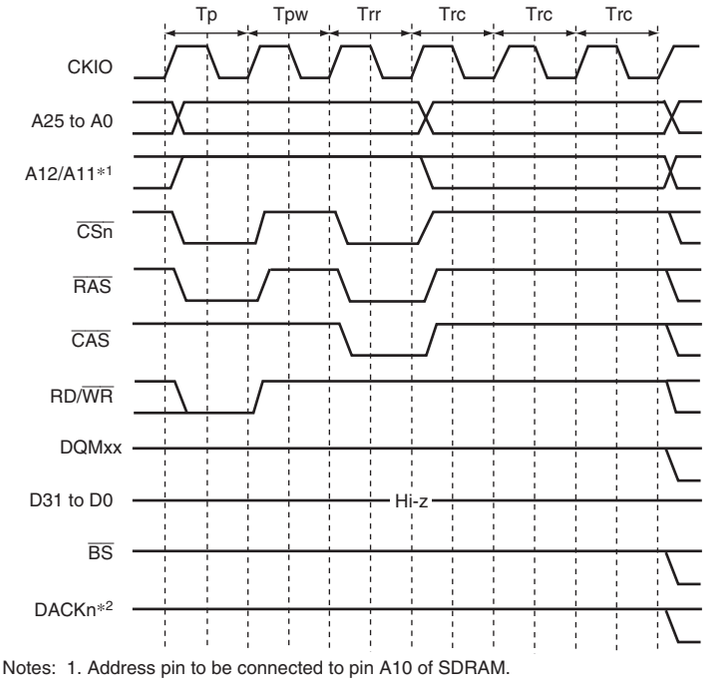
chanhvinh.com

(8) Refreshing

This module has a function for controlling SDRAM refreshing. Auto-refreshing can be performed by clearing the RMODE bit to 0 and setting the RFSH bit to 1 in SDCR. A continuous refreshing can be performed by setting the RRC2 to RRC0 bits in RTCSR. If SDRAM is not accessed for a long period, self-refresh mode, in which the power consumption for data retention is low, can be activated by setting both the RMODE bit and the RFSH bit to 1.

(a) Auto-refreshing

Refreshing is performed at intervals determined by the input clock selected by bits CKS2 to CKS0 in RTCSR, and the value set by in RTCOR. The value of bits CKS2 to CKS0 in RTCOR should be set so as to satisfy the refresh interval stipulation for the SDRAM used. First make the settings for RTCOR, RTCNT, and the RMODE and RFSH bits in SDCR, and then make the CKS2 to CKS0 and RRC2 to RRC0 settings. When the clock is selected by bits CKS2 to CKS0, RTCNT starts counting up from the value at that time. The RTCNT value is constantly compared with the RTCOR value, and if the two values are the same, a refresh request is generated and an auto-refresh is performed for the number of times specified by the RRC2 to RRC0. At the same time, RTCNT is cleared to zero and the count-up is restarted.
Figure 8.27 shows the auto-refresh cycle timing. After starting the auto refreshing, PALL command is issued in the Tp cycle to make all the banks to pre-charged state from active state when some bank is being pre-charged. Then REF command is issued in the Trr cycle after inserting idle cycles of which number is specified by the WTRP1 and WTRP0 bits in CS3WCR. A new command is not issued for the duration of the number of cycles specified by the WTRC1 and WTRC0 bits in CS3WCR after the Trr cycle. The WTRC1 and WTRC0 bits must be set so as to satisfy the SDRAM refreshing cycle time stipulation (tRC). An idle cycle is inserted between the Tp cycle and Trr cycle when the setting value of the WTRP1 and WTRP0 bits in CS3WCR is longer than or equal to 1 cycle.



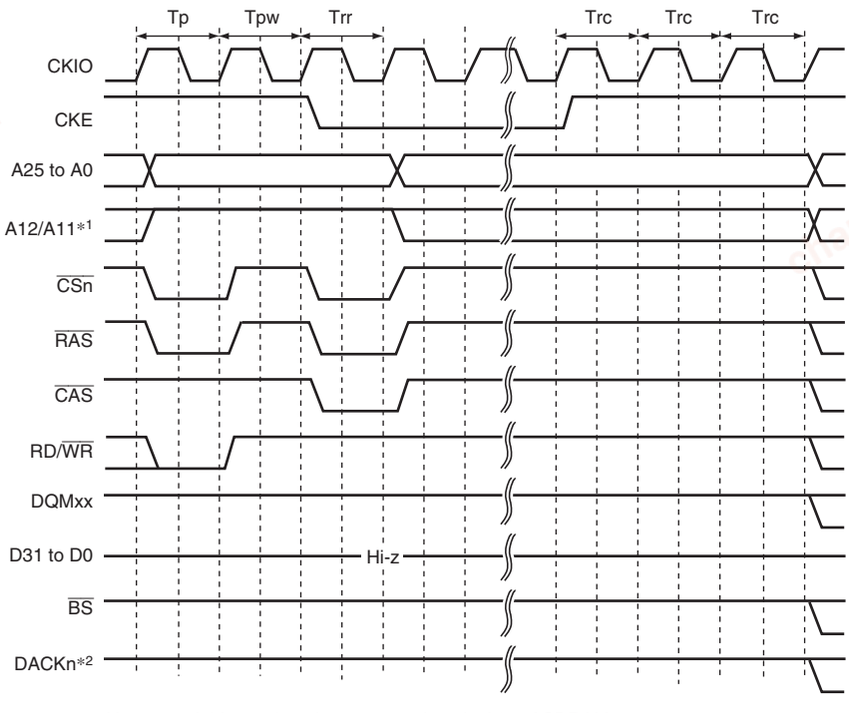
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.27 Auto-Refresh Timing

chanhvinh.com

(b) Self-refreshing

Self-refresh mode is a kind of standby mode, in which the refresh timing and refresh addresses are generated within the SDRAM. Self-refreshing is activated by setting both the RMODE bit and the RFSH bit in SDCR to 1. After starting the self-refreshing, PALL command is issued in Tp cycle after the completion of the pre-charging bank. A SELF command is then issued after inserting idle cycles of which number is specified by the WTRP1 and WTRP0 bits in CS3WSR.
SDRAM cannot be accessed while in the self-refresh state. Self-refresh mode is cleared by clearing the RMODE bit to 0. After self-refresh mode has been cleared, command issuance is disabled for the number of cycles specified by the WTRC1 and WTRC0 bits in CS3WCR.
Self-refresh timing is shown in Figure 8.28. Settings must be made so that self-refresh clearing and data retention are performed correctly, and auto-refreshing is performed at the correct intervals. When self-refreshing is activated from the state in which auto-refreshing is set, auto-refreshing is restarted if the RFSH bit is set to 1 and the RMODE bit is cleared to 0 when self-refresh mode is cleared. If the transition from clearing of self-refresh mode to the start of auto-refreshing takes time, this time should be taken into consideration when setting the initial value of RTCNT. Making the RTCNT value 1 less than the RTCOR value will enable refreshing to be started immediately.
After self-refreshing has been set, the self-refresh state continues even if the chip standby state is entered using the LSI standby function, and is maintained even after recovery from standby mode due to an interrupt. Note that the necessary signals such as CKE must be driven even in standby state by setting the HIZCNT bit in CMNCR to 1.
In case of a power-on reset, the bus state controller's registers are initialized, and therefore the self-refresh state is cleared.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.28 Self-Refresh Timing

chanhvinh.com

(9) Relationship between Refresh Requests and Bus Cycles

If a refresh request occurs during bus cycle execution, the refresh cycle must wait for the bus cycle to be completed. If a new refresh request occurs while waiting for the previous refresh request, the previous refresh request is deleted. To refresh correctly, a bus cycle longer than the refresh interval must be prevented from occurring.

(10) Power-Down Mode

If the PDOWN bit in SDCR is set to 1, the SDRAM is placed in power-down mode by bringing the CKE signal to the low level in the non-access cycle. This power-down mode can effectively lower the power consumption in the non-access cycle. However, note that if an access occurs in power-down mode, a cycle of overhead occurs because a cycle is needed to assert the CKE in order to cancel the power-down mode.
Figure 8.29 shows the access timing in power-down mode.

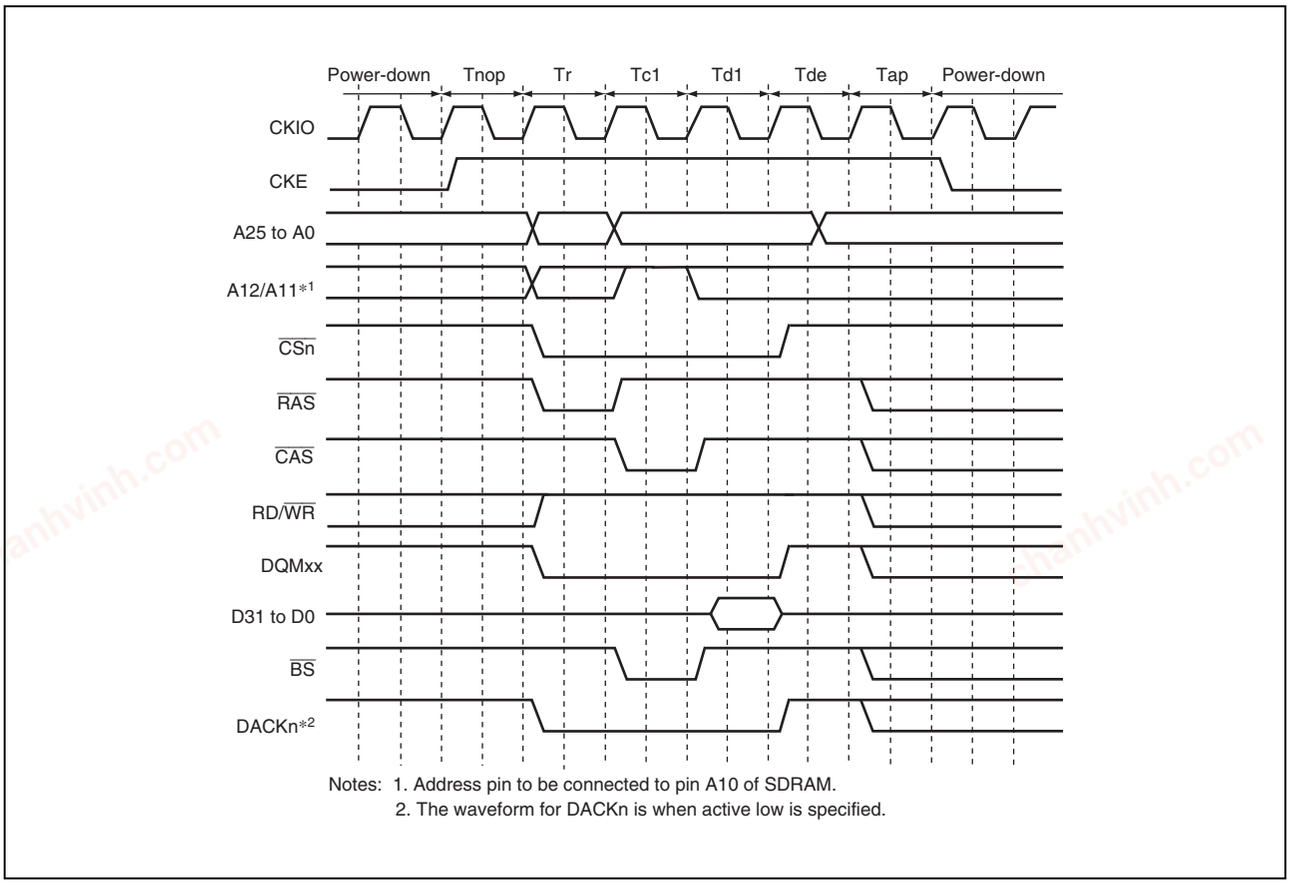


Figure 8.29 Power-Down Mode Access Timing

chanhvinh.com

(11) Power-On Sequence

In order to use SDRAM, mode setting must first be made for SDRAM after the pose interval specified for the SDRAM to be used after powering on. The pose interval should be obtained by a power-on reset generating circuit or software.
To perform SDRAM initialization correctly, the registers of this module must first be set, followed by a write to the SDRAM mode register. In SDRAM mode register setting, the address signal value at that time is latched by a combination of the $\overline{CSn}$, $\overline{RAS}$, $\overline{CAS}$, and RD/$\overline{WR}$ signals. If the value to be set is X, the bus state controller provides for value X to be written to the SDRAM mode register by performing a 16-bit write to address H'3FFFD000 + X for area 2 SDRAM, and to address H'3FFFE000 + X for area 3 SDRAM. In this operation the data is ignored, but the mode write is performed as a byte-size access. To set burst read/single write or burst read/burst write (CAS latency 2 to 3, wrap type = sequential, and burst length 1) supported by the LSI, arbitrary data is written in 16 bits to the access addresses shown in Table 8.15. In this time 0 is output at the external address pins of A12 or later.

Table 8.15 Access Address in SDRAM Mode Register Write

- Setting for Area 2
  Burst read/single write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'3FFFD440	H'0000440
	3	H'3FFFD460	H'0000460
32 bits	2	H'3FFFD880	H'0000880
	3	H'3FFFD8C0	H'00008C0

Burst read/burst write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'3FFFD040	H'0000040
	3	H'3FFFD060	H'0000060
32 bits	2	H'3FFFD080	H'0000080
	3	H'3FFFD0C0	H'00000C0

- Setting for Area 3
  Burst read/single write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'3FFFE440	H'0000440
	3	H'3FFFE460	H'0000460
32 bits	2	H'3FFFE880	H'0000880
	3	H'3FFFE8C0	H'00008C0

Burst read/burst write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'3FFFE040	H'0000040
	3	H'3FFFE060	H'0000060
32 bits	2	H'3FFFE080	H'0000080
	3	H'3FFFE0C0	H'00000C0

chanhvinh.com

Mode register setting timing is shown in Figure 8.30. A PALL command (all bank pre-charge command) is firstly issued. A REF command (auto refresh command) is then issued 8 times. An MRS command (mode register write command) is finally issued. Idle cycles, of which number is specified by the WTRP1 and WTRP0 bits in CS3WCR, are inserted between the PALL and the first REF. Idle cycles, of which number is specified by the WTRC1 and WTRC0 bits in CS3WCR, are inserted between REF and REF, and between the 8th REF and MRS. One or more idle cycles are inserted between the MRS and a command to be issued next.
It is necessary to keep idle time of certain cycles for SDRAM before issuing PALL command after power-on. Refer to the manual of the SDRAM for the idle time to be needed. When the pulse width of the reset signal is longer than the idle time, mode register setting can be started immediately after the reset, but care should be taken when the pulse width of the reset signal is shorter than the idle time.

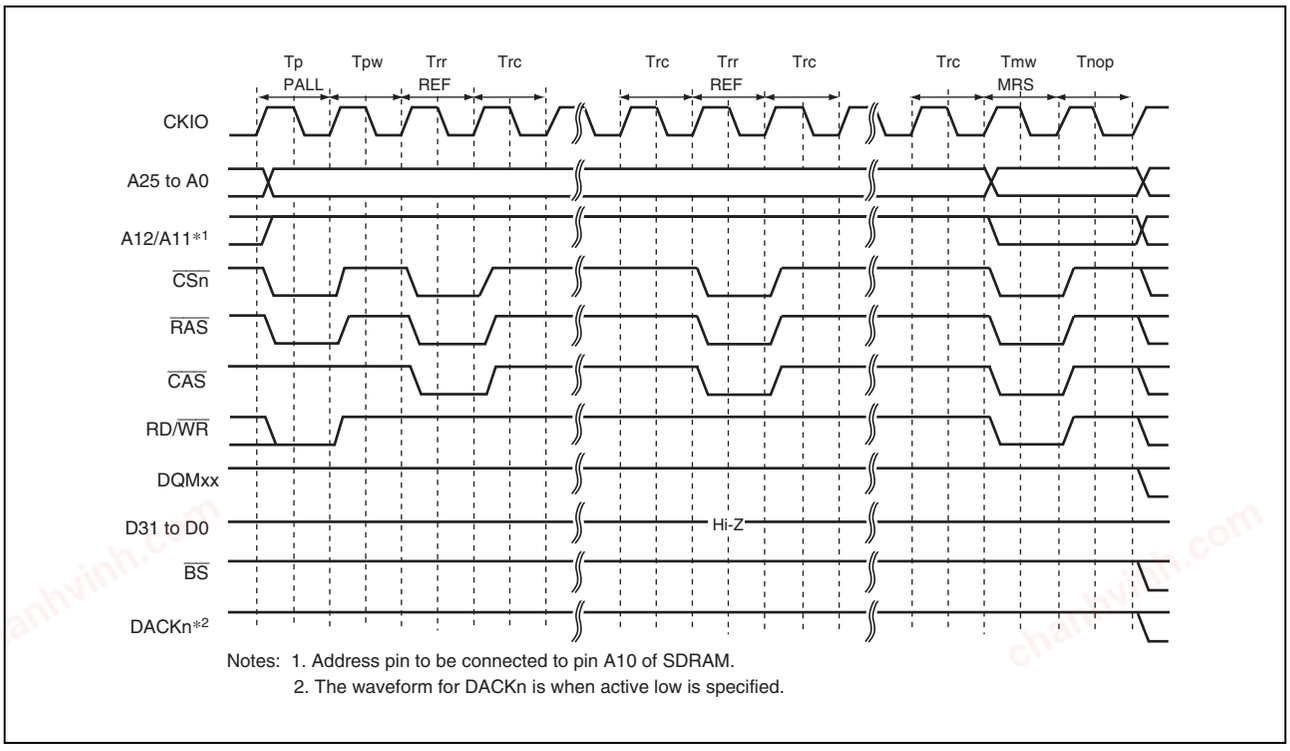


Figure 8.30 SDRAM Mode Write Timing (Based on JEDEC)

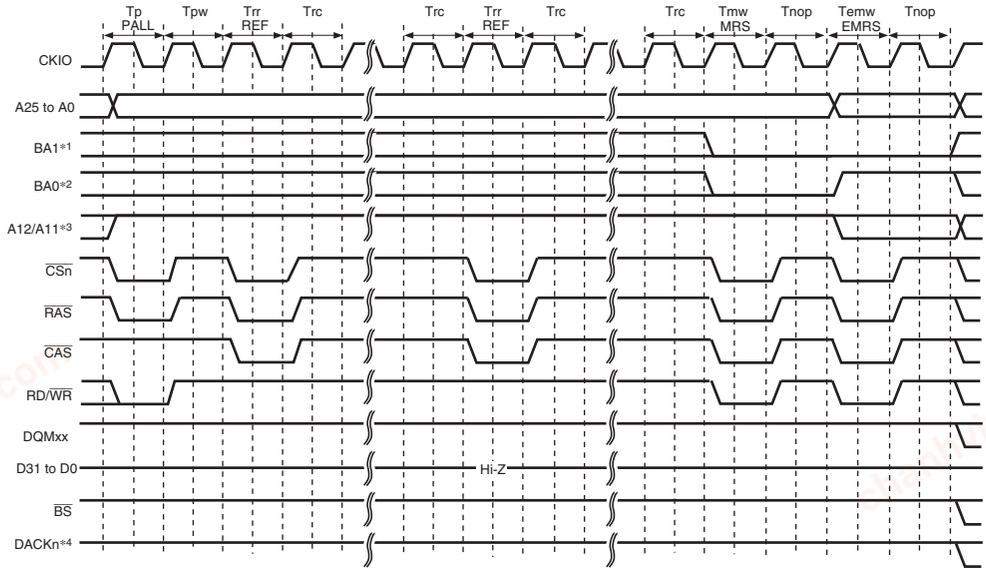
(12) Low-Power SDRAM

The low-power SDRAM can be accessed using the same protocol as the normal SDRAM.
The differences between the low-power SDRAM and normal SDRAM are that partial refresh takes place that puts only a part of the SDRAM in the self-refresh state during the self-refresh function, and that power consumption is low during refresh under user conditions such as the operating temperature. The partial refresh is effective in systems in which the data in a work area other than the specific area can be lost without severe repercussions. For details, refer to the Data Sheet for the low-power SDRAM to be used.
The low-power SDRAM supports the extension mode register in addition to the mode registers as the normal SDRAM. This LSI supports issuing of the extension mode register write command (EMRS).
The EMRS command is issued according to the conditions specified in table below. For example, if data H'0YYYYYYY is written to address H'3FFFEXX0 in 32 bits, the commands are issued to the CS3 space in the following sequence: PALL -> REF × 8 -> MRS -> EMRS. In this case, the MRS and EMRS issue addresses are H'0000XX0 and H'YYYYYYY, respectively. If data H'1YYYYYYY is written to address H'3FFFEXX0 in 32 bits, the commands are issued to the CS3 space in the following sequence: PALL -> MRS -> EMRS.

chanhvinh.com

Table 8.16 Output Addresses when EMRS Command Is Issued

Command to be Issued	Access Address	Access Data	Write Access Size	MRS Command Issue Address	EMRS Command Issue Address
CS2 MRS	H'3FFFDXX0	H'********	16 bits	H'0000XX0	—
CS3 MRS	H'3FFFEXX0	H'********	16 bits	H'0000XX0	—
CS2 MRS + EMRS (with refresh)	H'3FFFDXX0	H'0YYYYYYY	32 bits	H'0000XX0	H'YYYYYYY
CS3 MRS + EMRS (with refresh)	H'3FFFEXX0	H'0YYYYYYY	32 bits	H'0000XX0	H'YYYYYYY
CS2 MRS + EMRS (without refresh)	H'3FFFDXX0	H'1YYYYYYY	32 bits	H'0000XX0	H'YYYYYYY
CS3 MRS + EMRS (without refresh)	H'3FFFEXX0	H'1YYYYYYY	32 bits	H'0000XX0	H'YYYYYYY



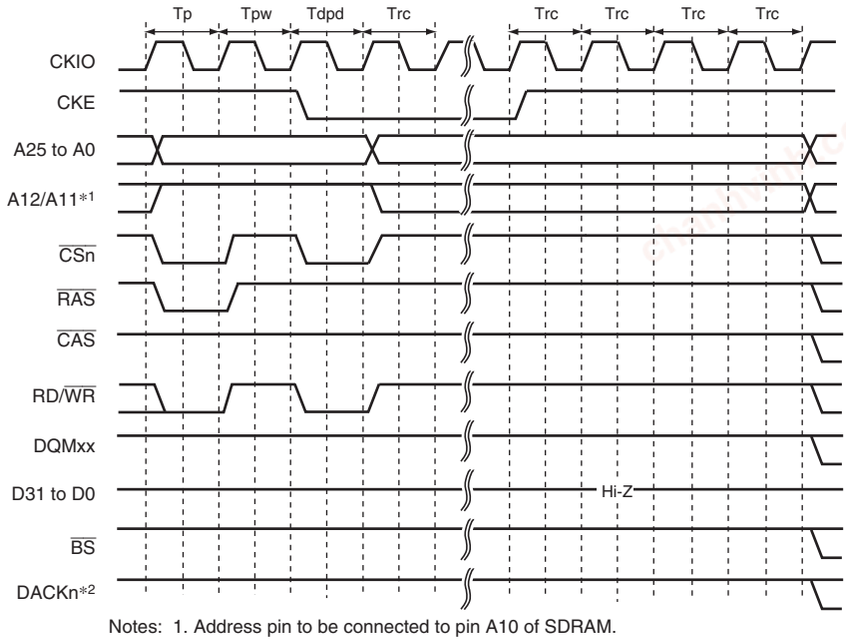
Notes: 1. Address pin to be connected to pin BA1 of SDRAM.
2. Address pin to be connected to pin BA0 of SDRAM.
3. Address pin to be connected to pin A10 of SDRAM.
4. The waveform for DACKn is when active low is specified.

Figure 8.31 EMRS Command Issue Timing

- Deep power-down mode
  The low-power SDRAM supports the deep power-down mode as a low-power consumption mode. In the partial self-refresh function, self-refresh is performed on a specific area. In the deep power-down mode, self-refresh will not be performed on any memory area. This mode is effective in systems where all of the system memory areas are used as work areas.

If the RMODE bit in the SDCR is set to 1 while the DEEP and RFSH bits in the SDCR are set to 1, the low-power SDRAM enters the deep power-down mode. If the RMODE bit is cleared to 0, the CKE signal is pulled high to cancel the deep power-down mode. Before executing an access after returning from the deep power-down mode, the power-up sequence must be re-executed.

chanhvinh.com



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.32 Deep Power-Down Mode Transition Timing

chanhvinh.com

8.5.7 Burst ROM (Clocked Asynchronous) Interface

The burst ROM (clocked asynchronous) interface is used to access a memory with a high-speed read function using a method of address switching called the burst mode or page mode. In a burst ROM (clocked asynchronous) interface, basically the same access as the normal space is performed, but the 2nd and subsequent access cycles are performed only by changing the address, without negating the $\overline{RD}$ signal at the end of the 1st cycle. In the 2nd and subsequent access cycles, addresses are changed at the falling edge of the CKIO.

For the 1st access cycle, the number of wait cycles specified by the W3 to W0 bits in CSnWCR is inserted. For the 2nd and subsequent access cycles, the number of wait cycles specified by the BW1 and BW0 bits in CSnWCR is inserted.

In the access to the burst ROM (clocked asynchronous), the $\overline{BS}$ signal is asserted only to the first access cycle. An external wait input is valid only to the first access cycle.

In the single access or write access that does not perform the burst operation in the burst ROM (clocked asynchronous) interface, access timing is same as a normal space.

Table 8.17 lists a relationship between bus width, access size, and the number of bursts. Figure 8.33 shows a timing chart.

Table 8.17 Relationship between Bus Width, Access Size, and Number of Bursts

Bus Width	Access Size	CSnWCR. BST[1:0] Bits	Number of Bursts	Access Count
8 bits	8 bits	Not affected	1	1
	16 bits	Not affected	2	1
	32 bits	Not affected	4	1
	16 bytes	00	16	1
		01	4	4
	32 bytes	00	16	2
		01	4	8
	64 bytes	00	16	4
		01	4	16
16 bits	8 bits	Not affected	1	1
	16 bits	Not affected	1	1
	32 bits	Not affected	2	1
	16 bytes	00	8	1
		01	2	4
		10*1	4	2
			2, 4, 2	3
	32 bytes	00	8	2
		01	2	8
		10*1	4	4
			2, 4, 2	6
	64 bytes	00	8	4
		01	2	16
		10*1	4	8
			2, 4, 2	12
32 bits	8 bits	Not affected	1	1
	16 bits	Not affected	1	1
	32 bits	Not affected	1	1
	16 bytes	Not affected	4	1
	32 bytes	Not affected	4	2
	64 bytes	Not affected	4	4

Note 1. When the bus width is 16 bits, the access size is 16 bytes or more, and the BST[1:0] bits in CSnWCR are 10, the number of bursts and access count depend on the access start address. At address H'xxx0 or H'xxx8, 4-4 burst access is performed. At address H'xxx4 or H'xxxC, 2-4-2 burst access is performed.

chanhvinh.com

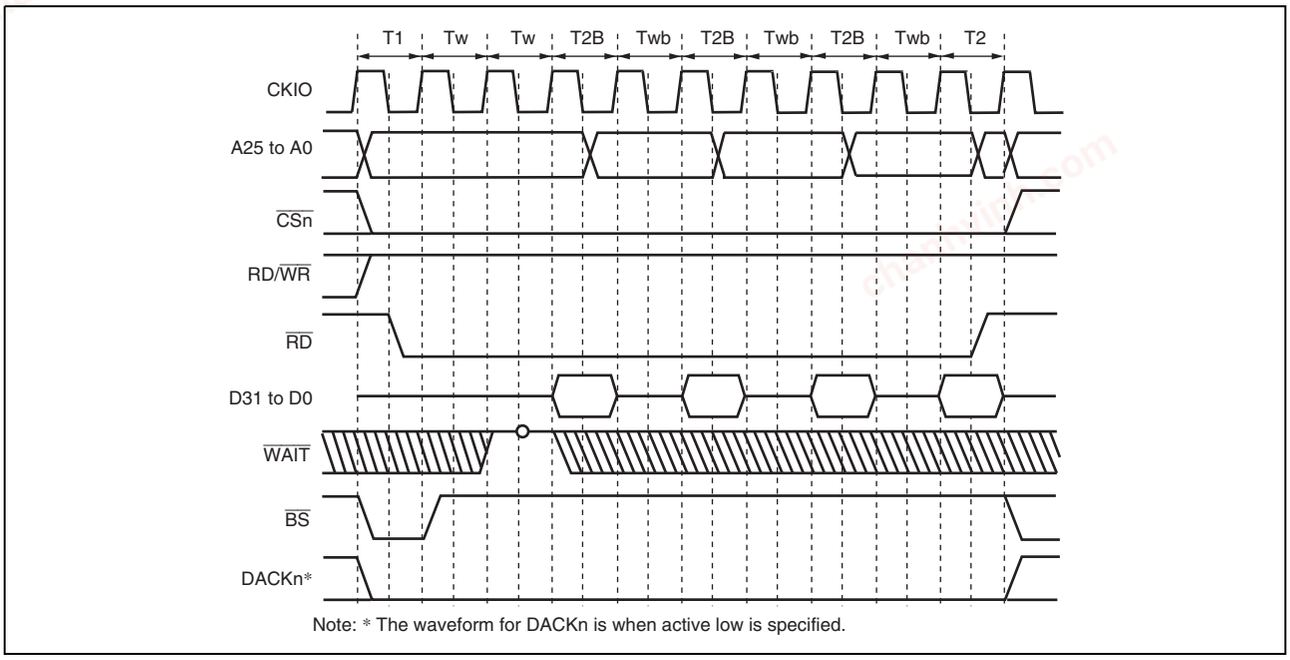


Figure 8.33 Burst ROM Access Timing (Clocked Asynchronous) (Bus Width = 32 Bits, 16-Byte Transfer (Number of Burst 4), Wait Cycles Inserted in First Access = 2, Wait Cycles Inserted in Second and Subsequent Access Cycles = 1)

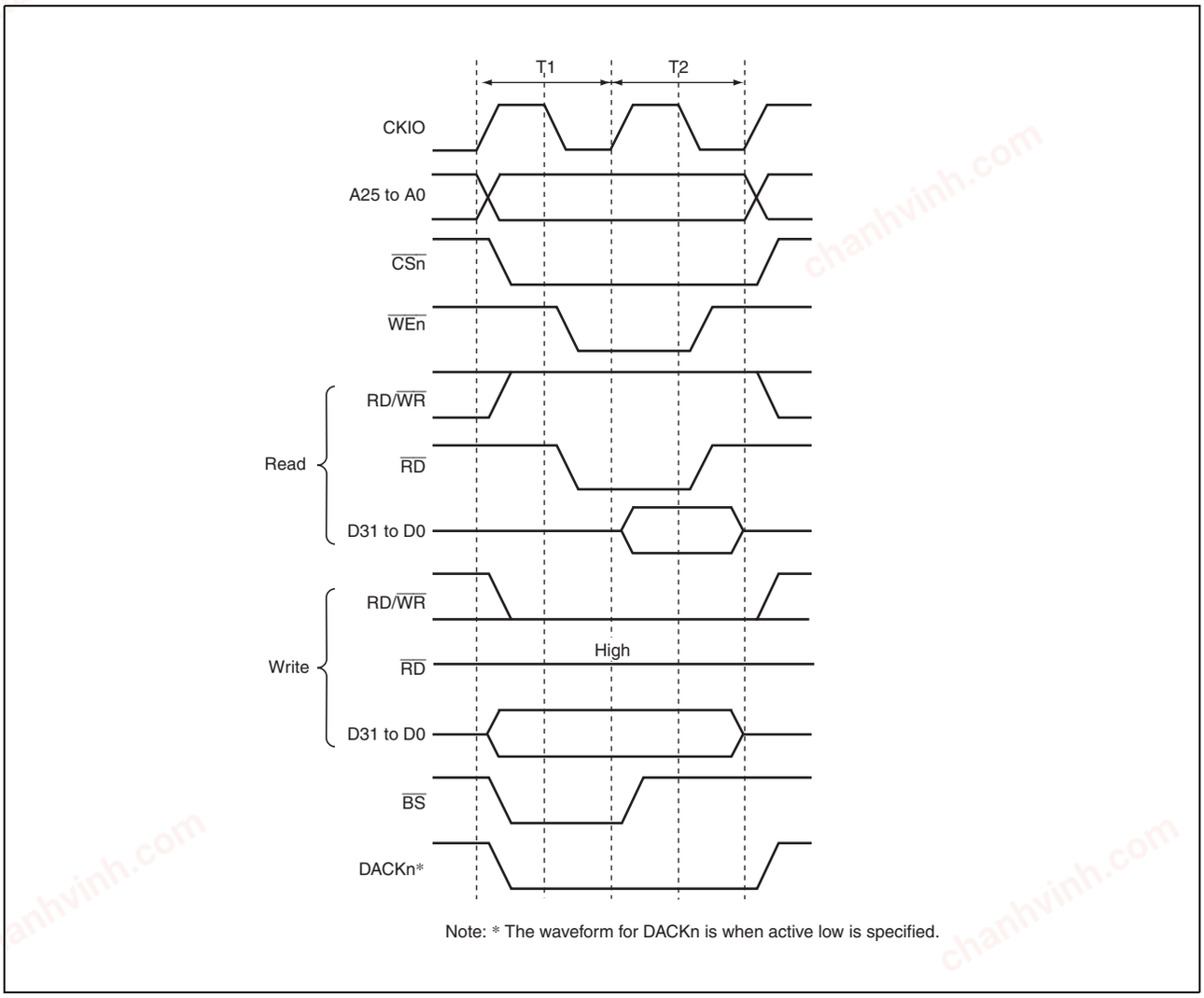
8.5.8 SRAM Interface with Byte Selection

The SRAM interface with byte selection is a memory interface that outputs the byte selection signal ($\overline{\text{WEn}}$) in both read and write bus cycles. This interface has 16-bit data pins and accesses SRAMs having upper and lower byte selection pins, such as $\overline{\text{UB}}$ and $\overline{\text{LB}}$.

When the BAS bit in CSnWCR is cleared to 0 (initial value), the write access timing of the SRAM interface with byte selection is the same as that for the normal space interface. While in read access of a byte-selection SRAM interface, the byte-selection signal is output from the $\overline{\text{WEn}}$ pin, which is different from that for the normal space interface. The basic access timing is shown in Figure 8.34. In write access, data is written to the memory according to the timing of the byte-selection pin ($\overline{\text{WEn}}$). For details, refer to the Data Sheet for the corresponding memory.

If the BAS bit in CSnWCR is set to 1, the $\overline{\text{WEn}}$ pin and RD/$\overline{\text{WR}}$ pin timings change. Figure 8.35 shows the basic access timing. In write access, data is written to the memory according to the timing of the write enable pin (RD/$\overline{\text{WR}}$). The data hold timing from RD/$\overline{\text{WR}}$ negation to data write must be acquired by setting the HW1 and HW0 bits in CSnWCR. Figure 8.36 shows the access timing when a software wait is specified.

chanhvinh.com



Note: * The waveform for DACKn is when active low is specified.

Figure 8.34 Basic Access Timing for SRAM with Byte Selection (BAS = 0)

chanhvinh.com

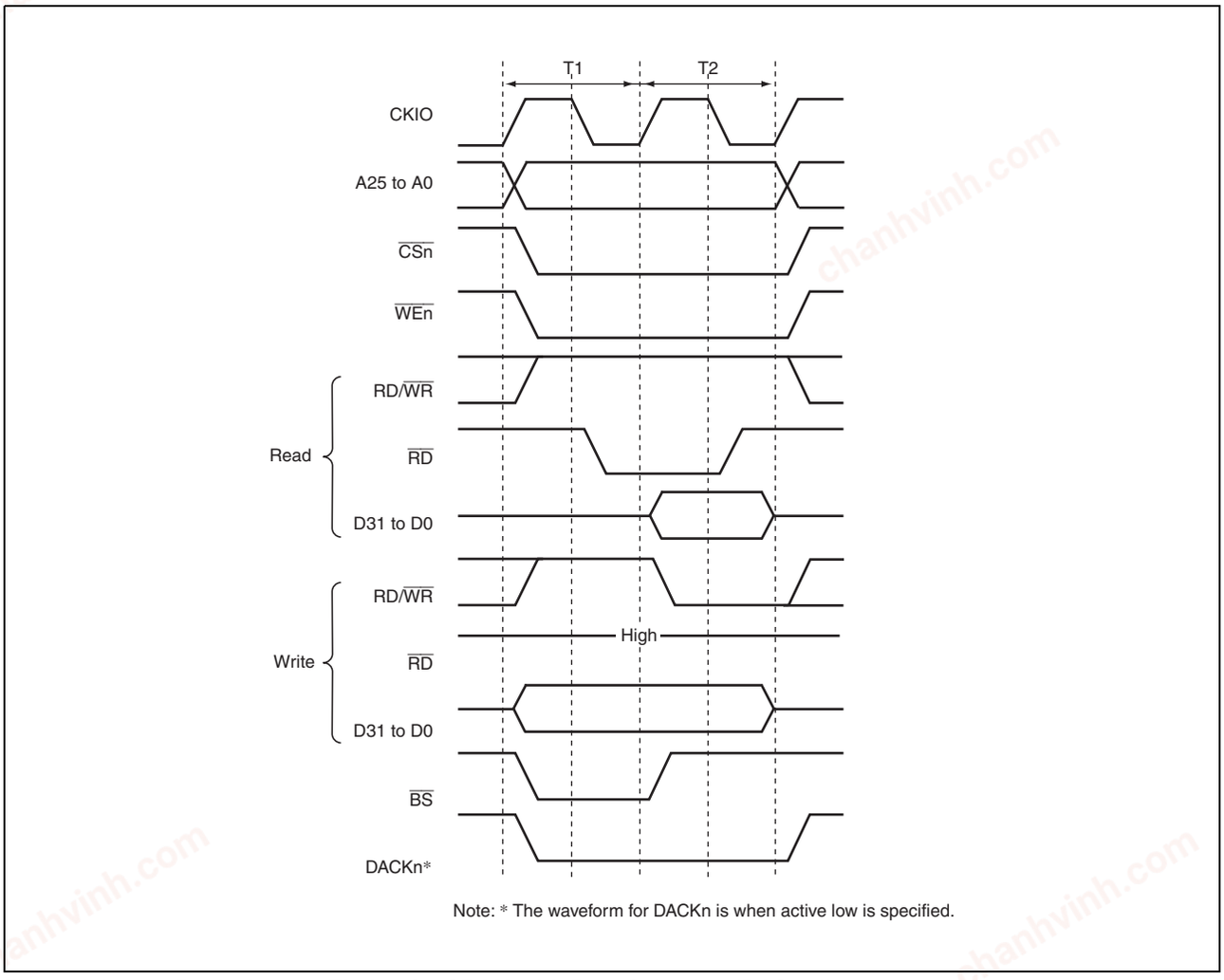


Figure 8.35 Basic Access Timing for SRAM with Byte Selection (BAS = 1)

chanhvinh.com

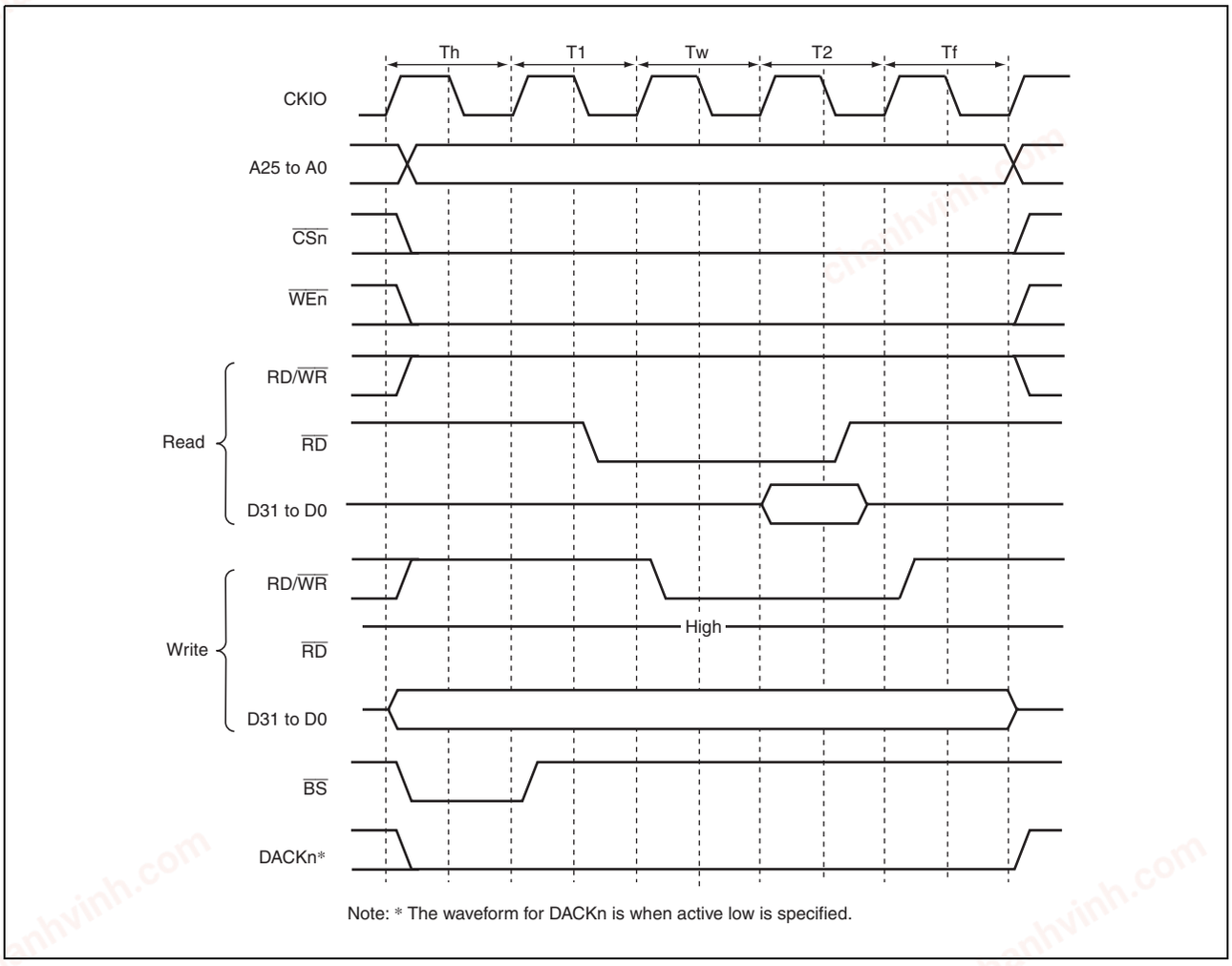


Figure 8.36 Wait Timing for SRAM with Byte Selection (BAS = 1) (SW[1:0] = 01, WR[3:0] = 0001, HW[1:0] = 01)

chanhvinh.com

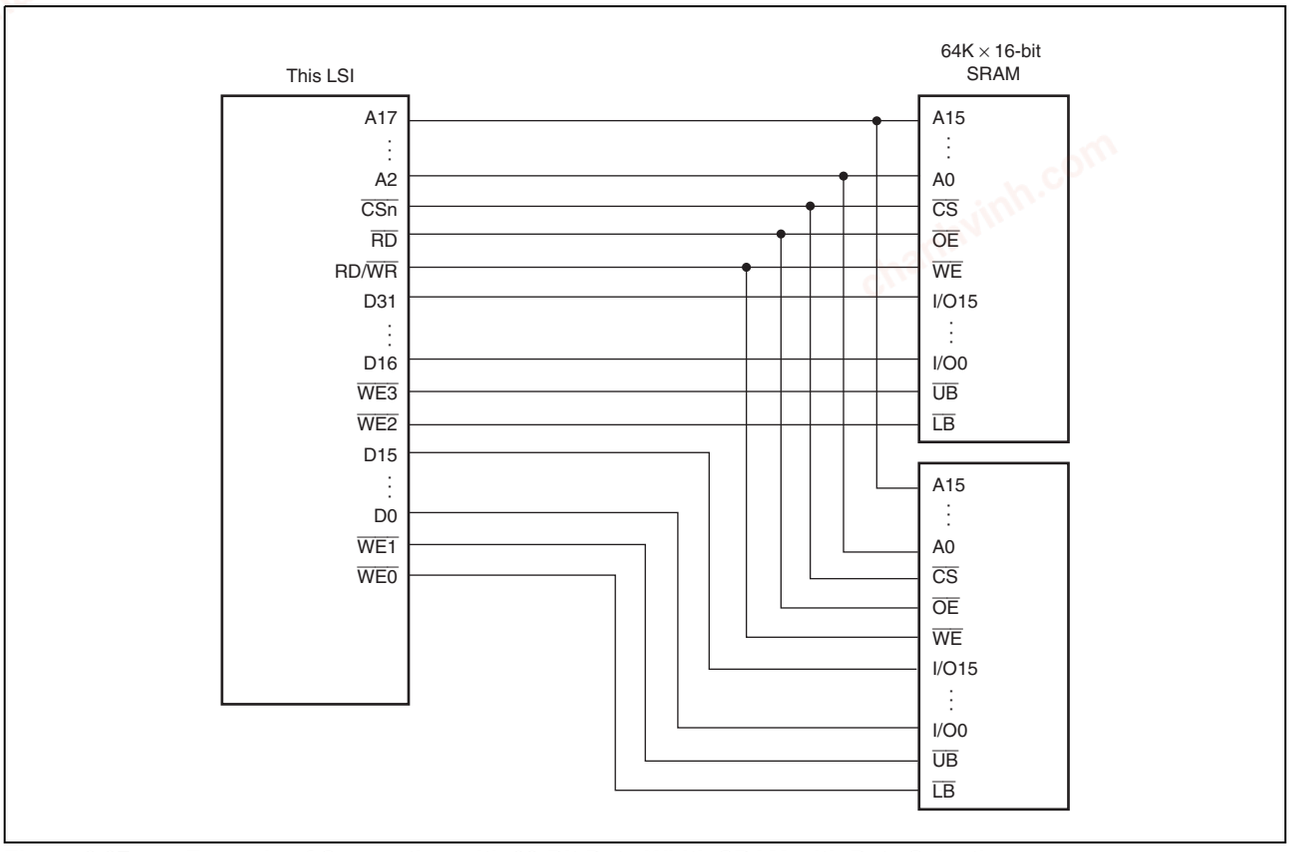


Figure 8.37 Example of Connection with 32-Bit Data-Width SRAM with Byte Selection

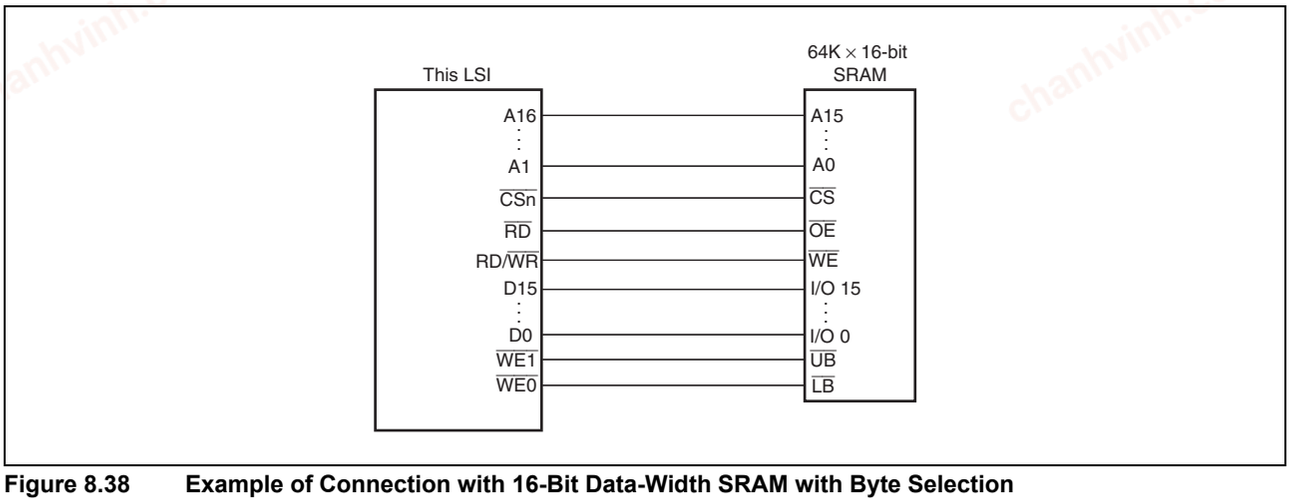


Figure 8.38 Example of Connection with 16-Bit Data-Width SRAM with Byte Selection

chanhvinh.com

8.5.9 Burst ROM (Clocked Synchronous) Interface

The burst ROM (clocked synchronous) interface is supported to access a ROM with a synchronous burst function at high speed. The burst ROM interface accesses the burst ROM in the same way as a normal space. This interface is valid only for area 0.

In the first access cycle, wait cycles are inserted. In this case, the number of wait cycles to be inserted is specified by the W3 to W0 bits in CS0WCR. In the second and subsequent cycles, the number of wait cycles to be inserted is specified by the BW1 and BW0 bits in CS0WCR.

While the burst ROM (clocked synchronous) is accessed, the $\overline{BS}$ signal is asserted only for the first access cycle and an external wait input is also valid for the first access cycle.

When the bus width is 16 bits, the burst length must be specified as 8. When the bus width is 32 bits, the burst length must be specified as 4. The burst ROM interface does not support the 8-bit bus width for the burst ROM.

The burst ROM interface performs burst operations for all read access. For example, in a 32-bit access over a 16-bit bus, valid 16-bit data is read two times and invalid 16-bit data is read six times. These invalid data read cycles increase the memory access time and degrade the program execution speed and DMA transfer speed. To prevent this problem, it is recommended using a read in a 16-byte or more access size. The burst ROM interface performs write access in the same way as normal space access.

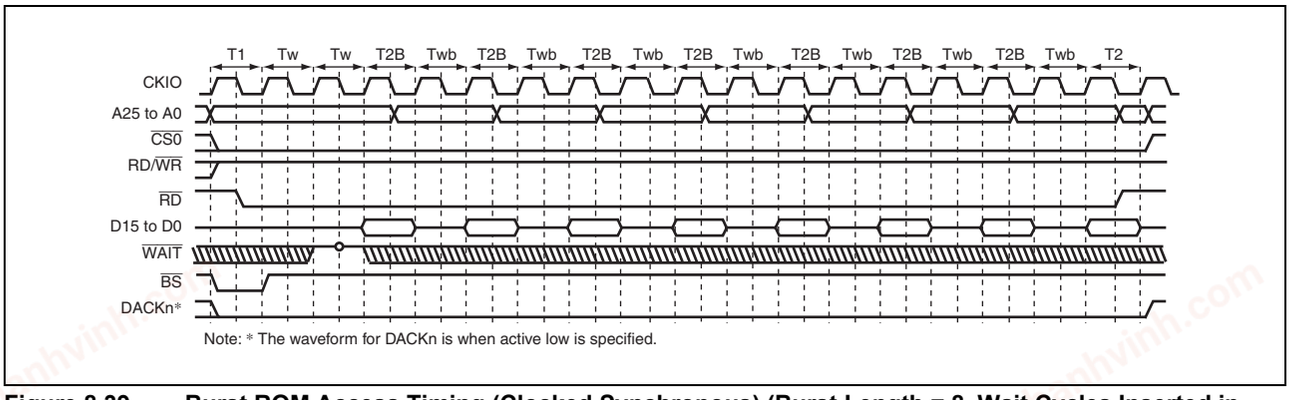


Figure 8.39 Burst ROM Access Timing (Clocked Synchronous) (Burst Length = 8, Wait Cycles Inserted in First Access = 2, Wait Cycles Inserted in Second and Subsequent Access Cycles = 1)

chanhvinh.com

8.5.10 Wait between Access Cycles

As the operating frequency of LSIs becomes higher, the off-operation of the data buffer often collides with the next data access when the read operation from devices with slow access speed is completed. As a result of these collisions, the reliability of the device is low and malfunctions may occur. A function that avoids data collisions by inserting idle (wait) cycles between continuous access cycles has been newly added.
The number of wait cycles between access cycles can be set by the WM bit in CSnWCR, bits IWW2 to IWW0, IWRWD2 to IWRWD0, IWRWS2 to IWRWS0, IWRRD2 to IWRRD0, and IWRRS2 to IWRRS 0 in CSnBCR. The conditions for setting the idle cycles between access cycles are shown below.

1. Continuous access cycles are write-read or write-write
2. Continuous access cycles are read-write for different spaces
3. Continuous access cycles are read-write for the same space
4. Continuous access cycles are read-read for different spaces
5. Continuous access cycles are read-read for the same space

For the specification of the number of idle cycles between access cycles described above, refer to the description of each register.
Besides the idle cycles between access cycles specified by the registers, idle cycles must be inserted to interface with the internal bus or to obtain the minimum pulse width for a multiplexed pin ($\overline{\text{WEn}}$). The following gives detailed information about the idle cycles and describes how to estimate the number of idle cycles.
The number of idle cycles on the external bus from $\overline{\text{CSn}}$ negation to $\overline{\text{CSn}}$ or $\overline{\text{CSm}}$ assertion is described below.
There are seven conditions that determine the number of idle cycles on the external bus as shown in Table 8.18. The effects of these conditions are shown in Figure 8.40.

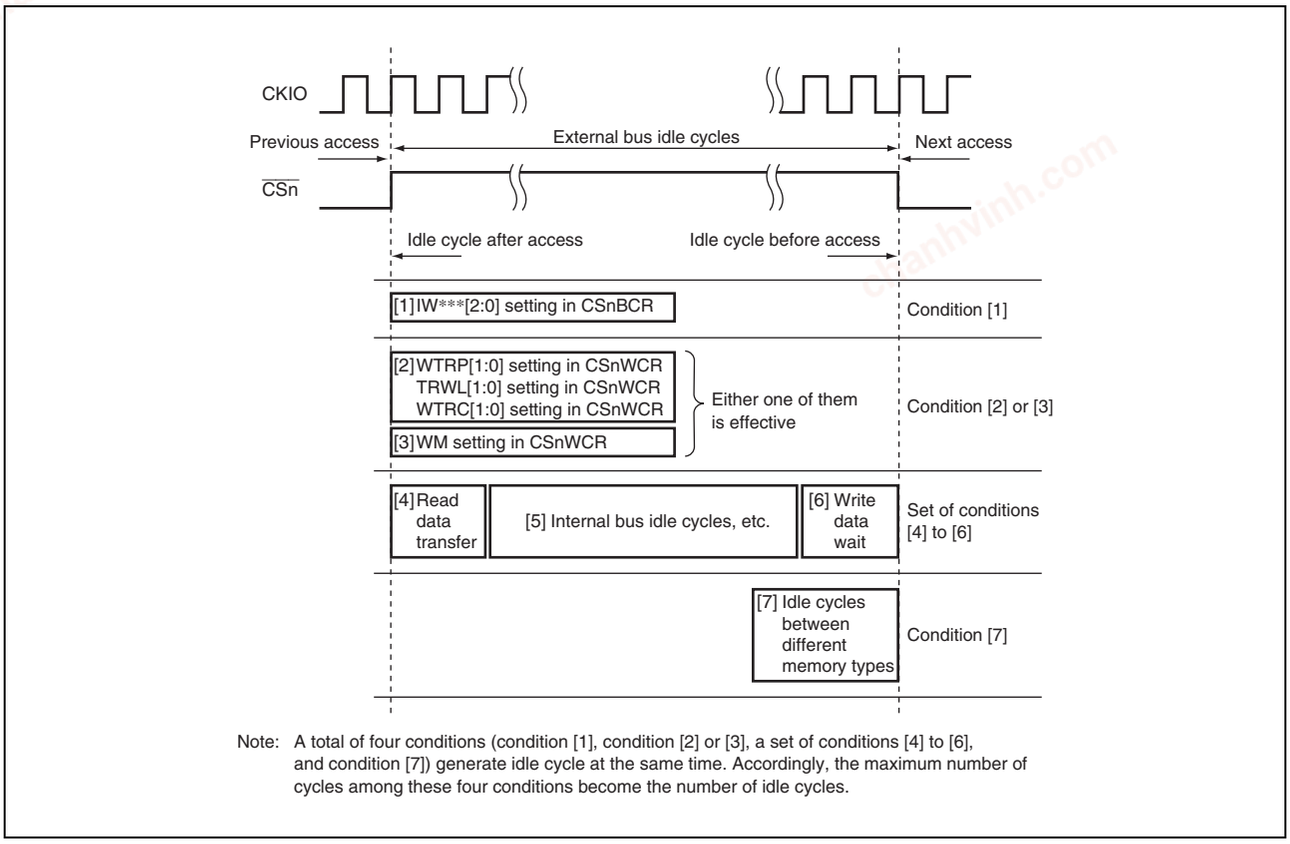
chanhvinh.com

Table 8.18 Conditions for Determining Number of Idle Cycles

No.	Condition	Description	Range	Note
[1]	IW***[2:0] in CSnBCR	These bits specify the number of idle cycles for access. The number of idle cycles can be specified independently for each combination of the previous and next cycles. For example, in the case where reading CS1 space followed by reading other CS space, the bits IWRRD[2:0] in CS1BCR should be set to B'100 to specify six or more idle cycles. This condition is effective only for access cycles other than single address transfer and generates idle cycles after the access is completed.	0 to 12	Do not set 0 for the number of idle cycles between memory types which are not allowed to be accessed successively.
[2]	SDRAM-related bits in CSnWCR	These bits specify precharge completion and startup wait cycles and idle cycles between commands for SDRAM access. This condition is effective only for SDRAM access and generates idle cycles after the access is completed	0 to 3	Specify these bits in accordance with the specification of the target SDRAM.
[3]	WM in CSnWCR	This bit enables or disables external $\overline{\text{WAIT}}$ pin input for the memory types other than SDRAM. When this bit is cleared to 0 (external $\overline{\text{WAIT}}$ enabled), one idle cycle is inserted to check the external $\overline{\text{WAIT}}$ pin input after the access is completed. When this bit is set to 1 (disabled), no idle cycle is generated.	0 or 1	
[4]	Read data transfer cycle	One idle cycle is inserted after a read access is completed. This idle cycle is not generated for the first or middle cycles in divided access cycles. This is neither generated when the HW[1:0] bits in CSnWCR are not B'00.	0 or 1	One idle cycle is always generated after a read cycle with SDRAM.
[5]	Internal bus idle cycles, etc.	External bus access requests from the CPU or the direct memory access controller and their results are passed through the internal bus. The external bus enters idle state during internal bus idle cycles or while a bus other than the external bus is being accessed. This condition is not effective for divided access cycles, which are generated by the bus state controller when the access size is larger than the external data bus width.	0 or larger	The number of internal bus idle cycles may not become 0 depending on the CPU: internal bus: CKIO
[6]	Write data wait cycles	During write access, a write cycle is executed on the external bus only after the write data becomes ready. This write data wait period generates idle cycles before the write cycle. Note that when the previous cycle is a write cycle and the internal bus idle cycles are shorter than the previous write cycle, write data can be prepared in parallel with the previous write cycle and therefore, no idle cycle is generated (write buffer effect).	0 or 1	For write → write or write → read access cycles, successive access cycles without idle cycles may be available due to the write buffer effect described in the left column. If successive access cycles without idle cycles are not allowed, specify the minimum number of idle cycles between access cycles through CSnBCR.
[7]	Idle cycles between different memory types	To ensure the minimum pulse width on the signal-multiplexed pins, idle cycles may be inserted before access after memory types are switched. For some memory types, idle cycles are inserted even when memory types are not switched.	0 to 2	The number of idle cycles depends on the target memory types. See Table 8.19.

In the above conditions, a total of four conditions, that is, condition [1], condition [2] or [3] (either one is effective), a set of conditions [4] to [6] (these are generated successively, and therefore the sum of them should be taken as one set of idle cycles), and condition [7] are generated at the same time. The maximum number of idle cycles among these four conditions become the number of idle cycles on the external bus. To ensure the minimum idle cycles, be sure to make register settings for condition [1].

chanhvinh.com



Note: A total of four conditions (condition [1], condition [2] or [3], a set of conditions [4] to [6], and condition [7]) generate idle cycle at the same time. Accordingly, the maximum number of cycles among these four conditions become the number of idle cycles.

Figure 8.40 Idle Cycle Conditions

Table 8.19 Number of Idle Cycles Inserted between Access Cycles to Different Memory Types

	Next Cycle						
Previous Cycle	**SRAM**	**Burst ROM (Asynchronous)**	**MPX-I/O**	**Byte SRAM (BAS = 0)**	**Byte SRAM (BAS = 1)**	**SDRAM**	**Burst ROM (Synchronous)**
SRAM	0	0	1	0	0/1*	0/1*	0
Burst ROM (asynchronous)	0	0	1	0	0/1*	0/1*	0
MPX-I/O	1	1	0	1	1	1	1
Byte SRAM (BAS = 0)	0	0	1	0	0/1*	0/1*	0
Byte SRAM (BAS = 1)	0/1*	0/1*	1/2*	0/1*	0	0	0/1*
SDRAM	1	1	2	1	0	0	1
Burst ROM (synchronous)	0	0	1	0	1	1	0

Note: * The number of idle cycles is determined by the setting of bits HW[1:0] in the CSnWCR register for the previous cycle. The values on the left and right sides of the virgules show the numbers of idle cycles when HW[1:0] ≠ B'00 and HW[1:0] = B'00, respectively. If the memory connected to the CSn space in the previous cycle is of a type for which bits HW[1:0] in the CSnWCR register are ineffective, the number of idle cycles will be the value on the right side.

chanhvinh.com

8.5.11 Others

(1) Reset

This module can be initialized completely only at power-on reset. At power-on reset, all signals are negated and data output buffers are turned off regardless of the bus cycle state after the internal reset is synchronized with the internal clock. All control registers are initialized. In software standby and sleep, control registers of the bus state controller are not initialized.

(2) Caution on Write Buffer

Since the bus state controller incorporates a one-stage write buffer, it can execute an access via the internal bus before the previous external bus cycle is completed in a write cycle. If the on-chip module is read or written after the external low-speed memory is written, the on-chip module can be accessed before the completion of the external low-speed memory write cycle.

In read cycles, the CPU is placed in the wait state until read operation has been completed. To continue the process after the data write to the device has been completed, perform a dummy read to the same address to check for completion of the write before the next process to be executed.

The write buffer of the bus state controller functions in the same way for an access by a bus master other than the CPU such as the direct memory access controller. Accordingly, to perform DMA transfers, the next read cycle is initiated before the previous write cycle is completed. Note, however, that if both the DMA source and destination addresses exist in external memory space, the next read cycle will not be initiated until the previous write cycle is completed.

Changing the registers in this module while the write buffer is operating may disrupt correct write access. Therefore, do not change the registers in this module immediately after a write access. If this change becomes necessary, do it after executing a dummy read of the write data.

(3) On-Chip Peripheral Module Access

To access an on-chip module register, two or more peripheral module clock (P0φ or P1φ) cycles are required.

When the CPU writes data to the internal peripheral registers, the CPU performs the succeeding instructions without waiting for the completion of writing to registers.

For example, a case is described here in which the system is transferring to the software standby mode for power savings. To make this transition, the WFI instruction must be performed after setting the STBY bit in the STBCR1 register to 1. However a dummy read of the STBCR1 register is required before executing the WFI instruction. If a dummy read is omitted, the CPU executes the WFI instruction before the STBY bit is set to 1, thus the system enters sleep mode not software standby mode. A dummy read of the STBCR1 register is indispensable to complete writing to the STBY bit.

To reflect the change by internal peripheral registers while performing the succeeding instructions, execute a dummy read of registers to which write instruction is given and then perform the succeeding instructions.

chanhvinh.com

9. Direct Memory Access Controller

The direct memory access controller can be used in place of the CPU to perform high-speed transfers between external devices that have DACK (transfer request acknowledge signal), external memory, on-chip memory, memory-mapped external devices, and on-chip peripheral modules.

9.1 Features

- Number of channels selectable: 16 channels (CH0 to CH15). Only the CH0 channel can receive external requests.
- 4-Gbyte address space (according to the architecture)
- Transfer data size: Byte, two bytes, four bytes, eight bytes, 16 bytes, 32 bytes, 64 bytes, and 128 bytes
- Maximum transfer count: 2^{32} - 1 bytes
- Address mode: Dual address mode
- Transfer requests: Can be selected from the three types of external request, on-chip peripheral module request, and auto request (software trigger)
  The following modules can issue on-chip peripheral module requests.
  —Serial communication interface with FIFO: 10 sources
  —A/D converter: 1 source
  —Multi-function timer pulse unit 2: 5 sources
  —USB2.0 host/function module: 4 sources
  —Serial sound interface: 7 sources
  —Renesas SPDIF interface: 2 sources
  —CD-ROM decoder: 1 source (RZ/A1L only)
  —SD host interface: 4 sources
  —MMC host interface: 2 sources
  —Renesas serial peripheral interface: 6 sources
  —IEBus™ controller: 2 sources (RZ/A1L only)
  —OS timer: 2 sources
  —SCUX: 8 sources
  —Media local bus: 1 source (RZ/A1L only)
  —Serial communication interface: 4 sources
  —I^2C bus interface: 8 sources
  —LIN interface: 2 sources (RZ/A1L only)
- Transfer mode: Single transfer mode and block transfer mode are selectable.
- Priority: The channel priority levels within channels 0 to 7 and within channels 8 to 15 are selectable between fixed mode and round-robin mode (the channel priority level between the group of channels 0 to 7 and the group of channels 8 to 15 is round-robin mode).
- Interrupt request: An interrupt request can be sent to the CPU on completion of data transfer (DMA transfer end interrupt per channel) or on occurrence of a transfer error (DMA error interrupt).
- External request detection: Low level detection, high level detection, rising edge detection, and falling edge detection are selectable for DREQ input detection.
- The DMA registers have a continuous execution function that allows the next DMA transfer to be executed continuously by making settings for the next DMA transfer during execution of the current DMA transfer. This continuous execution function can be enabled or disabled independently in each channel.
- Link mode: In this mode, the setting data (descriptor data) located in the memory by the CPU is automatically retrieved by the DMAC, and DMA transfer is performed according to those values.
- Buffer sweep: If an ongoing DMA transfer is forced to end, the data already retrieved into the buffer can be output before DMA transfer ends.

chanhvinh.com

- Interval: A specific DMA transfer interval can be specified to adjust the bus occupancy.

9.2 Input/Output Pins

Table 9.1 lists the pin configuration. This module has pins for a single channel (CH0) as the external bus use.

Table 9.1 Pin Configuration

Channel	Name	Pin Name	I/O	Function
0	DMA transfer request	DREQ0	Input	DMA transfer request input from an external device to channel 0
	DMA transfer request acknowledge	DACK0	Output	DMA transfer request acknowledge output from channel 0 of this module
	DMA transfer end	TEND0	Output	DMA transfer end output for channel 0 of this module

Note 1. For the active level of DACK0 and TEND0, refer to section 8, Bus State Controller.

9.3 Register Configuration

The register configuration is shown in the figure below.

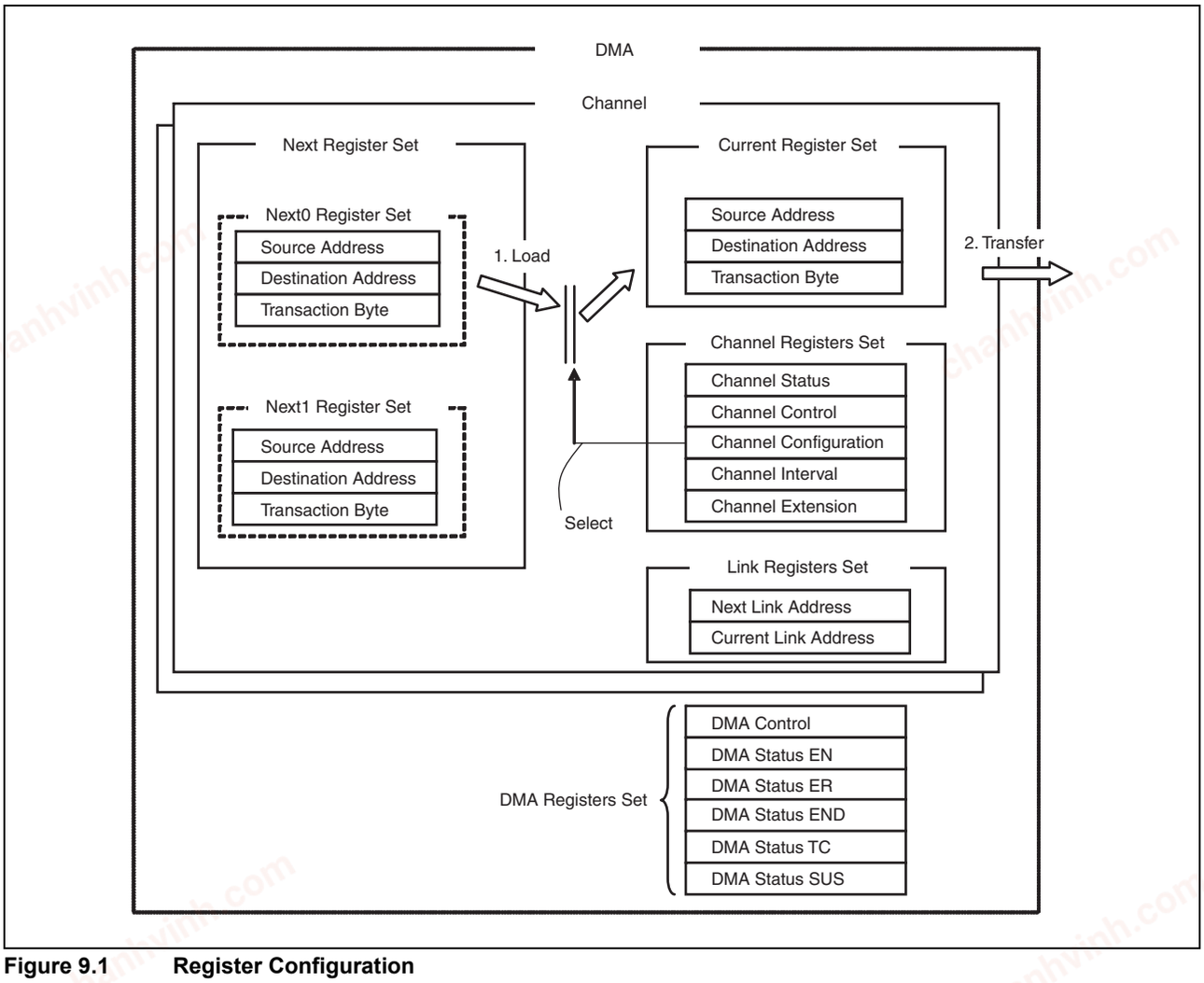


Figure 9.1 Register Configuration

(a) Next Register Set

This register set is used to set the source address, destination address, and transfer byte count of the DMA transaction to be executed next.

chanhvinh.com

It consists of the Next0 Register Set and the Next1 Register Set.
In register mode, set this register set by using software. In link mode, the descriptor read data is automatically set in the Next0 Register Set.
These register set values are loaded to the Current Register Set and used for DMA transfer.

(b) Current Register Set

This register set indicates the source address, destination address, and transfer byte count of the currently executed DMA transaction.
The values are loaded from the Next0/1 Register Set (register mode) or from the descriptor read data (link mode). The user cannot write directly to this register set.
The register set is automatically updated each time a DMA transaction is executed.

(c) Channel Register Set

This register set is used to make the DMA transfer settings.
The settings to be made with this register set include channel status indication, channel control, DMA transaction setting, and DMA transaction interval.

(d) Link Register Set

This register set consists of a register that sets the address of the descriptor to be loaded next in link mode (Next Link Address Register) and a register that indicates the address of the currently executed descriptor (Current Link Address Register).
The Current Link Address Register is automatically updated when a descriptor is read. The user cannot write directly to this register set.

(e) DMA Register Set

This register set consists of a register that controls DMA as a whole and registers that indicate the status of the corresponding channels. It enables channel priority control as well as the monitoring of the channel status (EN, ER, END, TC, and SUS).

(f) Extended Resource Selector Register Set

This register set is used to select the on-chip peripheral module to perform DMA transfer and the external request.

chanhvinh.com

9.4 Register Descriptions

Table 9.2 lists the register configuration. There are eleven control registers and five status registers for each channel, and twelve common control registers are used by all channels. In addition, there is one extension resource selector per two channels. Each channel number is expressed in the register names, as in N0SA_0 for N0SA in channel 0.

Table 9.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	Next0 source address register 0	N0SA_0	RW	H'00000000	H'E8200000	32
	Next0 destination address register 0	N0DA_0	RW	H'00000000	H'E8200004	32
	Next0 transaction byte register 0	N0TB_0	RW	H'00000000	H'E8200008	32
	Next1 source address register 0	N1SA_0	RW	H'00000000	H'E820000C	32
	Next1 destination address register 0	N1DA_0	RW	H'00000000	H'E8200010	32
	Next1 transaction byte register 0	N1TB_0	RW	H'00000000	H'E8200014	32
	Current source address register 0	CRSA_0	R	H'00000000	H'E8200018	32
	Current destination address register 0	CRDA_0	R	H'00000000	H'E820001C	32
	Current transaction byte register 0	CRTB_0	R	H'00000000	H'E8200020	32
	Channel status register 0	CHSTAT_0	R	H'00000000	H'E8200024	32
	Channel control register 0	CHCTRL_0	RW	H'00000000	H'E8200028	32
	Channel configuration register 0	CHCFG_0	RW	H'00000000	H'E820002C	32
	Channel interval register 0	CHITVL_0	RW	H'00000000	H'E8200030	32
	Channel extension register 0	CHEXT_0	RW	H'00000000	H'E8200034	32
	Next link address register 0	NXLA_0	RW	H'00000000	H'E8200038	32
	Current link address register 0	CRLA_0	R	H'00000000	H'E820003C	32
1	Next0 source address register 1	N0SA_1	RW	H'00000000	H'E8200040	32
	Next0 destination address register 1	N0DA_1	RW	H'00000000	H'E8200044	32
	Next0 transaction byte register 1	N0TB_1	RW	H'00000000	H'E8200048	32
	Next1 source address register 1	N1SA_1	RW	H'00000000	H'E820004C	32
	Next1 destination address register 1	N1DA_1	RW	H'00000000	H'E8200050	32
	Next1 transaction byte register 1	N1TB_1	RW	H'00000000	H'E8200054	32
	Current source address register 1	CRSA_1	R	H'00000000	H'E8200058	32
	Current destination address register 1	CRDA_1	R	H'00000000	H'E820005C	32
	Current transaction byte register 1	CRTB_1	R	H'00000000	H'E8200060	32
	Channel status register 1	CHSTAT_1	R	H'00000000	H'E8200064	32
	Channel control register 1	CHCTRL_1	RW	H'00000000	H'E8200068	32
	Channel configuration register 1	CHCFG_1	RW	H'00000000	H'E820006C	32
	Channel interval register 1	CHITVL_1	RW	H'00000000	H'E8200070	32
	Channel extension register 1	CHEXT_1	RW	H'00000000	H'E8200074	32
	Next link address register 1	NXLA_1	RW	H'00000000	H'E8200078	32
	Current link address register 1	CRLA_1	R	H'00000000	H'E820007C	32

chanhvinh.com

Table 9.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
2	Next0 source address register 2	N0SA_2	RW	H'00000000	H'E8200080	32
	Next0 destination address register 2	N0DA_2	RW	H'00000000	H'E8200084	32
	Next0 transaction byte register 2	N0TB_2	RW	H'00000000	H'E8200088	32
	Next1 source address register 2	N1SA_2	RW	H'00000000	H'E820008C	32
	Next1 destination address register 2	N1DA_2	RW	H'00000000	H'E8200090	32
	Next1 transaction byte register 2	N1TB_2	RW	H'00000000	H'E8200094	32
	Current source address register 2	CRSA_2	R	H'00000000	H'E8200098	32
	Current destination address register 2	CRDA_2	R	H'00000000	H'E820009C	32
	Current transaction byte register 2	CRTB_2	R	H'00000000	H'E82000A0	32
	Channel status register 2	CHSTAT_2	R	H'00000000	H'E82000A4	32
	Channel control register 2	CHCTRL_2	RW	H'00000000	H'E82000A8	32
	Channel configuration register 2	CHCFG_2	RW	H'00000000	H'E82000AC	32
	Channel interval register 2	CHITVL_2	RW	H'00000000	H'E82000B0	32
	Channel extension register 2	CHEXT_2	RW	H'00000000	H'E82000B4	32
	Next link address register 2	NXLA_2	RW	H'00000000	H'E82000B8	32
	Current link address register 2	CRLA_2	R	H'00000000	H'E82000BC	32
3	Next0 source address register 3	N0SA_3	RW	H'00000000	H'E82000C0	32
	Next0 destination address register 3	N0DA_3	RW	H'00000000	H'E82000C4	32
	Next0 transaction byte register 3	N0TB_3	RW	H'00000000	H'E82000C8	32
	Next1 source address register 3	N1SA_3	RW	H'00000000	H'E82000CC	32
	Next1 destination address register 3	N1DA_3	RW	H'00000000	H'E82000D0	32
	Next1 transaction byte register 3	N1TB_3	RW	H'00000000	H'E82000D4	32
	Current source address register 3	CRSA_3	R	H'00000000	H'E82000D8	32
	Current destination address register 3	CRDA_3	R	H'00000000	H'E82000DC	32
	Current transaction byte register 3	CRTB_3	R	H'00000000	H'E82000E0	32
	Channel status register 3	CHSTAT_3	R	H'00000000	H'E82000E4	32
	Channel control register 3	CHCTRL_3	RW	H'00000000	H'E82000E8	32
	Channel configuration register 3	CHCFG_3	RW	H'00000000	H'E82000EC	32
	Channel interval register 3	CHITVL_3	RW	H'00000000	H'E82000F0	32
	Channel extension register 3	CHEXT_3	RW	H'00000000	H'E82000F4	32
	Next link address register 3	NXLA_3	RW	H'00000000	H'E82000F8	32
	Current link address register 3	CRLA_3	R	H'00000000	H'E82000FC	32

chanhvinh.com

Table 9.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
4	Next0 source address register 4	N0SA_4	RW	H'00000000	H'E8200100	32
	Next0 destination address register 4	N0DA_4	RW	H'00000000	H'E8200104	32
	Next0 transaction byte register 4	N0TB_4	RW	H'00000000	H'E8200108	32
	Next1 source address register 4	N1SA_4	RW	H'00000000	H'E820010C	32
	Next1 destination address register 4	N1DA_4	RW	H'00000000	H'E8200110	32
	Next1 transaction byte register 4	N1TB_4	RW	H'00000000	H'E8200114	32
	Current source address register 4	CRSA_4	R	H'00000000	H'E8200118	32
	Current destination address register 4	CRDA_4	R	H'00000000	H'E820011C	32
	Current transaction byte register 4	CRTB_4	R	H'00000000	H'E8200120	32
	Channel status register 4	CHSTAT_4	R	H'00000000	H'E8200124	32
	Channel control register 4	CHCTRL_4	RW	H'00000000	H'E8200128	32
	Channel configuration register 4	CHCFG_4	RW	H'00000000	H'E820012C	32
	Channel interval register 4	CHITVL_4	RW	H'00000000	H'E8200130	32
	Channel extension register 4	CHEXT_4	RW	H'00000000	H'E8200134	32
	Next link address register 4	NXLA_4	RW	H'00000000	H'E8200138	32
	Current link address register 4	CRLA_4	R	H'00000000	H'E820013C	32
5	Next0 source address register 5	N0SA_5	RW	H'00000000	H'E8200140	32
	Next0 destination address register 5	N0DA_5	RW	H'00000000	H'E8200144	32
	Next0 transaction byte register 5	N0TB_5	RW	H'00000000	H'E8200148	32
	Next1 source address register 5	N1SA_5	RW	H'00000000	H'E820014C	32
	Next1 destination address register 5	N1DA_5	RW	H'00000000	H'E8200150	32
	Next1 transaction byte register 5	N1TB_5	RW	H'00000000	H'E8200154	32
	Current source address register 5	CRSA_5	R	H'00000000	H'E8200158	32
	Current destination address register 5	CRDA_5	R	H'00000000	H'E820015C	32
	Current transaction byte register 5	CRTB_5	R	H'00000000	H'E8200160	32
	Channel status register 5	CHSTAT_5	R	H'00000000	H'E8200164	32
	Channel control register 5	CHCTRL_5	RW	H'00000000	H'E8200168	32
	Channel configuration register 5	CHCFG_5	RW	H'00000000	H'E820016C	32
	Channel interval register 5	CHITVL_5	RW	H'00000000	H'E8200170	32
	Channel extension register 5	CHEXT_5	RW	H'00000000	H'E8200174	32
	Next link address register 5	NXLA_5	RW	H'00000000	H'E8200178	32
	Current link address register 5	CRLA_5	R	H'00000000	H'E820017C	32

chanhvinh.com

Table 9.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
6	Next0 source address register 6	N0SA_6	RW	H'00000000	H'E8200180	32
	Next0 destination address register 6	N0DA_6	RW	H'00000000	H'E8200184	32
	Next0 transaction byte register 6	N0TB_6	RW	H'00000000	H'E8200188	32
	Next1 source address register 6	N1SA_6	RW	H'00000000	H'E820018C	32
	Next1 destination address register 6	N1DA_6	RW	H'00000000	H'E8200190	32
	Next1 transaction byte register 6	N1TB_6	RW	H'00000000	H'E8200194	32
	Current source address register 6	CRSA_6	R	H'00000000	H'E8200198	32
	Current destination address register 6	CRDA_6	R	H'00000000	H'E820019C	32
	Current transaction byte register 6	CRTB_6	R	H'00000000	H'E82001A0	32
	Channel status register 6	CHSTAT_6	R	H'00000000	H'E82001A4	32
	Channel control register 6	CHCTRL_6	RW	H'00000000	H'E82001A8	32
	Channel configuration register 6	CHCFG_6	RW	H'00000000	H'E82001AC	32
	Channel interval register 6	CHITVL_6	RW	H'00000000	H'E82001B0	32
	Channel extension register 6	CHEXT_6	RW	H'00000000	H'E82001B4	32
	Next link address register 6	NXLA_6	RW	H'00000000	H'E82001B8	32
	Current link address register 6	CRLA_6	R	H'00000000	H'E82001BC	32
7	Next0 source address register 7	N0SA_7	RW	H'00000000	H'E82001C0	32
	Next0 destination address register 7	N0DA_7	RW	H'00000000	H'E82001C4	32
	Next0 transaction byte register 7	N0TB_7	RW	H'00000000	H'E82001C8	32
	Next1 source address register 7	N1SA_7	RW	H'00000000	H'E82001CC	32
	Next1 destination address register 7	N1DA_7	RW	H'00000000	H'E82001D0	32
	Next1 transaction byte register 7	N1TB_7	RW	H'00000000	H'E82001D4	32
	Current source address register 7	CRSA_7	R	H'00000000	H'E82001D8	32
	Current destination address register 7	CRDA_7	R	H'00000000	H'E82001DC	32
	Current transaction byte register 7	CRTB_7	R	H'00000000	H'E82001E0	32
	Channel status register 7	CHSTAT_7	R	H'00000000	H'E82001E4	32
	Channel control register 7	CHCTRL_7	RW	H'00000000	H'E82001E8	32
	Channel configuration register 7	CHCFG_7	RW	H'00000000	H'E82001EC	32
	Channel interval register 7	CHITVL_7	RW	H'00000000	H'E82001F0	32
	Channel extension register 7	CHEXT_7	RW	H'00000000	H'E82001F4	32
	Next link address register 7	NXLA_7	RW	H'00000000	H'E82001F8	32
	Current link address register 7	CRLA_7	R	H'00000000	H'E82001FC	32
Common for 0 to 7	DMA control registers 0 to 7	DCTRL_0_7	R/W	H'00000000	H'E8200300	32
	DMA status EN registers 0 to 7	DSTAT_EN_0_7	R	H'00000000	H'E8200310	32
	DMA status ER registers 0 to 7	DSTAT_ER_0_7	R	H'00000000	H'E8200314	32
	DMA status END registers 0 to 7	DSTAT_END_0_7	R	H'00000000	H'E8200318	32
	DMA status TC registers 0 to 7	DSTAT_TC_0_7	R	H'00000000	H'E820031C	32
	DMA status SUS registers 0 to 7	DSTAT_SUS_0_7	R	H'00000000	H'E8200320	32

chanhvinh.com

Table 9.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
8	Next0 source address register 8	N0SA_8	RW	H'00000000	H'E8200400	32
	Next0 destination address register 8	N0DA_8	RW	H'00000000	H'E8200404	32
	Next0 transaction byte register 8	N0TB_8	RW	H'00000000	H'E8200408	32
	Next1 source address register 8	N1SA_8	RW	H'00000000	H'E820040C	32
	Next1 destination address register 8	N1DA_8	RW	H'00000000	H'E8200410	32
	Next1 transaction byte register 8	N1TB_8	RW	H'00000000	H'E8200414	32
	Current source address register 8	CRSA_8	R	H'00000000	H'E8200418	32
	Current destination address register 8	CRDA_8	R	H'00000000	H'E820041C	32
	Current transaction byte register 8	CRTB_8	R	H'00000000	H'E8200420	32
	Channel status register 8	CHSTAT_8	R	H'00000000	H'E8200424	32
	Channel control register 8	CHCTRL_8	RW	H'00000000	H'E8200428	32
	Channel configuration register 8	CHCFG_8	RW	H'00000000	H'E820042C	32
	Channel interval register 8	CHITVL_8	RW	H'00000000	H'E8200430	32
	Channel extension register 8	CHEXT_8	RW	H'00000000	H'E8200434	32
	Next link address register 8	NXLA_8	RW	H'00000000	H'E8200438	32
	Current link address register 8	CRLA_8	R	H'00000000	H'E820043C	32
9	Next0 source address register 9	N0SA_9	RW	H'00000000	H'E8200440	32
	Next0 destination address register 9	N0DA_9	RW	H'00000000	H'E8200444	32
	Next0 transaction byte register 9	N0TB_9	RW	H'00000000	H'E8200448	32
	Next1 source address register 9	N1SA_9	RW	H'00000000	H'E820044C	32
	Next1 destination address register 9	N1DA_9	RW	H'00000000	H'E8200450	32
	Next1 transaction byte register 9	N1TB_9	RW	H'00000000	H'E8200454	32
	Current source address register 9	CRSA_9	R	H'00000000	H'E8200458	32
	Current destination address register 9	CRDA_9	R	H'00000000	H'E820045C	32
	Current transaction byte register 9	CRTB_9	R	H'00000000	H'E8200460	32
	Channel status register 9	CHSTAT_9	R	H'00000000	H'E8200464	32
	Channel control register 9	CHCTRL_9	RW	H'00000000	H'E8200468	32
	Channel configuration register 9	CHCFG_9	RW	H'00000000	H'E820046C	32
	Channel interval register 9	CHITVL_9	RW	H'00000000	H'E8200470	32
	Channel extension register 9	CHEXT_9	RW	H'00000000	H'E8200474	32
	Next link address register 9	NXLA_9	RW	H'00000000	H'E8200478	32
	Current link address register 9	CRLA_9	R	H'00000000	H'E820047C	32

chanhvinh.com

Table 9.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
10	Next0 source address register 10	N0SA_10	RW	H'00000000	H'E8200480	32
	Next0 destination address register 10	N0DA_10	RW	H'00000000	H'E8200484	32
	Next0 transaction byte register 10	N0TB_10	RW	H'00000000	H'E8200488	32
	Next1 source address register 10	N1SA_10	RW	H'00000000	H'E820048C	32
	Next1 destination address register 10	N1DA_10	RW	H'00000000	H'E8200490	32
	Next1 transaction byte register 10	N1TB_10	RW	H'00000000	H'E8200494	32
	Current source address register 10	CRSA_10	R	H'00000000	H'E8200498	32
	Current destination address register 10	CRDA_10	R	H'00000000	H'E820049C	32
	Current transaction byte register 10	CRTB_10	R	H'00000000	H'E82004A0	32
	Channel status register 10	CHSTAT_10	R	H'00000000	H'E82004A4	32
	Channel control register 10	CHCTRL_10	RW	H'00000000	H'E82004A8	32
	Channel configuration register 10	CHCFG_10	RW	H'00000000	H'E82004AC	32
	Channel interval register 10	CHITVL_10	RW	H'00000000	H'E82004B0	32
	Channel extension register 10	CHEXT_10	RW	H'00000000	H'E82004B4	32
	Next link address register 10	NXLA_10	RW	H'00000000	H'E82004B8	32
	Current link address register 10	CRLA_10	R	H'00000000	H'E82004BC	32
11	Next0 source address register 11	N0SA_11	RW	H'00000000	H'E82004C0	32
	Next0 destination address register 11	N0DA_11	RW	H'00000000	H'E82004C4	32
	Next0 transaction byte register 11	N0TB_11	RW	H'00000000	H'E82004C8	32
	Next1 source address register 11	N1SA_11	RW	H'00000000	H'E82004CC	32
	Next1 destination address register 11	N1DA_11	RW	H'00000000	H'E82004D0	32
	Next1 transaction byte register 11	N1TB_11	RW	H'00000000	H'E82004D4	32
	Current source address register 11	CRSA_11	R	H'00000000	H'E82004D8	32
	Current destination address register 11	CRDA_11	R	H'00000000	H'E82004DC	32
	Current transaction byte register 11	CRTB_11	R	H'00000000	H'E82004E0	32
	Channel status register 11	CHSTAT_11	R	H'00000000	H'E82004E4	32
	Channel control register 11	CHCTRL_11	RW	H'00000000	H'E82004E8	32
	Channel configuration register 11	CHCFG_11	RW	H'00000000	H'E82004EC	32
	Channel interval register 11	CHITVL_11	RW	H'00000000	H'E82004F0	32
	Channel extension register 11	CHEXT_11	RW	H'00000000	H'E82004F4	32
	Next link address register 11	NXLA_11	RW	H'00000000	H'E82004F8	32
	Current link address register 11	CRLA_11	R	H'00000000	H'E82004FC	32

chanhvinh.com

Table 9.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
12	Next0 source address register 12	N0SA_12	RW	H'00000000	H'E8200500	32
	Next0 destination address register 12	N0DA_12	RW	H'00000000	H'E8200504	32
	Next0 transaction byte register 12	N0TB_12	RW	H'00000000	H'E8200508	32
	Next1 source address register 12	N1SA_12	RW	H'00000000	H'E820050C	32
	Next1 destination address register 12	N1DA_12	RW	H'00000000	H'E8200510	32
	Next1 transaction byte register 12	N1TB_12	RW	H'00000000	H'E8200514	32
	Current source address register 12	CRSA_12	R	H'00000000	H'E8200518	32
	Current destination address register 12	CRDA_12	R	H'00000000	H'E820051C	32
	Current transaction byte register 12	CRTB_12	R	H'00000000	H'E8200520	32
	Channel status register 12	CHSTAT_12	R	H'00000000	H'E8200524	32
	Channel control register 12	CHCTRL_12	RW	H'00000000	H'E8200528	32
	Channel configuration register 12	CHCFG_12	RW	H'00000000	H'E820052C	32
	Channel interval register 12	CHITVL_12	RW	H'00000000	H'E8200530	32
	Channel extension register 12	CHEXT_12	RW	H'00000000	H'E8200534	32
	Next link address register 12	NXLA_12	RW	H'00000000	H'E8200538	32
	Current link address register 12	CRLA_12	R	H'00000000	H'E820053C	32
13	Next0 source address register 13	N0SA_13	RW	H'00000000	H'E8200540	32
	Next0 destination address register 13	N0DA_13	RW	H'00000000	H'E8200544	32
	Next0 transaction byte register 13	N0TB_13	RW	H'00000000	H'E8200548	32
	Next1 source address register 13	N1SA_13	RW	H'00000000	H'E820054C	32
	Next1 destination address register 13	N1DA_13	RW	H'00000000	H'E8200550	32
	Next1 transaction byte register 13	N1TB_13	RW	H'00000000	H'E8200554	32
	Current source address register 13	CRSA_13	R	H'00000000	H'E8200558	32
	Current destination address register 13	CRDA_13	R	H'00000000	H'E820055C	32
	Current transaction byte register 13	CRTB_13	R	H'00000000	H'E8200560	32
	Channel status register 13	CHSTAT_13	R	H'00000000	H'E8200564	32
	Channel control register 13	CHCTRL_13	RW	H'00000000	H'E8200568	32
	Channel configuration register 13	CHCFG_13	RW	H'00000000	H'E820056C	32
	Channel interval register 13	CHITVL_13	RW	H'00000000	H'E8200570	32
	Channel extension register 13	CHEXT_13	RW	H'00000000	H'E8200574	32
	Next link address register 13	NXLA_13	RW	H'00000000	H'E8200578	32
	Current link address register 13	CRLA_13	R	H'00000000	H'E820057C	32

chanhvinh.com

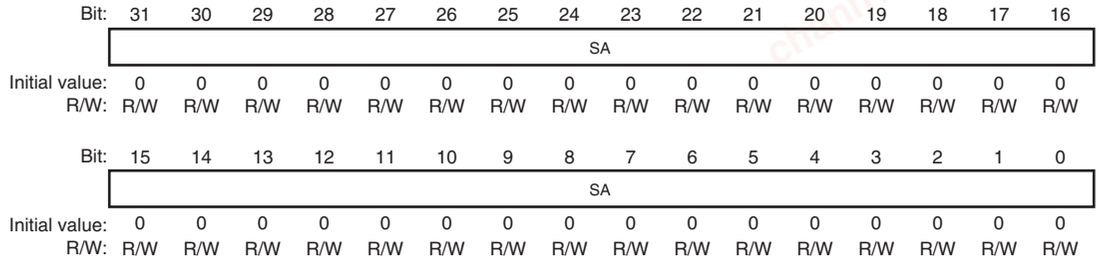
Table 9.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
14	Next0 source address register 14	N0SA_14	RW	H'00000000	H'E8200580	32
	Next0 destination address register 14	N0DA_14	RW	H'00000000	H'E8200584	32
	Next0 transaction byte register 14	N0TB_14	RW	H'00000000	H'E8200588	32
	Next1 source address register 14	N1SA_14	RW	H'00000000	H'E820058C	32
	Next1 destination address register 14	N1DA_14	RW	H'00000000	H'E8200590	32
	Next1 transaction byte register 14	N1TB_14	RW	H'00000000	H'E8200594	32
	Current source address register 14	CRSA_14	R	H'00000000	H'E8200598	32
	Current destination address register 14	CRDA_14	R	H'00000000	H'E820059C	32
	Current transaction byte register 14	CRTB_14	R	H'00000000	H'E82005A0	32
	Channel status register 14	CHSTAT_14	R	H'00000000	H'E82005A4	32
	Channel control register 14	CHCTRL_14	RW	H'00000000	H'E82005A8	32
	Channel configuration register 14	CHCFG_14	RW	H'00000000	H'E82005AC	32
	Channel interval register 14	CHITVL_14	RW	H'00000000	H'E82005B0	32
	Channel extension register 14	CHEXT_14	RW	H'00000000	H'E82005B4	32
	Next link address register 14	NXLA_14	RW	H'00000000	H'E82005B8	32
	Current link address register 14	CRLA_14	R	H'00000000	H'E82005BC	32
15	Next0 source address register 15	N0SA_15	RW	H'00000000	H'E82005C0	32
	Next0 destination address register 15	N0DA_15	RW	H'00000000	H'E82005C4	32
	Next0 transaction byte register 15	N0TB_15	RW	H'00000000	H'E82005C8	32
	Next1 source address register 15	N1SA_15	RW	H'00000000	H'E82005CC	32
	Next1 destination address register 15	N1DA_15	RW	H'00000000	H'E82005D0	32
	Next1 transaction byte register 15	N1TB_15	RW	H'00000000	H'E82005D4	32
	Current source address register 15	CRSA_15	R	H'00000000	H'E82005D8	32
	Current destination address register 15	CRDA_15	R	H'00000000	H'E82005DC	32
	Current transaction byte register 15	CRTB_15	R	H'00000000	H'E82005E0	32
	Channel status register 15	CHSTAT_15	R	H'00000000	H'E82005E4	32
	Channel control register 15	CHCTRL_15	RW	H'00000000	H'E82005E8	32
	Channel configuration register 15	CHCFG_15	RW	H'00000000	H'E82005EC	32
	Channel interval register 15	CHITVL_15	RW	H'00000000	H'E82005F0	32
	Channel extension register 15	CHEXT_15	RW	H'00000000	H'E82005F4	32
	Next link address register 15	NXLA_15	RW	H'00000000	H'E82005F8	32
	Current link address register 15	CRLA_15	R	H'00000000	H'E82005FC	32
Common for 8 to 15	DMA control registers 8 to 15	DCTRL_8_15	R/W	H'00000000	H'E8200700	32
	DMA status EN registers 8 to 15	DSTAT_EN_8_15	R	H'00000000	H'E8200710	32
	DMA status ER registers 8 to 15	DSTAT_ER_8_15	R	H'00000000	H'E8200714	32
	DMA status END registers 8 to 15	DSTAT_END_8_15	R	H'00000000	H'E8200718	32
	DMA status TC registers 8 to 15	DSTAT_TC_8_15	R	H'00000000	H'E820071C	32
	DMA status SUS registers 8 to 15	DSTAT_SUS_8_15	R	H'00000000	H'E8200720	32
0/1	DMA extended resource selector 0	DMARS0	R/W	H'00000000	H'FCFE1000	32
2/3	DMA extended resource selector 1	DMARS1	R/W	H'00000000	H'FCFE1004	32
4/5	DMA extended resource selector 2	DMARS2	R/W	H'00000000	H'FCFE1008	32
6/7	DMA extended resource selector 3	DMARS3	R/W	H'00000000	H'FCFE100C	32
8/9	DMA extended resource selector 4	DMARS4	R/W	H'00000000	H'FCFE1010	32
10/11	DMA extended resource selector 5	DMARS5	R/W	H'00000000	H'FCFE1014	32
12/13	DMA extended resource selector 6	DMARS6	R/W	H'00000000	H'FCFE1018	32
14/15	DMA extended resource selector 7	DMARS7	R/W	H'00000000	H'FCFE101C	32

chanhvinh.com

9.4.1 Next Source Address Register n (N0SA_n, N1SA_n)

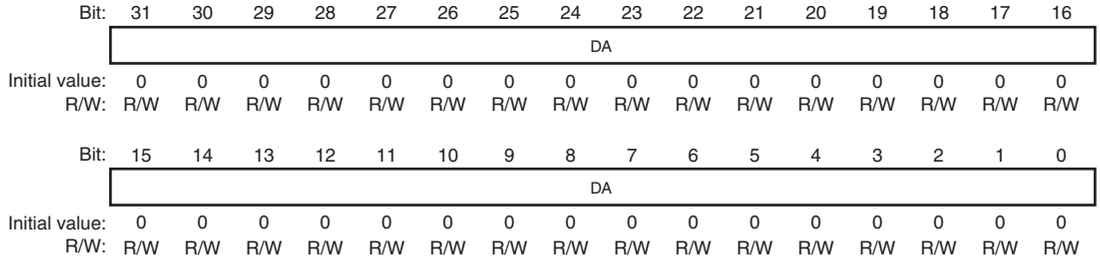
This register sets the DMA transfer source address (32 bits) of DMA channel n (n = 0 to 15) which is to be executed next.
N0SA_n is for the Next0 Register Set, and N1SA_n is for the Next1 Register Set.
In register mode, set this register set by using software. In link mode, the descriptor read data is automatically set in the Next0 Register Set.
These register set values are loaded to the Current Register Set and used for DMA transfer.



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	SA	All 0	R/W	Source Address Sets the start address of the DMA transfer source.

9.4.2 Next Destination Address Register n (N0DA_n, N1DA_n)

This register sets the DMA transfer destination address (32 bits) of DMA channel n (n = 0 to 15) which is to be executed next.
N0DA_n is for the Next0 Register Set, and N1DA_n is for the Next1 Register Set.
In register mode, set this register set by using software. In link mode, the descriptor read data is automatically set in the Next0 Register Set.
These register set values are loaded to the Current Register Set and used for DMA transfer.



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DA	All 0	R/W	Destination Address Sets the start address of the DMA transfer destination.

chanhvinh.com

9.4.3 Next Transaction Byte Register n (N0TB_n, N1TB_n)

This register sets the total transfer byte count (DMA transaction) of DMA channel (n = 0 to 15) which is to be executed next.

N0TB_n is for the Next0 Register Set, and N1TB_n is for the Next1 Register Set.

In register mode, set this register set by using software. In link mode, the descriptor read data is automatically set in the Next0 Register Set.

These register set values are loaded to the Current Register Set and used for DMA transfer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TB															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TB															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TB	All 0	R/W	Transaction Byte Sets the total transfer byte count. Caution: Do not start a DMA transaction with 0 set in this register.

9.4.4 Current Source Address Register (CRSA_n)

This register indicates the DMA transfer source address of DMA channel n (n = 0 to 15).

The values are loaded from the Next0/1 Register Set in register mode or from the descriptor read data in link mode. You cannot write to this register set using software.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CRSA															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CRSA															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CRSA	All 0	R	Current Source Address Register Indicates the read address of the next DMA transaction. The value automatically increments during the DMA transaction. (The value is fixed when 1 is set in SAD of the CHCFG_n register.) The value increments when a read transfer starts. Read this register after DMA stops (0 is set in EN of the CHSTAT_n register). (Any value obtained during the DMA operation should be handled as a reference value.)

chanhvinh.com

9.4.5 Current Destination Address Register (CRDA_n)

This register indicates the DMA transfer destination address of DMA channel n (n = 0 to 15).
The values are loaded from the Next0/1 Register Set in register mode or from the descriptor read data in link mode. You cannot write to this register set using software.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CRDA															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CRDA															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CRDA	All 0	R	Current Destination Address Register Indicates the write address of the next DMA transaction. The value automatically increments during the DMA transaction. (The value is fixed when 1 is set in DAD of the CHCFG_n register.) The value increments when a write transfer starts. Read this register after DMA stops (0 is set in EN of the CHSTAT_n register). (Any value obtained during the DMA operation should be handled as a reference value.)

9.4.6 Current Transaction Byte Register (CRTB_n)

This register indicates the total transfer byte count of DMA channel n (n = 0 to 15). The value of this register becomes 0 when the transaction ends.
The values are loaded from the Next0/1 Register Set in register mode or from the descriptor read data in link mode. You cannot write to this register set using software.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CRTB															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CRTB															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CRTB	All 0	R	Current Transaction Byte Register Indicates the remaining transfer byte count of the currently executed DMA transaction. The value automatically decrements during the DMA transaction. The value decrements when a write transfer is completed. Read this register after DMA stops (0 is set in EN of the CHSTAT_n register). (Any value obtained during the DMA operation should be handled as a reference value.)

chanhvinh.com

9.4.7 Channel Status Register n (CHSTAT_n)

This register indicates the status of DMA channel n (n = 0 to 15).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	INTMSK
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	MODE	DER	DW	DL	SR	TC	END	ER	SUS	TACT	RQST	EN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved area. Set 0. A read operation results in 0 being read.
16	INTMSK	0	R	Indicates the temporary mask status of the DMA transfer end interrupt. 1: Masked temporarily 0: Unmasked temporarily Set condition(s): • When SETINTMSK (CHCTRL_n) is set to 1 Reset condition(s): • When CLRINTMSK (CHCTRL_n) is set to 1 • When SWRST (CHCTRL_n) is set to 1
15 to 12	—	All 0	R	Reserved area. Set 0. A read operation results in 0 being read.
11	MODE	0	R	DMA Mode Indicates the DMA mode. It corresponds to the value set in the DMS bit of the CHCFG_n register. 0: Register mode 1: Link mode
10	DER	0	R	Descriptor Error Indicates whether the link valid value of the read descriptor is invalid (LV = 0) (this is not dependent on the DIM level of the descriptor). If a descriptor error has occurred, the transfer is stopped but no DMA error interrupt occurs. 0: Descriptor Error not detected 1: Descriptor Error detected Set condition(s): • When the LV value loaded with the descriptor in link mode is 0 Reset condition(s): • When SWRST (CHCTRL_n) is set to 1
9	DW	0	R	Descriptor WriteBack Indicates the descriptor writeback status. The bit maintains 1 if a bus error is received during descriptor writeback. 0: Operation other than writeback is being performed for the header in link mode. 1: (ER = 0) Writeback is being performed for the header in link mode. (ER = 1) A bus error occurs during writeback for the header in link mode. Set condition(s): • When header writeback in link mode starts Reset condition(s): • When header writeback in link mode ends with an OK response • When SWRST (CHCTRL_n) is set to 1

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
8	DL	0	R	Descriptor Load Indicates whether the descriptor is being loaded. The bit maintains 1 if a bus error is received during descriptor load. 0: Operation other than descriptor load 1: (ER = 0) Descriptor load is in progress in link mode. (ER = 1) A bus error occurs during descriptor load in link mode. Set condition(s): • When descriptor load in link mode starts Reset condition(s): • When descriptor load in link mode ends with an OK response • When SWRST (CHCTRL_n) is set to 1
7	SR	0	R	Selected Register Set Indicates the register set currently selected in register mode. 0: Next0 Register Set 1: Next1 Register Set Set condition(s): • When RSEL (CHCFG_n) is set to 1 Reset condition(s): • When RSEL (CHCFG_n) is set to 0
6	TC	0	R	Terminal Count Indicates whether the DMA transaction is completed. 0: DMA transfer not completed 1: DMA transfer completed Set condition(s): • When data equivalent to the total transfer byte count set in the CRTB register has been transferred in register mode • When data equivalent to the total transfer byte count set in the CRTB register has been transferred in link mode, with 1 set in WBD of the descriptor header • When descriptor writeback is completed in link mode, with 0 set in WBD of the descriptor header Clear condition(s): • When the CLRTC (CHCTRL_n) bit is set to 1 • When the SWRST (CHCTRL_n) bit is set to 1
5	END	0	R	DMAEND Interrupted Indicates whether the DMA transaction is completed and whether the DMA transfer end interrupt has occurred. 0: DMA transfer not completed 1: DMA transfer completed Set condition(s): • When one of the set conditions for the TC bit is met and 0 is set in DEM of the CHCFG_n register • When the descriptor is read in link mode and both LV of the header and DIM are set to 0 Clear condition(s): • When CLREND (CHCTRL_n) is set to 1 • When SWRST (CHCTRL_n) is set to 1
4	ER	0	R	Error bit Indicates that a DMA error interrupt has occurred because an error response has been received from the transfer source or destination and a bus error has occurred during the DMA transfer. 0: No bus error has occurred 1: A DMA error interrupt has occurred due to a bus error Set condition(s): • When a bus error has occurred during a bus cycle Clear condition(s): • When SWRST (CHCTRL_n) is set to 1

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
3	SUS	0	R	Suspend Indicates whether the channel is suspended. 0: Channel_n not suspended 1: Channel_n suspended Set condition(s): • When SETSUS (CHCTRL_n) is set to 1 during a DMA transfer on Channel_n, creating a SUSPEND status internally Clear condition(s): • When CLRSUS (CHCTRL_n) is set to 1 • When CLREN (CHCTRL_n) is set to 1
2	TACT	0	R	Transaction Active Indicates whether the DMAC is active. This bit is intended to check that the channel is completely inactive. 0: DMA on Channel_n inactive 1: DMA on Channel_n active Set condition(s): • When a DMA transaction starts on Channel_n Clear condition(s): • When a DMA transaction is completed
1	RQST	0	R	Request Indicates whether a transfer request is being received. 0: DMA transfer request not being received 1: DMA transfer request being received Set condition(s): • When the STG bit (CHCTRL_n) is set to 1 (auto request) • When a transfer request is received from the DMA request source set in the CHCFG_n register Clear condition(s): • When SWRST (CHCTRL_n) is set to 1 • When CLRRQ (CHCTRL_n) is set to 1 • When a transfer is executed on the side specified by REQD (CHCFG_n) in single transfer mode (TM = 0). • When all DMA transactions are completed in register mode (the transaction ends with REN set to 0) • When the DMA transfer of the last descriptor (LE = 1) is completed in link mode • When descriptor read stops (LV = 0) in link mode • When a DMA transaction is completed in link mode while DEM is set to 0 • When a bus error is received due to an error response
0	EN	0	R	Enable Indicates whether the operation of DMA channel n is enabled or disabled. 0: Operation disabled 1: Operation enabled Set condition(s): • When SETEN (CHCTRL_n) is set to 1 Clear condition(s): • When SWRST (CHCTRL_n) is set to 1 • When CLREN (CHCTRL_n) is set to 1 • When a bus error is received due to an error response during the transfer • When all DMA transactions are completed in register mode (the transaction ends with REN set to 0) • When the DMA transfer of the last descriptor (LE = 1) is completed in link mode (writeback when WBD is set to 0) • When descriptor read stops (LV = 0) in link mode

<Caution> If the ER bit is set to 1 for any transfer, the whole transfer should be handled as invalid.
To suspend a DMA transaction, mask or clear the transfer request or clear the Enable bit (for the procedure, see section 9.7.11 (2) Transfer Stop).
If you make a transfer request from an on-chip peripheral module or the external DREQ input concurrently with an auto request (by setting 1 in the STG bit) for the same one channel, you cannot identify the trigger source that takes effect. Make sure that only one of these transfer requests is used in the system.
When transfer is requested by an auto request, wait for the last requested DMA transfer to complete (use the Current Register or other data to check the status) before setting the STG bit for the next transfer request.

chanhvinh.com

9.4.8 Channel Control Register n (CHCTRL_n)

This register controls the DMA transfer operation on DMA channel n (n = 0 to 15).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CLRINT MSK	SETINT MSK
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	CLR SUS	SET SUS	-	CLRTC	CLR END	CLRRQ	SWRST	STG	CLREN	SETEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved area. Set 0. A read operation results in 0 being read.
17	CLRINTMSK	0	R/W	When this bit is set to 1, the mask of the DMA transfer end interrupt is cleared. Also, the INTMSK bit of the CHSTAT_n register is set to 0. If the mask is cleared when 1 is set in both LVINT of the DCTRL register and END of the CHSTAT_n register, the DMA transfer end interrupt becomes active. (It does not become active when 0 is set in LVINT.) A read operation results in 0 being read. 1: Clears the mask set by SETINTMSK. 0: Does not affect the operation.
16	SETINTMSK	0	R/W	When this bit is set to 1, the DMA transfer end interrupt is temporarily masked. Also, the INTMSK bit of the CHSTAT_n register is set to 1. A read operation results in 0 being read. 1: Masks the DMA transfer end interrupt. 0: Does not affect the operation.
15 to 10	—	All 0	R	Reserved area. Set 0. A read operation results in 0 being read.
9	CLRSUS	0	R/W	Clear Suspend Clears the suspend status. Setting this bit to 1 when 1 is set in SUS of the CHSTAT_n register can clear the suspend status. An attempt to read this bit results in 0 being read. 1: Clears the suspend status of the current DMA transfer. 0: Does not affect the operation.
8	SETSUS	0	R/W	Set Suspend Suspends the current DMA transfer. Setting this bit to 1 when 1 is set in EN of the CHSTAT_n register can suspend the current DMA transfer. An attempt to read this bit results in 0 being read. 1: Suspends the current DMA transfer. 0: Does not affect the operation.
7	—	0	R	Reserved area. Set 0. A read operation results in 0 being read.
6	CLRTC	0	R/W	Clear TC bit Setting this bit to 1 can clear the TC bit of the CHSTAT_n register. An attempt to read this bit results in 0 being read. 1: Clears the TC bit. 0: Does not affect the operation.
5	CLREND	0	R/W	Clear End bit Setting this bit to 1 can clear the END bit of the CHSTAT_n register. Also, the DMA transfer end interrupt is cleared. An attempt to read this bit results in 0 being read. 1: Clears the END bit. 0: Does not affect the operation.
4	CLRRQ	0	R/W	Clear Request bit Setting this bit to 1 can clear the RQST bit of the CHSTAT_n register. An attempt to read this bit results in 0 being read. 1: Clears the RQST bit. 0: Does not affect the operation.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
3	SWRST	0	R/W	Software Reset Setting this bit to 1 can clear the channel status register (CHSTAT_n). When setting this bit to 1, make sure that both the EN bit and TACT bit are set to 0. An attempt to read this bit results in 0 being read. 1: Resets the channel status register. 0: Does not affect the operation.
2	STG	0	R/W	Software Trigger Setting this bit to 1 sets an auto request. If this bit is set at the same time the SWRST bit is set, the clear operation by the SWRST bit takes precedence. An attempt to read this bit results in 0 being read. 1: Sets a transfer request triggered by an auto request (sets 1 in the RQST bit). 0: Does not affect the operation.
1	CLREN	0	R/W	Clear Enable Setting this bit to 1 can clear the EN bit (for details, see section 9.7.11 (2) Transfer Stop). An attempt to read this bit results in 0 being read. 1: Stops the DMA transfer (clears the EN bit). 0: Does not affect the operation.
0	SETEN	0	R/W	Set Enable Enables a DMA transfer on DMA channel n. If this bit is set at the same time the SWRST bit is set, the clear operation by the SWRST bit takes precedence and the transfer does not start. An attempt to read this bit results in 0 being read. 1: Enables a DMA transfer (sets 1 in the EN bit). 0: Does not affect the operation.

chanhvinh.com

9.4.9 Channel Configuration Register n (CHCFG_n)

This register controls the DMA transfer operation on DMA channel n (n = 0 to 15).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DMS	REN	RSW	RSEL	SBE	-	-	DEM	-	TM	DAD	SAD	DDS[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDS[3:0]				-	AM[2:0]			-	LVL	HIEN	LOEN	REQD	SEL[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	DMS	0	R/W	DMA Mode Select Sets the DMA mode. 0: Register mode (initial value) 1: Link mode
30	REN	0	R/W	Register Set Enable After a DMA transaction is completed, DMA transfers are continued using the Next Register Set selected by RSEL. This bit is valid only in register mode. 0: Does not continue DMA transfers. 1: Continues DMA transfers. Set condition(s): • When 1 is written to this bit Clear condition(s): • When 0 is written to this bit • When a DMA transaction is completed, with REN set to 1
29	RSW	0	R/W	Register Select Switch Inverts RSEL automatically after a DMA transaction is completed. This bit is valid only in register mode. 0: Does not invert RSEL automatically after a DMA transaction (initial value). 1: Inverts RSEL automatically after a DMA transaction.
28	RSEL	0	R/W	Register Set Select Selects the Next Register Set to be executed next. This bit is valid only in register mode. When RSW is set to 1, this bit is inverted automatically when a DMA transaction is completed. 0: Executes the Next0 Register Set (initial value). 1: Executes the Next1 Register Set. Transition condition(s): • When a DMA transaction is completed, with RSW set to 1
27	SBE	0	R/W	Sweep Buffer Enable Selects whether to sweep (write) the data already read into the buffer and stop the DMA transfer if the Enable bit is cleared to 0 during a DMA transaction. The sweep mode is available only when REQD is set to 0. 0: Stops the DMA transfer without sweeping the buffer (initial value). 1: Stops the DMA transfer after sweeping the buffer.
26, 25	—	0	R	Reserved area. Set 0. A read operation results in 0 being read.
24	DEM	0	R/W	DMA Transfer End Interrupt Mask Masks the DMA transfer end interrupt for register mode transfer. If 1 is set in this bit when a DMA transfer end interrupt is output, the DMA transfer end interrupt signal is not asserted. In this case, DEM is cleared to 0 automatically. 0: Does not mask the DMA transfer end interrupt (initial value). 1: Masks the DMA transfer end interrupt. Clear condition(s): • When a DMA transaction is completed with DEM set to 1
23	—	0	R	Reserved area. Set 0. A read operation results in 0 being read.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
22	TM	0	R/W	Transfer Mode Sets the DMA transfer mode. 0: Single transfer mode (initial value) 1: Block transfer mode
21	DAD	0	R/W	Sets the destination address counting direction of DMA channel n. 0: Increment (initial value) 1: Fixed
20	SAD	0	R/W	Sets the source address counting direction of DMA channel n. 0: Increment (initial value) 1: Fixed
19 to 16	DDS [3:0]	0000	R/W	Destination Data Size Sets the DMA transfer size of the transfer destination.

Value	Size	Remark
0000	8 bits	Initial value
0001	16 bits	
0010	32 bits	
0011	64 bits	
0100	128 bits	
0101	256 bits	
0110	512 bits	
0111	1024 bits	
Other than the above	—	Setting prohibited

15 to 12	SDS [3:0]	0000	R/W	Source Data Size Sets the DMA transfer size of the transfer source.

Value	Size	Remark
0000	8 bits	Initial value
0001	16 bits	
0010	32 bits	
0011	64 bits	
0100	128 bits	
0101	256 bits	
0110	512 bits	
0111	1024 bits	
Other than the above	—	Setting prohibited

11	—	0	R	Reserved area. Set 0. A read operation results in 0 being read.
10 to 8	AM [2:0]	000	R/W	ACK Mode Sets the DMAACK output mode. 000: (initial value) 001: Level mode (active until the transfer request from an on-chip peripheral module or the external DREQ input becomes inactive) 01x: Bus cycle mode (active while the DMA transfer is in a bus cycle) 1xx: DMAACK not to be output (this setting should be made when an auto request is made by STG (CHCTRL_n))
7	—	0	R	Reserved area. Set 0. A read operation results in 0 being read.
6	LVL	0	R/W	Level Selects whether to detect a DMA request based on the level or edge of the signal. 0: Detects based on the edge (initial value). 1: Detects based on the level.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
5	HIEN	0	R/W	High Enable Selects whether to detect a DMA request using the High level or rising edge of the signal. When LVL = 0: HIEN = 1: Detects a request in response to the rising edge of the signal. HIEN = 0: Does not detect a request in response to the rising edge of the signal (initial value). When LVL = 1: HIEN = 1: Detects a request when the signal is at the High level. HIEN = 0: Does not detect a request even when the signal is at the High level (initial value).
4	LOEN	0	R/W	Low Enable Selects whether to detect a DMA request using the Low level or falling edge of the signal. When LVL = 0: LOEN = 1: Detects a request in response to the falling edge of the signal. LOEN = 0: Does not detect a request in response to the falling edge of the signal (initial value). When LVL = 1: LOEN = 1: Detects a request when the signal is at the Low level. LOEN = 0: Does not detect a request even when the signal is at the Low level (initial value).
3	REQD	0	R/W	Request Direction Selects whether DMAREQ selected by the SEL bit is the source or destination. This bit is also used to define when DMAACK is to become active. 0: Source; DMAACK is to become active when read (initial value). 1: Destination; DMAACK is to become active when written.
2 to 0	SEL[2:0]	000	R/W	These bits are used to set a DMAC channel. Set one of the following values so that the channel set by the SEL bits matches the CHCFG_n channel. 000: CH0/CH8 001: CH1/CH9 010: CH2/CH10 011: CH3/CH11 100: CH4/CH12 101: CH5/CH13 110: CH6/CH14 111: CH7/CH15

9.4.10 Channel Interval Register n (CHITVL_n)

This register sets the transfer interval for DMA channel n (n = 0 to 15).

For details, see section 9.7.9, Interval Count Function.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ITVL															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved area. Set 0. A read operation results in 0 being read.
15 to 0	ITVL	All 0	R/W	Sets the channel transfer interval.

chanhvinh.com

9.4.11 Channel Extension Register n (CHEXT_n)

This is an extension register for DMA channel n (n = 0 to 15).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DCA[3:0]				-	DPR[2:0]			SCA[3:0]				-	SPR[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Set 0. A read operation results in 0 being read.
15 to 12	DCA[3:0]	0000	R/W	Destination CACHE Sets the value to be output to AWCACHE[3:0] for DMA write transfer. See Note 1 below.
11	—	0	R	Set 0. A read operation results in 0 being read.
10 to 8	DPR[2:0]	000	R/W	Destination PROT Sets the value to be output to AWPROT[2:0] for DMA write transfer. See Note 2 below.
7 to 4	SCA[3:0]	0000	R/W	Source CACHE Sets the value to be output to ARCACHE[3:0] for DMA read transfer. See Note 1 below.
3	—	0	R	Set 0. A read operation results in 0 being read.
2 to 0	SPR[2:0]	000	R/W	Source PROT Sets the value to be output to ARPROT[2:0] for DMA read transfer. See Note 2 below.

Note 1. Cache support: Bits SCA and DCA are used to change the settings.
When the transfer destination or source is not in the external bus space, set these bits to 0000.
Even when the transfer destination or source is in the external bus space but the secondary cache is not in use, set these bits to 0000. In this case, the DACK0 output and TEND0 output are issued in response to the DREQ0 transfer request.
When the secondary cache is in use in the external bus space, set CACHE[3:0].

Note 2. Protection unit support: Bits SPR and DPR are used to change the settings.

For the setting value, see AMBA AXI Protocol Specification from Arm Limited.

chanhvinh.com

9.4.12 Next Link Address Register n (NXLA_n)

This is a 32-bit register that sets the link address of DMA channel n (n = 0 to 15).
For information about the link mode, see section 9.6.3, Link Mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NXLA															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NXLA															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	NXLA	All 0	R/W	Sets a link address. The low-order 2 bits are masked with 0s. Only an address aligned with a 4-byte boundary can be set.

9.4.13 Current Link Address Register n (CRLA_n)

This is a 32-bit register that indicates the link address of DMA channel n (n = 0 to 15).
For information about the link mode, see section 9.6.3, Link Mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CRLA															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CRLA															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CRLA	All 0	R	Indicates the address of the currently executed descriptor.

chanhvinh.com

9.4.14 DMA Control Register (DCTRL_0_7, DCTRL_8_15)

This register sets the transfer type for descriptor access and the arbitration between channels.
(DCTRL_0_7 is common for channels 0 to 7 and DCTRL_8_15 is common for channels 8 to 15.)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LWCA				-	LWPR			LDCA				-	LDPR		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LVINT	PR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	LWCA	0000	R/W	Link WriteBack CACHE Sets the value to be output to AWCACHE[3:0] during descriptor writeback in link mode. For the setting value, see Note in section 9.4.11, Channel Extension Register n (CHEXT_n).
27	—	0	R	Reserved area. Set 0. The initial value is 0.
26 to 24	LWPR	000	R/W	Link WriteBack PROT Sets the value to be output to AWPROT[2:0] during descriptor writeback in link mode. For the setting value, see AMBA AXI Protocol Specification from Arm Limited.
23 to 20	LDCA	0000	R/W	Link Descriptor CACHE Sets the value to be output to ARCACHE[3:0] during descriptor load in link mode. For the setting value, see Note in section 9.4.11, Channel Extension Register n (CHEXT_n).
19	—	0	R	Reserved area. Set 0. The initial value is 0.
18 to 16	LDPR	000	R/W	Link Descriptor PROT Sets the value to be output to ARPROT[2:0] during descriptor load in link mode. For the setting value, see AMBA AXI Protocol Specification from Arm Limited.
15 to 2	—	All 0	R	Reserved area. Set 0. A read operation results in 0 being read.
1	LVINT	0	R/W	Sets whether to use pulse output or level output for the DMA transfer end interrupt and DMA error interrupt. Set pulse output for this product. 0: Pulse output (initial value) 1: Level output
0	PR	0	R/W	Sets the transfer priority control mode between channels (see section 9.7.2, Priority Control for DMA Channels. 0: Fixed priority mode (initial value) 1: Round robin mode

chanhvinh.com

9.4.15 DMA Status EN Register (DSTAT_EN_0_7)

This register indicates the EN bit status of the CHSTAT_n register (n = 0 to 7).
Even if you write to this register, the values of the individual bits do not change.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved area. Set 0. A read operation results in 0 being read.
7	EN7	0	R	Indicates the EN bit status of DMA channel 7.
6	EN6	0	R	Indicates the EN bit status of DMA channel 6.
5	EN5	0	R	Indicates the EN bit status of DMA channel 5.
4	EN4	0	R	Indicates the EN bit status of DMA channel 4.
3	EN3	0	R	Indicates the EN bit status of DMA channel 3.
2	EN2	0	R	Indicates the EN bit status of DMA channel 2.
1	EN1	0	R	Indicates the EN bit status of DMA channel 1.
0	EN0	0	R	Indicates the EN bit status of DMA channel 0.

9.4.16 DMA Status EN Register (DSTAT_EN_8_15)

This register indicates the EN bit status of the CHSTAT_n register (n = 8 to 15).
Even if you write to this register, the values of the individual bits do not change.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved area. Set 0. A read operation results in 0 being read.
7	EN15	0	R	Indicates the EN bit status of DMA channel 15.
6	EN14	0	R	Indicates the EN bit status of DMA channel 14.
5	EN13	0	R	Indicates the EN bit status of DMA channel 13.
4	EN12	0	R	Indicates the EN bit status of DMA channel 12.
3	EN11	0	R	Indicates the EN bit status of DMA channel 11.
2	EN10	0	R	Indicates the EN bit status of DMA channel 10.
1	EN9	0	R	Indicates the EN bit status of DMA channel 9.
0	EN8	0	R	Indicates the EN bit status of DMA channel 8.

chanhvinh.com

9.4.17 DMA Status ER Register (DSTAT_ER_0_7)

This register indicates the ER bit status of the CHSTAT_n register (n = 0 to 7).
Even if you write to this register, the values of the individual bits do not change.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved area. Set 0. A read operation results in 0 being read.
7	ER7	0	R	Indicates the ER bit status of DMA channel 7.
6	ER6	0	R	Indicates the ER bit status of DMA channel 6.
5	ER5	0	R	Indicates the ER bit status of DMA channel 5.
4	ER4	0	R	Indicates the ER bit status of DMA channel 4.
3	ER3	0	R	Indicates the ER bit status of DMA channel 3.
2	ER2	0	R	Indicates the ER bit status of DMA channel 2.
1	ER1	0	R	Indicates the ER bit status of DMA channel 1.
0	ER0	0	R	Indicates the ER bit status of DMA channel 0.

9.4.18 DMA Status ER Register (DSTAT_ER_8_15)

This register indicates the ER bit status of the CHSTAT_n register (n = 8 to 15).
Even if you write to this register, the values of the individual bits do not change.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	ER15	ER14	ER13	ER12	ER11	ER10	ER9	ER8
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved area. Set 0. A read operation results in 0 being read.
7	ER15	0	R	Indicates the ER bit status of DMA channel 15.
6	ER14	0	R	Indicates the ER bit status of DMA channel 14.
5	ER13	0	R	Indicates the ER bit status of DMA channel 13.
4	ER12	0	R	Indicates the ER bit status of DMA channel 12.
3	ER11	0	R	Indicates the ER bit status of DMA channel 11.
2	ER10	0	R	Indicates the ER bit status of DMA channel 10.
1	ER9	0	R	Indicates the ER bit status of DMA channel 9.
0	ER8	0	R	Indicates the ER bit status of DMA channel 8.

chanhvinh.com

9.4.19 DMA Status END Register (DSTAT_END_0_7)

This register indicates the END bit status of the CHSTAT_n register (n = 0 to 7).
Even if you write to this register, the values of the individual bits do not change.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	END7	END6	END5	END4	END3	END2	END1	END0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved area. Set 0. A read operation results in 0 being read.
7	END7	0	R	Indicates the END bit status of DMA channel 7.
6	END6	0	R	Indicates the END bit status of DMA channel 6.
5	END5	0	R	Indicates the END bit status of DMA channel 5.
4	END4	0	R	Indicates the END bit status of DMA channel 4.
3	END3	0	R	Indicates the END bit status of DMA channel 3.
2	END2	0	R	Indicates the END bit status of DMA channel 2.
1	END1	0	R	Indicates the END bit status of DMA channel 1.
0	END0	0	R	Indicates the END bit status of DMA channel 0.

9.4.20 DMA Status END Register (DSTAT_END_8_15)

This register indicates the END bit status of the CHSTAT_n register (n = 8 to 15).
Even if you write to this register, the values of the individual bits do not change.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	END15	END14	END13	END12	END11	END10	END9	END8
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved area. Set 0. A read operation results in 0 being read.
7	END15	0	R	Indicates the END bit status of DMA channel 15.
6	END14	0	R	Indicates the END bit status of DMA channel 14.
5	END13	0	R	Indicates the END bit status of DMA channel 13.
4	END12	0	R	Indicates the END bit status of DMA channel 12.
3	END11	0	R	Indicates the END bit status of DMA channel 11.
2	END10	0	R	Indicates the END bit status of DMA channel 10.
1	END9	0	R	Indicates the END bit status of DMA channel 9.
0	END8	0	R	Indicates the END bit status of DMA channel 8.

chanhvinh.com

9.4.21 DMA Status TC Register (DSTAT_TC_0_7)

This register indicates the TC bit status of the CHSTAT_n register (n = 0 to 7).
Even if you write to this register, the values of the individual bits do not change.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved area. Set 0. A read operation results in 0 being read.
7	TC7	0	R	Indicates the TC bit status of DMA channel 7.
6	TC6	0	R	Indicates the TC bit status of DMA channel 6.
5	TC5	0	R	Indicates the TC bit status of DMA channel 5.
4	TC4	0	R	Indicates the TC bit status of DMA channel 4.
3	TC3	0	R	Indicates the TC bit status of DMA channel 3.
2	TC2	0	R	Indicates the TC bit status of DMA channel 2.
1	TC1	0	R	Indicates the TC bit status of DMA channel 1.
0	TC0	0	R	Indicates the TC bit status of DMA channel 0.

9.4.22 DMA Status TC Register (DSTAT_TC_8_15)

This register indicates the TC bit status of the CHSTAT_n register (n = 8 to 15).
Even if you write to this register, the values of the individual bits do not change.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved area. Set 0. A read operation results in 0 being read.
7	TC15	0	R	Indicates the TC bit status of DMA channel 15.
6	TC14	0	R	Indicates the TC bit status of DMA channel 14.
5	TC13	0	R	Indicates the TC bit status of DMA channel 13.
4	TC12	0	R	Indicates the TC bit status of DMA channel 12.
3	TC11	0	R	Indicates the TC bit status of DMA channel 11.
2	TC10	0	R	Indicates the TC bit status of DMA channel 10.
1	TC9	0	R	Indicates the TC bit status of DMA channel 9.
0	TC8	0	R	Indicates the TC bit status of DMA channel 8.

chanhvinh.com

9.4.23 DMA Status SUS Register (DSTAT_SUS_0_7)

This register indicates the SUS bit status of the CHSTAT_n register (n = 0 to 7).
Even if you write to this register, the values of the individual bits do not change.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	SUS7	SUS6	SUS5	SUS4	SUS3	SUS2	SUS1	SUS0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved area. Set 0. A read operation results in 0 being read.
7	SUS7	0	R	Indicates the SUS bit status of DMA channel 7.
6	SUS6	0	R	Indicates the SUS bit status of DMA channel 6.
5	SUS5	0	R	Indicates the SUS bit status of DMA channel 5.
4	SUS4	0	R	Indicates the SUS bit status of DMA channel 4.
3	SUS3	0	R	Indicates the SUS bit status of DMA channel 3.
2	SUS2	0	R	Indicates the SUS bit status of DMA channel 2.
1	SUS1	0	R	Indicates the SUS bit status of DMA channel 1.
0	SUS0	0	R	Indicates the SUS bit status of DMA channel 0.

9.4.24 DMA Status SUS Register (DSTAT_SUS_8_15)

This register indicates the SUS bit status of the CHSTAT_n register (n = 8 to 15).
Even if you write to this register, the values of the individual bits do not change.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	SUS15	SUS14	SUS13	SUS12	SUS11	SUS10	SUS9	SUS8
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved area. Set 0. A read operation results in 0 being read.
7	SUS15	0	R	Indicates the SUS bit status of DMA channel 15.
6	SUS14	0	R	Indicates the SUS bit status of DMA channel 14.
5	SUS13	0	R	Indicates the SUS bit status of DMA channel 13.
4	SUS12	0	R	Indicates the SUS bit status of DMA channel 12.
3	SUS11	0	R	Indicates the SUS bit status of DMA channel 11.
2	SUS10	0	R	Indicates the SUS bit status of DMA channel 10.
1	SUS9	0	R	Indicates the SUS bit status of DMA channel 9.
0	SUS8	0	R	Indicates the SUS bit status of DMA channel 8.

chanhvinh.com

9.4.25 DMA Extension Resource Selectors 0 to 7 (DMARS0 to DMARS7)

DMARS are 32-bit readable/writable registers that specify the DMA transfer sources from peripheral modules in each channel. DMARS0 is for channels 0 and 1, DMARS1 is for channels 2 and 3, and so on.

Table 9.4 shows the specifiable combinations.

DMARS can specify transfer requests to be accepted for the following triggers.

The following modules can issue on-chip peripheral module requests.

Serial communication interface with FIFO: 10 sources
A/D converter: 1 source
Multi-function timer pulse unit 2: 5 sources
USB2.0 host/function module: 4 sources
Serial sound interface: 7 sources
Renesas SPDIF interface: 2 sources
CD-ROM decoder: 1 source (RZ/A1L only)
SD host interface: 4 sources
MMC host interface: 2 sources
Renesas serial peripheral interface: 6 sources
IEBus™ controller: 2 sources (RZ/A1L only)
OS timer: 2 sources
SCUX: 8 sources
Media local bus: 1 source (RZ/A1L only)
Serial communication interface: 4 sources
I^2C bus interface: 8 sources
LIN interface: 2 sources (RZ/A1L only)

Some on-chip peripheral modules in this product use the same signal both for an interrupt request and for a DMA transfer request. If such a module is selected by a DMARS register, the signal works as a DMA transfer request signal and interrupt requests to the interrupt controller are masked. To enable the interrupt, clear the setting of DMARS (set all MID[6:0] and RID[1:0] to 0).

chanhvinh.com

- DMARS0

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	CH1 MID[6:0]							CH1 RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	CH0 MID[6:0]							CH0 RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- DMARS1

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	CH3 MID[6:0]							CH3 RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	CH2 MID[6:0]							CH2 RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- DMARS2

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	CH5 MID[6:0]							CH5 RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	CH4 MID[6:0]							CH4 RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- DMARS3

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	CH7 MID[6:0]							CH7 RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	CH6 MID[6:0]							CH6 RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

chanhvinh.com

- DMARS4

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	CH9 MID[6:0]							CH9 RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	CH8 MID[6:0]							CH8 RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- DMARS5

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	CH11 MID[6:0]							CH11 RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	CH10 MID[6:0]							CH10 RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- DMARS6

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	CH13 MID[6:0]							CH13 RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	CH12 MID[6:0]							CH12 RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- DMARS7

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	CH15 MID[6:0]							CH15 RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	CH14 MID[6:0]							CH14 RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

chanhvinh.com

9.5 Operation

When there is a DMA transfer request, the DMAC starts the transfer according to the predetermined channel priority order; when the transfer end conditions are satisfied, it ends the transfer. Transfers can be requested in three modes: auto request, external request, and on-chip peripheral module request.

9.5.1 Transfer Flow

After the next source address register (N0SA_n/N1SA_n), next destination address register (N0DA_n/N1DA_n), next transaction byte register (N0TB_n/N1TB_n), channel control register (CHCTRL_n), channel configuration register (CHCFG_n), channel extension register (CHEXT_n), DMA control register (DCTRL_0_7/DCTRL_8_15), and DMA extension resource selector (DMARS) are set for the target transfer conditions, the DMAC transfers data according to the following procedure:

1. Checks to see if transfer is enabled (EN = 0 and TACT = 0 in channel status register).
2. Clears the channel status register (set 1 in the SWRST bit of the channel control register).
3. Enables DMA transfer (set 1 in the SETEN bit of the channel control register).
4. When a transfer request comes and transfer is enabled, the DMAC transfers one transfer unit of data (depending on the DDS[3:0] and SDS[3:0] bit settings). For an auto request, the transfer begins automatically when 1 is set in the STG bit of the channel control register. The CRTB_n value will be decremented by 1 for each transfer.
5. If 0 is set in the REN bit of the channel configuration register when transfer has been completed for the specified count (when CRTB_n reaches 0), transfer ends normally. If the DEM bit of the channel configuration register is set to 0 at this time, a DMA transfer end interrupt is sent to the CPU. If the REN bit is 1 when CRTB_n reaches 0, transfer operations are continued with the values of N0SA_n/N1SA_n, N0DA_n/N1DA_n, and N0TB_n/N1TB_n set by the RSEL bit of the channel configuration register until there are no more transfer requests.
6. When an address error in the DMAC is generated, the transfer is stopped. Transfers are also stopped when 1 is set in the CLREN bit of CHCTRL_n.

9.5.2 DMA Transfer Requests

DMA transfer requests are basically generated in either the data transfer source or destination, but they can also be generated in external devices and on-chip peripheral modules that are neither the transfer source nor destination. Transfers can be requested in three modes: auto request, external request, and on-chip peripheral module request. External request or on-chip peripheral module request is selected by the DMARS0 to DMARS7 registers.

(1) Auto-Request Mode

When there is no transfer request signal from an external source, as in a memory-to-memory transfer or a transfer between memory and an on-chip peripheral module unable to request a transfer, auto-request mode allows the DMAC to automatically generate a transfer request signal internally. When the STG bit in channel control register is set to 1, the transfer begins so long as the TACT bit in channel status register is 0.

(2) External Request Mode

In this mode a transfer is performed at the transfer request signal (DREQ0) of an external device of the LSI. When the DMA transfer is enabled, DMA transfer is performed upon a DREQ input.
Choose to detect DREQ0 by either the edge or level of the signal input with the LVL, HIEN, and LOEN bits in channel configuration register 0 as shown below. The source of the transfer request does not have to be the data transfer source or destination.
For the output level settings of the DACK0 and TEND0 pins, refer to section 8, Bus State Controller.

chanhvinh.com

Table 9.3 Settings for External Request Detection

	CHCFG_0		
LVL	**HIEN**	**LOEN**	**Detection of External Request**
0	0	1	Falling edge detection
	1	0	Rising edge detection
1	0	1	Low level detection
	1	0	High level detection

When DREQ0 is accepted, the DREQ0 pin enters the request accept disabled state (non-sensitive period). After issuing an acknowledge DACK0 signal for the accepted DREQ0, the DREQ0 pin again enters the request accept enabled state.

(3) On-Chip Peripheral Module Request Mode

In this mode, the transfer is performed in response to the DMA transfer request signal from an on-chip peripheral module.

When a transfer request signal is sent in on-chip peripheral module request mode while DMA transfer is enabled, the DMA transfer is performed.

The DMA transfer request signals to be sent from on-chip peripheral modules or external pin input are listed in Table 9.4.

The transfer source or destination is fixed for some on-chip peripheral module requests. For details, see Table 9.4.

Table 9.4 On-Chip Peripheral Module Requests

				DMARS		CHCFG_n						
DMA Transfer Request Source	**DMA Transfer Request Signal**	**Transfer Source**	**Transfer Destination**	**MID**	**RID**	**TM**	**AM [2:0]**	**LVL**	**HIEN**	**LO EN**	**RE QD**	**SEL[2:0]**
OS timer channel 0	OSTM0TINT (compare match)	Arbitrary	Arbitrary	000_1000	11	0/1	010	0	1	0	0/1	Ch0: 000 Ch1: 001 Ch2: 010 Ch3: 011 Ch4: 100 Ch5: 101 Ch6: 110 Ch7: 111 Ch8: 000 Ch9: 001 Ch10: 010 Ch11: 011 Ch12: 100 Ch13: 101 Ch14: 110 Ch15: 111
OS timer channel 1	OSTM1TINT (compare match)	Arbitrary	Arbitrary	000_1001	11							
Multi-function timer pulse unit 2 channel 0	TGIA_0 (input capture/ compare match)	Arbitrary	Arbitrary	001_0000	11	0/1	001	1			0/1	
Multi-function timer pulse unit 2 channel 1	TGIA_1 (input capture/ compare match)	Arbitrary	Arbitrary	001_0001	11							
Multi-function timer pulse unit 2 channel 2	TGIA_2 (input capture/ compare match)	Arbitrary	Arbitrary	001_0010	11							
Multi-function timer pulse unit 2 channel 3	TGIA_3 (input capture/ compare match)	Arbitrary	Arbitrary	001_0011	11							
Multi-function timer pulse unit 2 channel 4	TGIA_4 (input capture/ compare match)	Arbitrary	Arbitrary	001_0100	11							
Serial communication interface with FIFO channel 0	TXI0 (transmit empty)	Arbitrary	SCFTDR_0	001_1000	01	0	010	1			1	
	RXI0 (receive data full)	SCFRDR_0	Arbitrary		10						0	
Serial communication interface with FIFO channel 1	TXI1 (transmit empty)	Arbitrary	SCFTDR_1	001_1001	01						1	
	RXI1 (receive data full)	SCFRDR_1	Arbitrary		10						0	
Serial communication interface with FIFO channel 2	TXI2 (transmit empty)	Arbitrary	SCFTDR_2	001_1010	01						1	
	RXI2 (receive data full)	SCFRDR_2	Arbitrary		10						0	

chanhvinh.com

Table 9.4 On-Chip Peripheral Module Requests

DMA Transfer Request Source	DMA Transfer Request Signal	Transfer Source	Transfer Destination	DMARS		CHCFG_n						
				MID	RID	TM	AM [2:0]	LVL	HIEN	LOEN	REQD	SEL[2:0]
Serial communication interface with FIFO channel 3	TXI3 (transmit empty)	Arbitrary	SCFTDR_3	001_1011	01	0	010	1	1	0	1	Ch0: 000 Ch1: 001 Ch2: 010 Ch3: 011 Ch4: 100 Ch5: 101 Ch6: 110 Ch7: 111 Ch8: 000 Ch9: 001 Ch10: 010 Ch11: 011 Ch12: 100 Ch13: 101 Ch14: 110 Ch15: 111
	RXI3 (receive data full)	SCFRDR_3	Arbitrary		10						0	
Serial communication interface with FIFO channel 4	TXI4 (transmit empty)	Arbitrary	SCFTDR_4	001_1100	01						1	
	RXI4 (receive data full)	SCFRDR_4	Arbitrary		10						0	
USB2.0 host/ function module channel 0	USB0_DMA0 (channel 0 transmit FIFO empty)	Arbitrary	D0FIFO_0 D0FIFOBn_0 (n = 0 to 7)	010_0000	11		010	1			1	
	USB0_DMA0 (channel 0 receive FIFO full)	D0FIFO_0 D0FIFOBn_0 (n = 0 to 7)	Arbitrary								0	
	USB0_DMA1 (channel 1 transmit FIFO empty)	Arbitrary	D1FIFO_0 D1FIFOBn_0 (n = 0 to 7)	010_0001	11						1	
	USB0_DMA1 (channel 1 receive FIFO full)	D1FIFO_0 D1FIFOBn_0 (n = 0 to 7)	Arbitrary								0	
USB2.0 host/ function module channel 1	USB1_DMA0 (channel 0 transmit FIFO empty)	Arbitrary	D0FIFO_1 D0FIFOBn_1 (n = 0 to 7)	010_0010	11						1	
	USB1_DMA0 (channel 0 receive FIFO full)	D0FIFO_1 D0FIFOBn_1 (n = 0 to 7)	Arbitrary								0	
	USB1_DMA1 (channel 1 transmit FIFO empty)	Arbitrary	D1FIFO_1 D1FIFOBn_1 (n = 0 to 7)	010_0011	11						1	
	USB1_DMA1 (channel 1 receive FIFO full)	D1FIFO_1 D1FIFOBn_1 (n = 0 to 7)	Arbitrary								0	
A/D converter	ADI (A/D conversion end)	ADDR	Arbitrary	010_0100	11	0/1	001	1			0	
IEBus™ controller (RZ/A1L only)	IEBBTD (data interrupt) (transmission in single mode)	Arbitrary	IEBB0DR	010_1000	11	0	010	0				
	IEBBTD (data interrupt) (reception in single mode)	IEBB0DR	Arbitrary									
	IEBBTD (data interrupt) (transmission in FIFO mode)	Arbitrary	IEBB0DR									
	IEBBTV (vector interrupt) (reception in FIFO mode)	IEBB0DR	Arbitrary	010_1001	11							
CD-ROM decoder (RZ/A1L only)	IREADY (decoding end)	STRMDOUT 0	Arbitrary	010_1010	11		010	1			0	
SD host interface 0	SDHI_0 transmission	Arbitrary	Data register	011_0000	01		010	1			1	
	SDHI_0 reception	Data register	Arbitrary		10						0	
SD host interface 1	SDHI_1 transmission	Arbitrary	Data register	011_0001	01						1	
	SDHI_1 reception	Data register	Arbitrary		10						0	

chanhvinh.com

Table 9.4 On-Chip Peripheral Module Requests

DMA Transfer Request Source	DMA Transfer Request Signal	Transfer Source	Transfer Destination	DMARS		CHCFG_n						
				MID	RID	TM	AM [2:0]	LVL	HIEN	LO EN	RE QD	SEL[2:0]
MMC host interface	Transmit data empty	Arbitrary	Data register	011_0010	01	0	010	1	1	0	1	Ch0: 000 Ch1: 001 Ch2: 010 Ch3: 011 Ch4: 100 Ch5: 101 Ch6: 110 Ch7: 111 Ch8: 000 Ch9: 001 Ch10: 010 Ch11: 011 Ch12: 100 Ch13: 101 Ch14: 110 Ch15: 111
	Receive data full	Data register	Arbitrary		10						0	
Serial sound interface channel 0	SSITXI0 (transmit data empty)	Arbitrary	SSIFTDR_0	011_1000	01		010	1			1	
	SSIRXI0 (receive data full)	SSIFRDR_0	Arbitrary		10						0	
Serial sound interface channel 1	SSITXI1 (transmit data empty)	Arbitrary	SSIFTDR_1	011_1001	01						1	
	SSIRXI1 (receive data full)	SSIFRDR_1	Arbitrary		10						0	
Serial sound interface channel 2	SSIRTI2 (transmit data empty)	Arbitrary	SSIFTDR_2	011_1010	11						1	
	SSIRTI2 (receive data full)	SSIFRDR_2	Arbitrary								0	
Serial sound interface channel 3	SSITXI3 (transmit data empty)	Arbitrary	SSIFTDR_3	011_1011	01						1	
	SSIRXI3 (receive data full)	SSIFRDR_3	Arbitrary		10						0	
SCUX	SCUTXI0 (FFD0_0 request)	Arbitrary	DMATD0_CIM	100_0000	01		001	1			1	
	SCURXI0 (FFU0_0 request)	DMATU0_CIM	Arbitrary		10						0	
	SCUTXI1 (FFD0_1 request)	Arbitrary	DMATD1_CIM	100_0001	01						1	
	SCURXI1 (FFU0_1 request)	DMATU1_CIM	Arbitrary		10						0	
	SCUTXI2 (FFD0_2 request)	Arbitrary	DMATD2_CIM	100_0010	01						1	
	SCURXI2 (FFU0_2 request)	DMATU2_CIM	Arbitrary		10						0	
	SCUTXI3 (FFD0_3 request)	Arbitrary	DMATD3_CIM	100_0011	01						1	
	SCURXI3 (FFU0_3 request)	DMATU3_CIM	Arbitrary		10						0	
Renesas serial peripheral interface channel 0	SPTI0 (transmit data empty)	Arbitrary	SPDR_0	100_1000	01		010	1			1	
	SPRI0 (receive data full)	SPDR_0	Arbitrary		10						0	
Renesas serial peripheral interface channel 1	SPTI1 (transmit data empty)	Arbitrary	SPDR_1	100_1001	01						1	
	SPRI1 (receive data full)	SPDR_1	Arbitrary		10						0	
Renesas serial peripheral interface channel 2	SPTI2 (transmit data empty)	Arbitrary	SPDR_2	100_1010	01						1	
	SPRI2 (receive data full)	SPDR_2	Arbitrary		10						0	
Renesas SPDIF interface	SPDIFTXI	Arbitrary	TDAD	101_0000	01		010	1			1	
	SPDIFRXI	RDAD	Arbitrary		10						0	
Media local bus (RZ/A1L only)	MLB_CINT (MLB channel write)	Arbitrary	Local Channel buffer	101_0011	11		010	1			1	
	MLB_CINT (MLB channel read)	Local Channel buffer	Arbitrary								0	

chanhvinh.com

Table 9.4 On-Chip Peripheral Module Requests

DMA Transfer Request Source	DMA Transfer Request Signal	Transfer Source	Transfer Destination	DMARS		CHCFG_n						
				MID	RID	TM	AM [2:0]	LVL	HIEN	LO EN	RE QD	SEL[2:0]
Serial communication interface channel 0	TXI0	Arbitrary	TDR0	101_1010	01	0	010	0	1	0	1	Ch0: 000 Ch1: 001 Ch2: 010 Ch3: 011 Ch4: 100 Ch5: 101 Ch6: 110 Ch7: 111 Ch8: 000 Ch9: 001 Ch10: 010 Ch11: 011 Ch12: 100 Ch13: 101 Ch14: 110 Ch15: 111
	RXI0	RDR0	Arbitrary		10						0	
Serial communication interface channel 1	TXI1	Arbitrary	TDR1	101_1011	01		010	0			1	
	RXI1	RDR1	Arbitrary		10						0	
I²C bus interface channel 0	INTRIICTI0 (transmit data empty)	Arbitrary	RIIC0DRT	110_0000	01		010	0			1	
	INTRIICRI0 (receive data full)	RIIC0DRR	Arbitrary		10						0	
I²C bus interface channel 1	INTRIICTI1 (transmit data empty)	Arbitrary	RIIC1DRT	110_0001	01						1	
	INTRIICRI1 (receive data full)	RIIC1DRR	Arbitrary		10						0	
I²C bus interface channel 2	INTRIICTI2 (transmit data empty)	Arbitrary	RIIC2DRT	110_0010	01						1	
	INTRIICRI2 (receive data full)	RIIC2DRR	Arbitrary		10						0	
I²C bus interface channel 3	INTRIICTI3 (transmit data empty)	Arbitrary	RIIC3DRT	110_0011	01						1	
	INTRIICRI3 (receive data full)	RIIC3DRR	Arbitrary		10						0	
LIN interface channel 0 (RZ/A1L only)	LIN0_INT_T (LIN mode)	Arbitrary	RLN30LDBRm (m = 1 to 8)	110_1000	01	0	010	0	1	0	1	
	LIN0_INT_R (LIN mode)	RLN30LDBRm (m = 1 to 8)	Arbitrary		10						0	
External request	DREQ0	Arbitrary	Arbitrary	000_0000	11	0/1	001/ 010/ 100	001: Falling edge detection 010: Rising edge detection 101: Low level detection 110: High level detection			0/1	000

Note: • CHCFG_n setting value
TM 0: Single transfer
1: Block transfer
AM 001: ACK level output
010: ACK bus cycle output
100: No ACK
LVL 0: REQ edge detection
1: REQ level detection
REQD 0: ACK output at read
1: ACK output at write

chanhvinh.com

9.6 DMA Mode

9.6.1 Mode Setting

You can toggle between register mode and link mode, by using the DMS field of the CHCFG_n register.

Table 9.5 DMA Mode Setting

DMS (CHCFG_n)	Mode	Description
0	Register mode	A DMA transfer is executed using the values set in the Next Register Set.
1	Link mode	A DMA transfer is executed using the descriptor set in the Current register. The DMAC repeatedly loads the descriptor and executes the DMA transfer unless otherwise set by the descriptor or stopped by the control register.

9.6.2 Register Mode

In register mode, a DMA transfer is executed using the values set in the internal registers.
Two sets of the source address, destination address, and transfer byte count (Next0 Register Set and Next1 Register Set) can be set. It is possible to select the Next Register Set to be used for the DMA transfer, as well as to execute two Next Register Sets continuously for the DMA transfer.

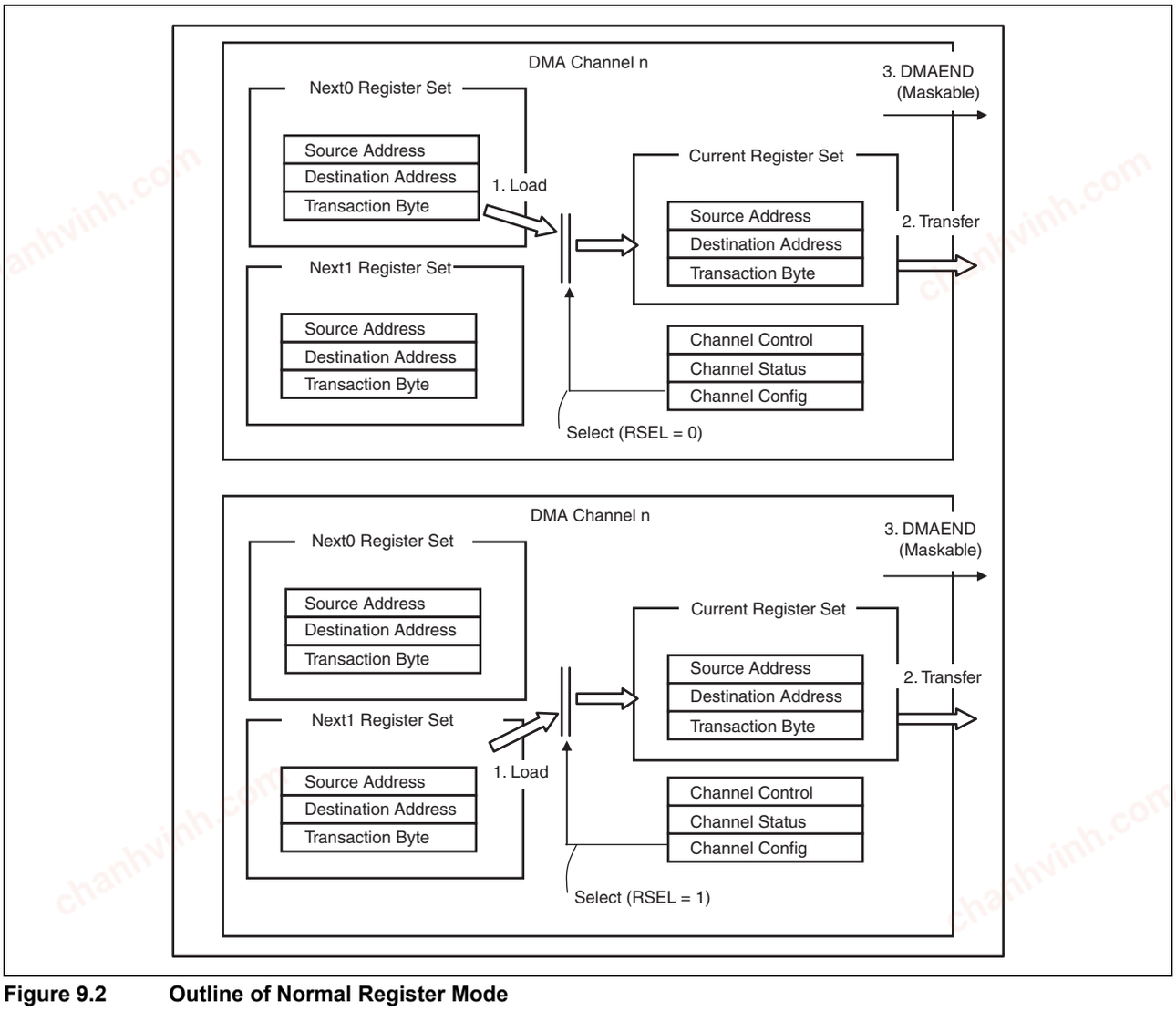


Figure 9.2 Outline of Normal Register Mode

chanhvinh.com

The above figure shows how the transfer is executed when the Next0 Register Set is used (upper part of the figure) and when the Next1 Register Set is used (lower part of the figure).

(1) Operation Flow

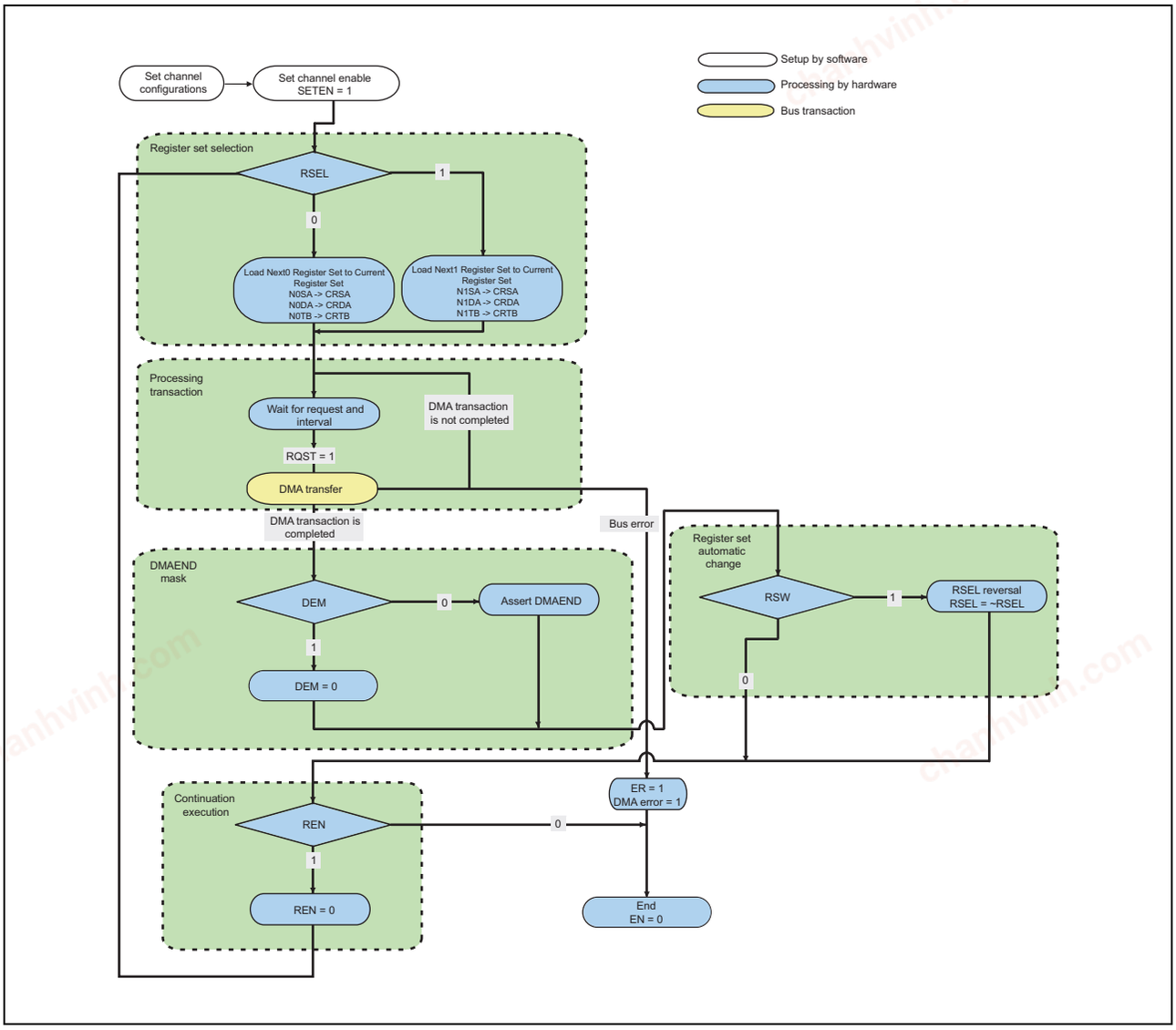


Figure 9.3 Register Mode Flow

chanhvinh.com

<Explanation of the register mode flow>

1. Channel setting (set channel configuration)
   The Next0 or Next1 Register Set (destination address, source address, and total transfer byte count) is set.
   In the Channel Register Set, the DMA Register Set (REQ, DMAACK, transfer size, etc.) is set. (See section 9.7, DMA Transfer.)

2. Register set selection (register set selection)
   When 1 is set in EN, the values set in the Next Register Set selected by RSEL are loaded to the Current Register Set.

3. DMA transaction (processing transaction)
   A DMA transfer is executed according to the set values. For details of the transfer, see section 9.7, DMA Transfer.

4. DMA transfer end interrupt mask (DMAINT mask)
   The DMA transfer end interrupt is masked according to the value set in the DEM bit of CHCFG_n. When 1 is set in DEM, the DMA transfer end interrupt is not output. Also, immediately after that, DEM is automatically cleared to 0.

5. Automatic register set change (register set automatic change)
   Whether to use the other Next Register Set is determined by the value set in the RSW bit of CHCFG_n.

6. Continuation of execution (continuation execution)
   Whether to continue the execution of the DMA transfer is determined by the value set in the REN bit of CHCFG_n. When 1 is set in REN, the execution of the DMA transfer is continued. Also, immediately after that, REN is automatically cleared to 0.

chanhvinh.com

(2) Register Setting

(a) Register mode setting

Select the register set to be executed.

Table 9.6 Register Mode Setting

DMS (CHCFG_n)	RSEL (CHCFG_n)	Description
0	0	Executes the Next0 Register Set.
	1	Executes the Next1 Register Set.

(b) DMA transfer end interrupt mask setting

The DMA transfer end interrupt can be masked individually for each register set.

Table 9.7 DMAINT Mask Setting

DEM (CHCFG_n)	Operation	Remark
0	When the DMA transaction is completed, a DMA transfer end interrupt is issued.	
1	Even when the DMA transaction is completed, a DMA transfer end interrupt is not issued. After the DMA transaction is completed, DEM is cleared to 0 by hardware.	

(c) Automatic register set execution setting

After DMA transfers, the DMA transaction of the selected register set is automatically executed.

Table 9.8 Automatic Register Set Execution Setting

REN (CHCFG_n)	Operation	Remark
0	When the DMA transaction of the register set selected by RSEL is completed, the EN bit is cleared and the DMA operation ends.	Set this value when you want to execute a DMA transaction once.
1	When a DMA transaction is completed, the DMAC continues to execute a DMA transfer by using the data set in the selected register set. When continuous transfers are successful, REN is cleared to 0.	Set this value when you want to continuously execute DMA transfers by using the data set in separate register sets.

chanhvinh.com

(d) Automatic register set change setting

When 1 is set in REN, the DMAC can automatically change to the register set to be executed next, after a DMA transaction is completed.

Table 9.9 Automatic Register Set Change Setting

RSW (CHCFG_n)	Operation	Remark
0	If 1 is set in REN when a DMA transaction is completed, the register set is not changed.	Set this value when you want to use only one register set.
1	If 1 is set in REN when a DMA transaction is completed, the value of RSEL is automatically inverted and the other register set is selected.	Set this value when you want to change the register set.

(3) Setting Examples

(a) When only the Next0 Register Set is used

Table 9.10 Register Mode Setting Example 1

DMS (CHCFG_n)	RSEL (CHCFG_n)	DEM (CHCFG_n)	RSW (CHCFG_n)	REN (CHCFG_n)
0 (Register mode)	0 (Next0)	0 (not masked)	0 (not switched)	0 (not continuously executed)

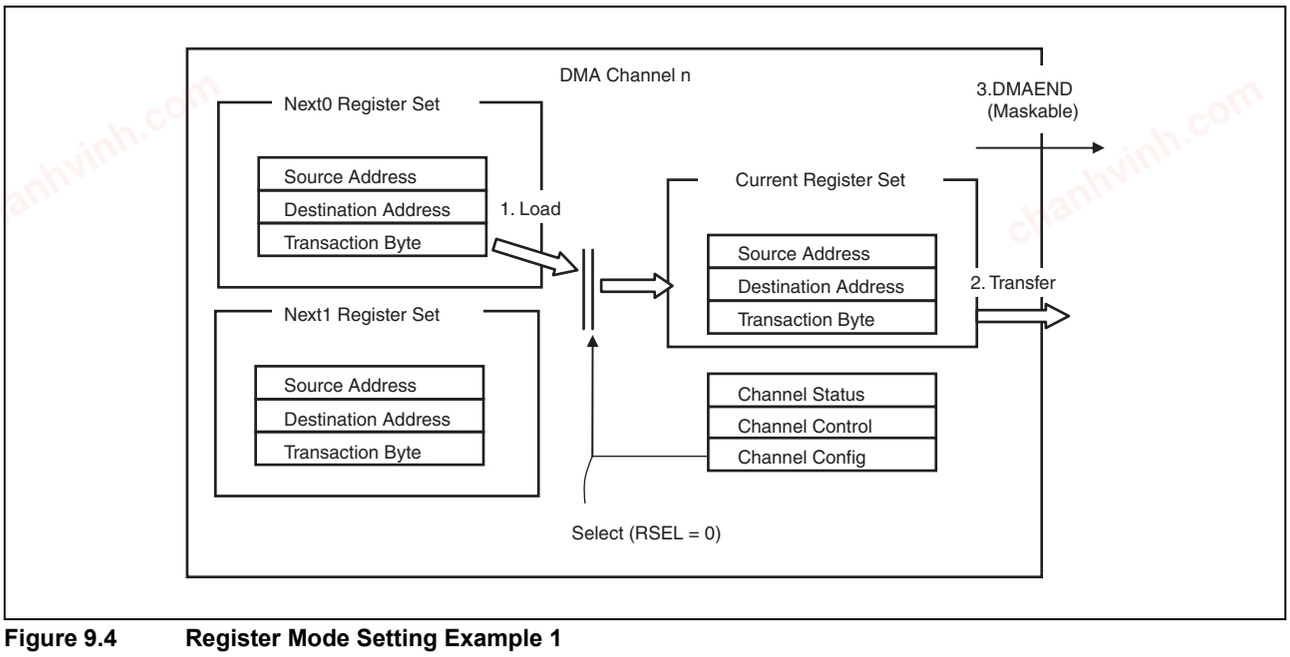


Figure 9.4 Register Mode Setting Example 1

1. 1 is set in EN (SETEN = 1), and the Next0 Register Set is loaded to the Current Register Set.
2. A DMA transaction is executed according to the values set in the Current Register Set and Channel Register Set.
3. Because 0 is set in DEM, the DMA transfer end interrupt is issued after the DMA transaction is completed.
4. Because 0 is set in REN, EN is cleared to 0 and the DMA transaction ends.

chanhvinh.com

(b) When two register sets are used continuously

Table 9.11 Automatic Register Set Execution Setting

DMS (CHCFG_n)	RSEL (CHCFG_n)	DEM (CHCFG_n)	RSW (CHCFG_n)	REN (CHCFG_n)
0 (Register mode)	0 (Next0)	1 (masked)	1 (switched)	1 (continuously executed)

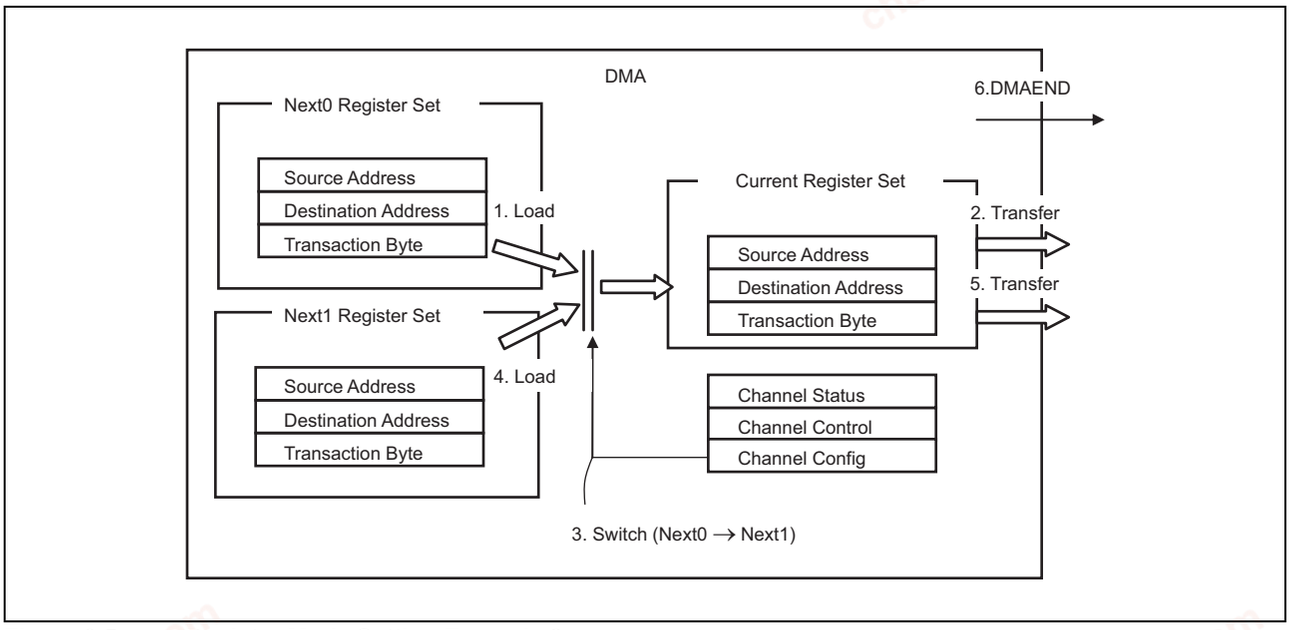


Figure 9.5 Register Mode Setting Example 2

1. 1 is set in EN (SETEN = 1), and the Next0 Register Set is loaded to the Current Register Set.
2. A DMA transaction is executed according to the values set in the Current Register Set and Channel Register Set.
3. Because 1 is set in DEM, DMA transfer end interrupt is not output after the DMA transaction is completed. Also, DEM is automatically cleared to 0.
4. Because 1 is set in REN, the execution is continued. Also, REN is automatically cleared to 0.
5. Because 1 is set in RSW, the register set to be executed next is switched (RSEL = 0 → 1).
6. The Next1 Register Set is loaded to the Current Register Set.
7. A DMA transaction is executed according to the values set in the Current Register Set and Channel Register Set.
8. Because 0 is set in DEM, the DMA transfer end interrupt is issued after the DMA transaction is completed.
9. Because 0 is set in REN, EN is cleared to 0 and the DMA transaction ends.

chanhvinh.com

9.6.3 Link Mode

In link mode, a descriptor stored in external memory is loaded as set values and a DMA transaction is executed using the loaded values. The DMAC contains a Next Link address and a Current Link address for each channel, and these addresses are used to set the descriptor address to be executed next and to indicate the descriptor address of the currently executed DMA transaction, respectively.

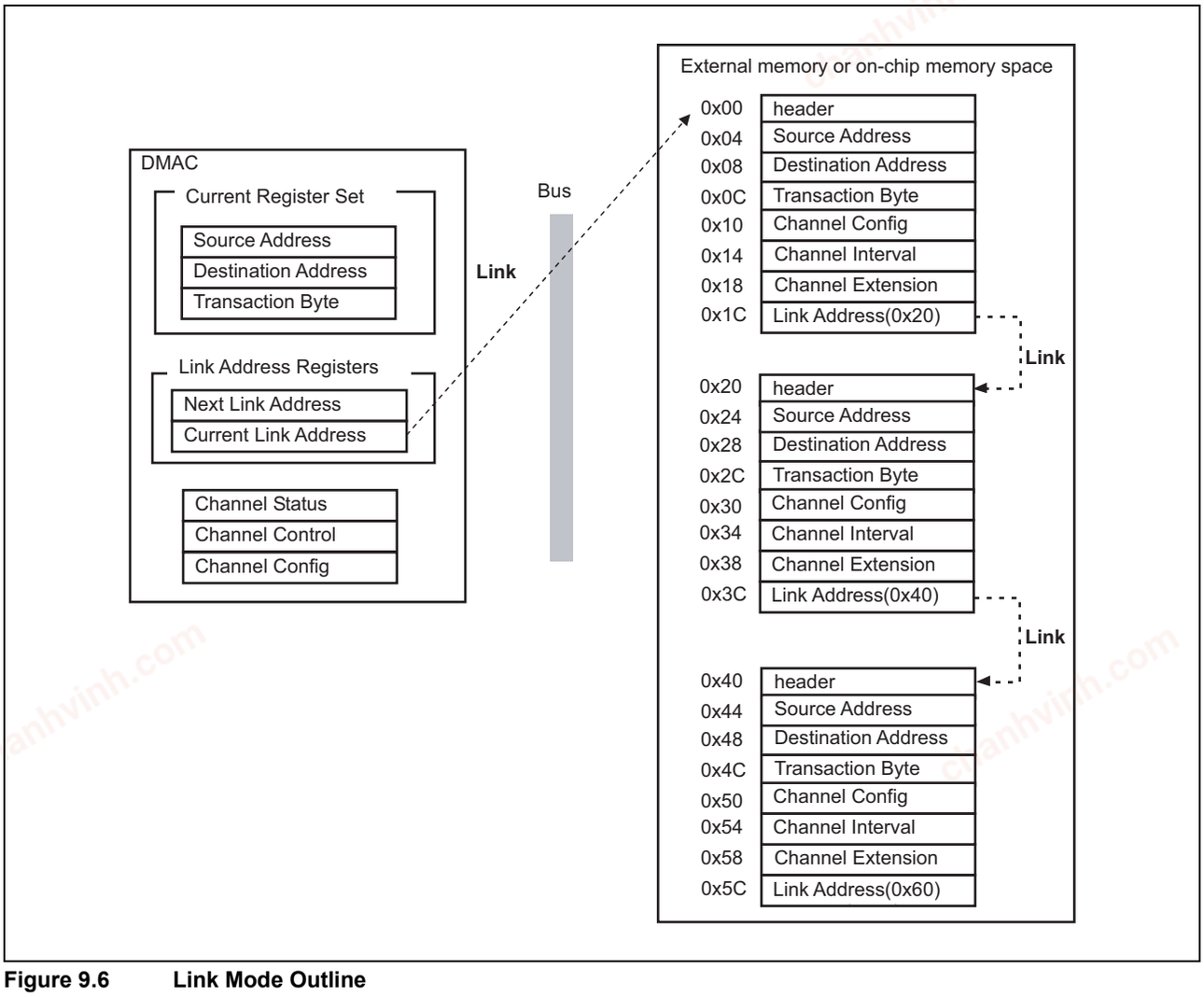


Figure 9.6 Link Mode Outline

chanhvinh.com

(1) Operation Flow

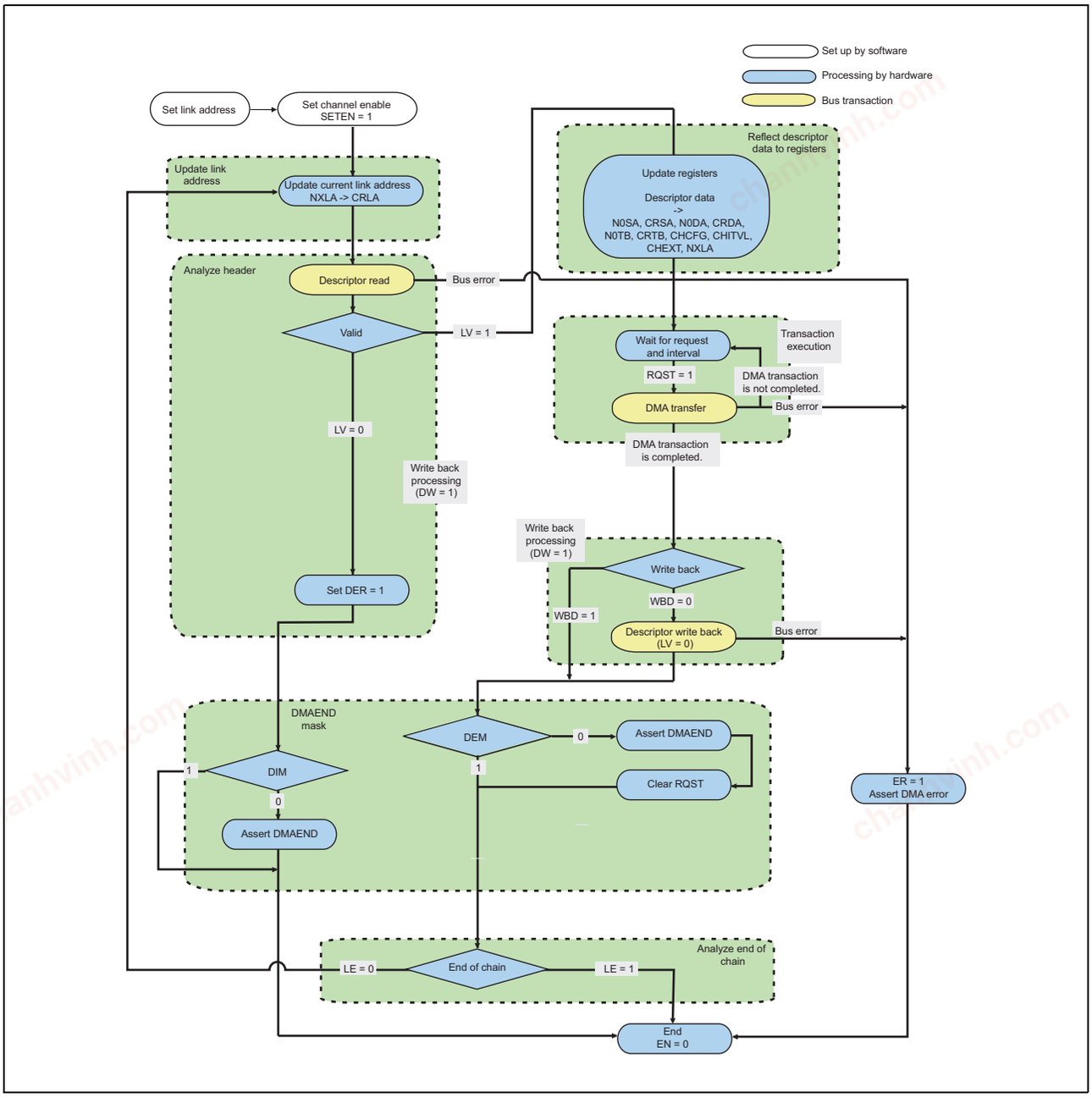


Figure 9.7 Link Mode Flow

chanhvinh.com

<Explanation of the link mode flow>

1. Channel setting
   The start address of the link destination is set in NXLA_n.

2. Link address update
   When 1 is set in EN (1 is set in SETEN), the Link address set in NXLA_n is loaded to CRLA_n.

3. Descriptor load and header analysis
   The DMAC begins to load the descriptor and then analyzes the content of the header. When LV is 0, the DMAC discards the loaded descriptor and sets 1 in DER to end the operation (EN = 0). In this case, if 0 is set in DIM of the header, DMAEND is issued.

4. Descriptor setting
   The loaded descriptor is set in the Current Register Set and Channel Register Set. Also, the next link address is set in NXLA_n.

5. DMA transaction
   A DMA transaction is executed according to the set values.

6. Header writeback
   When 0 is set in WBD of the header, the DMAC writes back the header with 0 set in its LV bit.

7. DMAINT mask
   When 0 is set in the DEM bit of CHCFG_n, the DMA transfer end interrupt is issued.

8. Link end analysis
   When 1 is set in LE of the header, the operation is ended by clearing EN to 0, after the transfer using the settings of the descriptor is completed. When 0 is set in LE, the Current register is updated and the load of the next descriptor begins. TEND is issued each time a descriptor is transferred.

chanhvinh.com

(2) Register Setting

(a) Link mode setting

To use the link mode, set 1 in the DMS bit of the CHCFG_n register.

Table 9.12 Link Mode Setting

DMS (CHCFG_n)	Description
1	Operates in link mode. This bit cannot be changed using a descriptor.

(b) Link address setting

There are two registers that indicate a link address: Next Link address register and Current Link address register.
To start the link mode, set a link address in the Next Link address register.
The Next Link address indicates the next link address after a descriptor is loaded. The Current Link address indicates the currently executed link address.

Table 9.13 Link Address Register Set

Register	Description
Next Link Address Register (NXLA_n)	Sets and indicates the next link address. Before starting the link mode, set a link address in this register.
Current Link Address Register (CRLA_n)	Indicates the currently executed link address. This register is read-only.

<Caution> In link mode, the settings can be changed by reading a descriptor. It is not possible, however, to synchronize the change of the settings with a peripheral module request or external request. Therefore, when using a peripheral module request or external request, set AM, LVL, HIEN, LOEN, and SEL of the CHCFG_n register before setting Enable and do not change any of these bits in the descriptor.

(3) Descriptor Setting

In a link address, prepare a descriptor with data arranged in the order shown below.
The DMAC reads the descriptor in burst mode.

(a) Descriptor data arrangement

Table 9.14 Descriptor Data Arrangement

Address	Data	Remark
Link address + 00H	header	
Link address + 04H	Source Address	
Link address + 08H	Destination Address	
Link address + 0CH	Transaction Byte	
Link address + 10H	Config	The register mode cannot be set.
Link address + 14H	Interval	
Link address + 18H	Extension	
Link address + 1CH	Next Link Address	

Remark: As a link address, set an address aligned along the 32-bit boundary.

chanhvinh.com

(b) header

The header indicates the status of the descriptor, as shown below.
The DMAC reads this area when a DMA transfer is started in link mode. Also, after a DMA transaction is completed, the DMAC writes back the transfer status to the area.

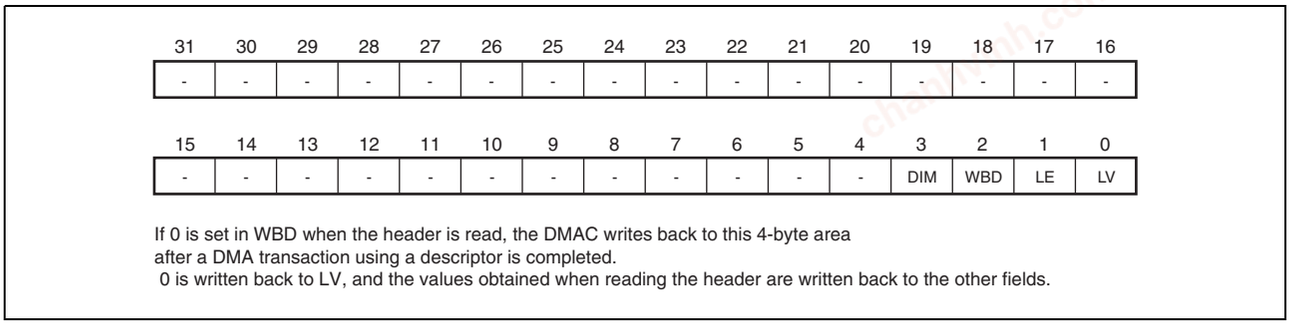


Figure 9.8 header Area

Table 9.15 Header Area

Bit Position	Bit Name	Meaning
31 to 4	—	—
3	DIM	Descriptor Interrupt Mask Sets whether to mask the DMA transfer end interrupt if 0 is set in LV when the header is loaded. 0: Issues a DMA transfer end interrupt. 1: Does not issue a DMA transfer end interrupt.
2	WBD	Write Back Disable Sets whether to mask LV bit writeback. When 1 is set in this bit, the DMAC does not perform writeback. 0: Writes the LV bit back to 0. 1: Does not write back the LV bit.
1	LE	Link End Indicates whether the link ends with the DMA transaction of this descriptor. Set 1 in this bit to indicate the end of the link. 0: The link continues. 1: The link ends.
0	LV	Link Valid Indicates whether this descriptor is valid. If 0 is set in WBD, the DMAC writes 0 in this bit after the DMA transaction written in the descriptor is executed. When setting the header, set 1 in this bit. 0: Descriptor invalid 1: Descriptor valid

(c) Descriptor data other than the header

The data items of the descriptor other than the header are the same as defined in the internal register specifications (note that the DMS bit of the CHCFG_n register cannot be changed using the descriptor). For information about the internal register specifications, see section 9.4, Register Descriptions.
For descriptor setting examples, see section 9.8, DMA Setting Examples.

(d) CACHE settings for descriptor access

The CACHE settings for descriptor access can be set in LWCA and LDCA of the DMA control register (DCTRL_0_7, DCTRL_8_15). Make these settings as appropriate for the access destination in which the descriptor is prepared.

chanhvinh.com

(e) Descriptor area and DMA transfer area

The following figure outlines the descriptor area and DMA transfer area that are accessed by the DMAC.

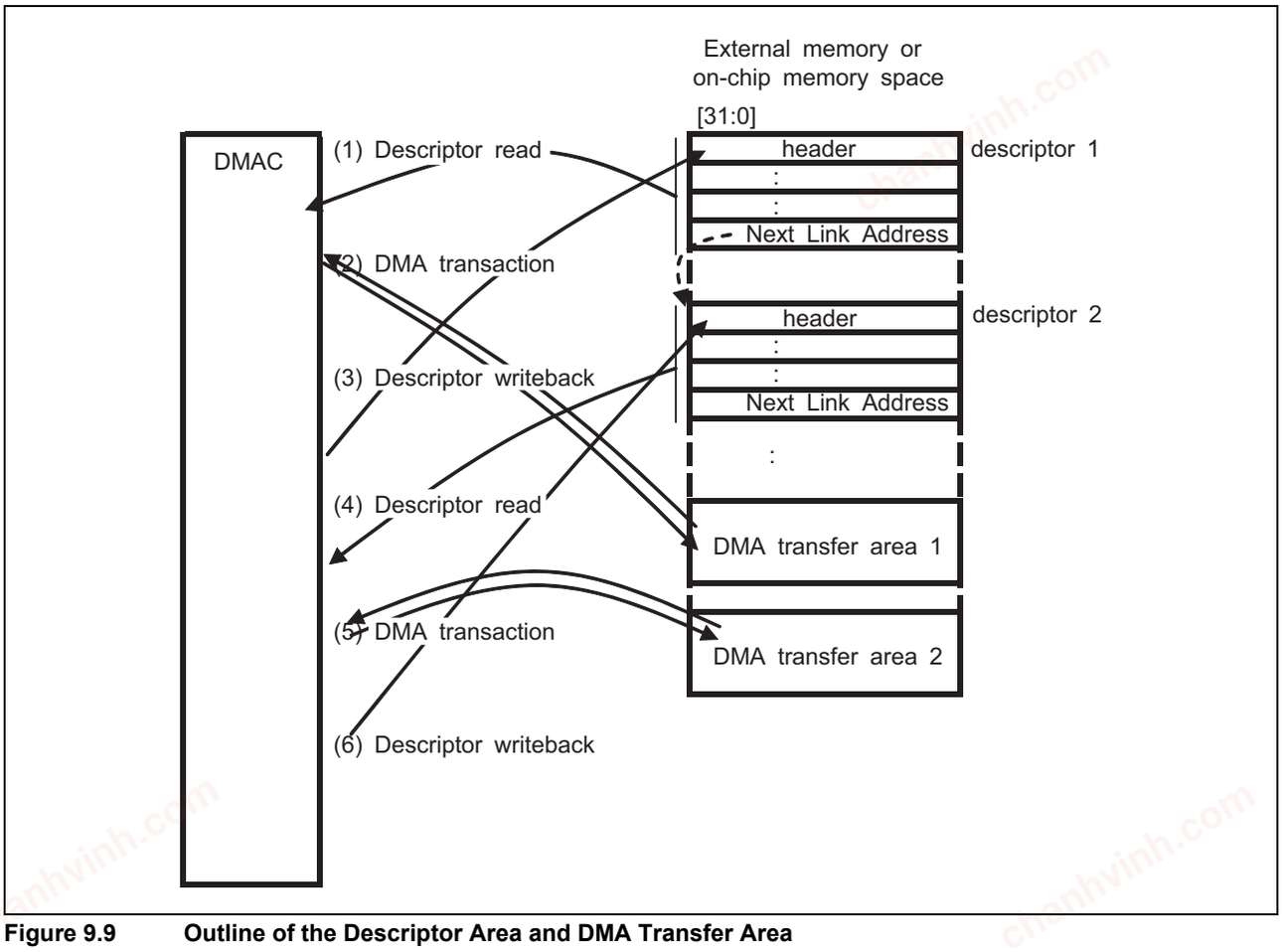


Figure 9.9 Outline of the Descriptor Area and DMA Transfer Area

1. Descriptor read
   The values set in the internal Next Link Address register are loaded to the Current Link Address register, and a descriptor is read from the external memory space (descriptor1) pointed to by the Current Link Address register.
2. DMA transfer
   When 1 is set in the LV bit of the header in the descriptor, a DMA transfer is executed according to the descriptor data.
3. Descriptor writeback
   When 0 is set in the WBD bit of the header after the DMA transfer of the set number of bytes is completed, the DMAC writes back data in word size to the header of descriptor1, with 0 set in LV and the other bits containing the values read in <1>.
4. Descriptor read
   When 0 is set in the LE bit of the header in the last read descriptor (<1>), the next descriptor is read from the address (descriptor2) indicated by Next Link Address in the descriptor.
5. DMA transfer
   When 1 is set in the LV bit of the header in the descriptor, a DMA transfer is executed according to the descriptor data.
6. Descriptor writeback
   When 0 is set in the WBD bit of the header after the DMA transfer of the set number of bytes is completed, the DMAC writes back data in word size to the header of descriptor2, with 0 set in LV and the other bits containing the values read in <4>.

chanhvinh.com

4 through 6 are repeated.
When the header contains 1 in LE and 0 in WBD, the DMAC executes a DMA transfer using the settings of that descriptor, writes back data with 0 set in the LV bit of the header and ends the operation.
When the header contains 1 in both LE and WBD, the DMAC executes a DMA transfer using the settings of that descriptor and ends the operation (without writing back).
When the header contains 0 in LV, the DMAC ends the operation (without executing a DMA transfer).

(4) Descriptor Configuration Examples

In link mode, a descriptor can be configured as shown below.

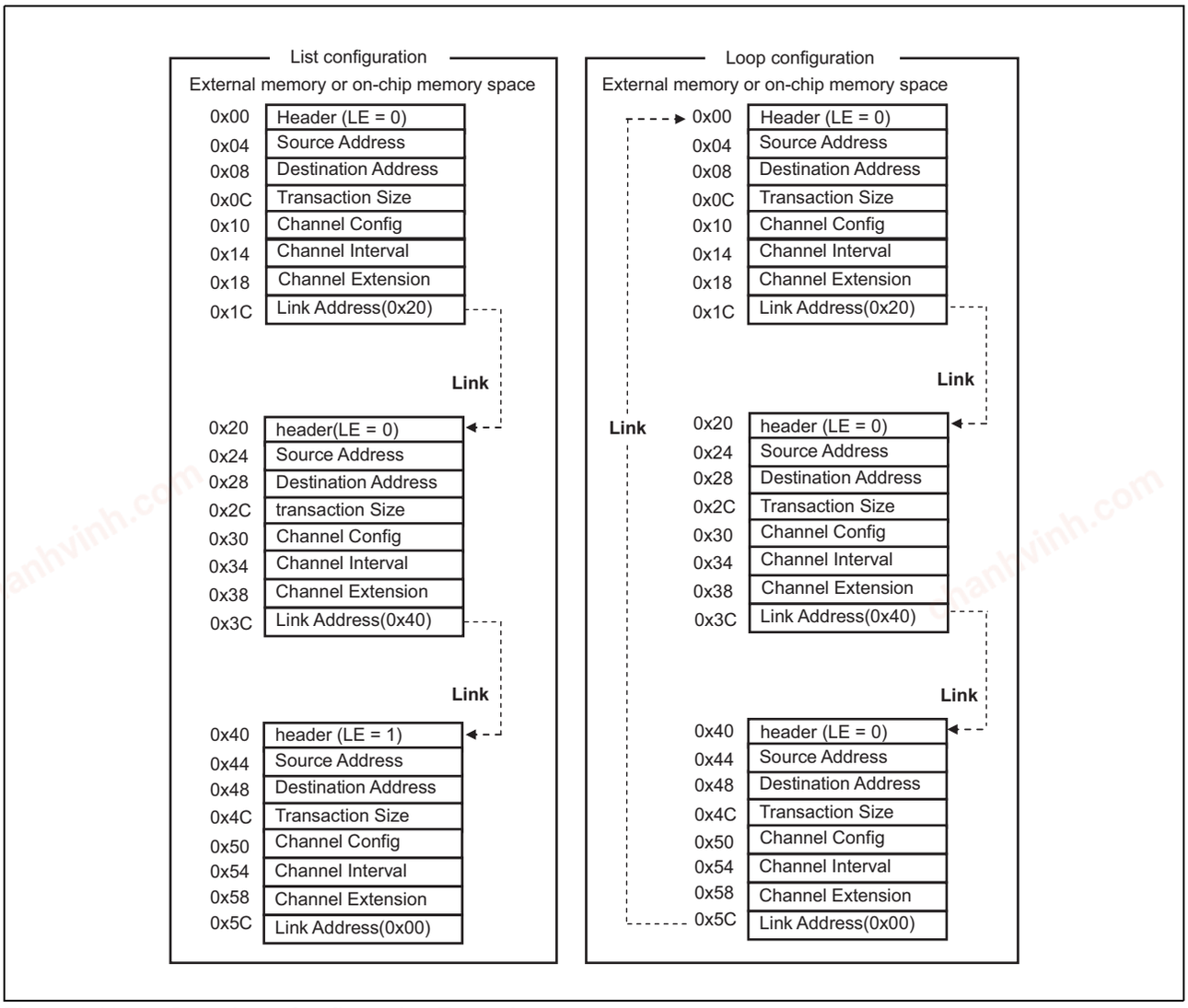


Figure 9.10 Descriptor Configuration Examples

- List configuration
  The link is ended by setting 1 in the LE bit of the header in the last descriptor.

- Loop configuration
  A descriptor can be created with a loop configuration, by setting the address of the top descriptor in the link address of the last descriptor. To end the loop, change the value of the LE bit of the header to 1 before the DMAC reads the descriptor, or follow the transfer suspension procedure.

chanhvinh.com

9.7 DMA Transfer

The basic operation of DMA transfer is described here.

9.7.1 Transfer Mode

Two transfer modes are supported: single transfer mode and block transfer mode.
To select a transfer mode, set the TM bit of CHCFG_n for each channel.

Table 9.16 Basic Transfer Setting

Transfer Mode	TM (CHCFG_n)	Function
Single transfer	0	A single DMA transfer is executed in response to a DMAREQ.
Block transfer	1	In response to a DMAREQ, the DMAC continues to execute the transfer until the DMA transaction is completed.

(1) Single Transfer Mode

When a DMA transfer request is received, a DMA transfer is executed once in the direction indicated by REQD (source or destination). A DMA transfer is executed once each time a transfer request is received, and this operation continues until the number of bytes loaded to CRTB_n is reached (arbitration between channels is accomplished for each DMA transfer).

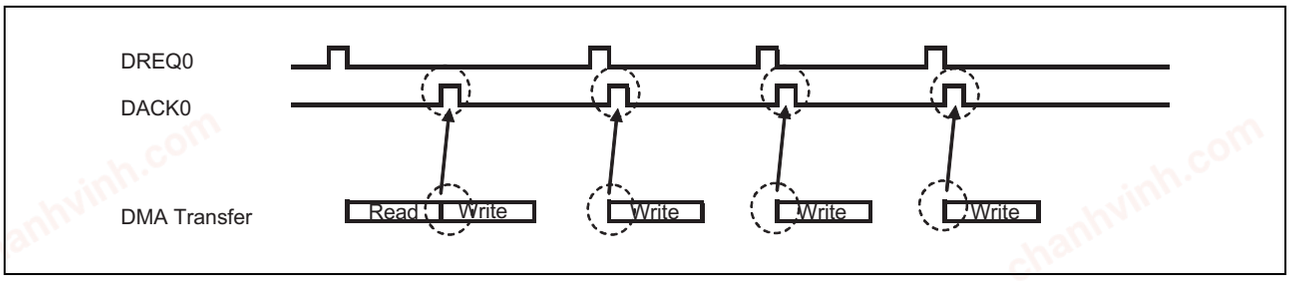


Figure 9.11 Single Transfer Mode (REQD = 1, SDS > DDS)

(2) Block Transfer Mode

Once a DMA transfer request is received, the DMAC continues to execute the transfer until data equivalent to the number of bytes loaded to the DMA transfer byte register (CRTB_n register) is transferred (the DMA transaction is completed) (arbitration between channels is accomplished for each DMA transfer).

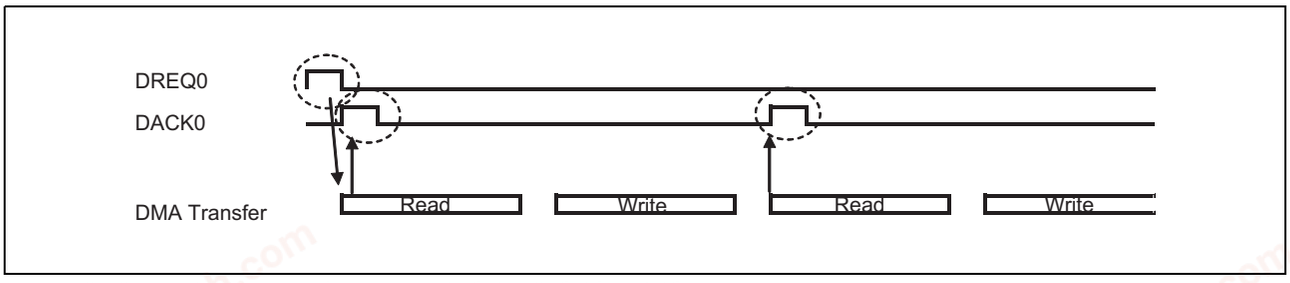


Figure 9.12 Block Transfer Mode (REQD = 0, SDS < DDS)

chanhvinh.com

9.7.2 Priority Control for DMA Channels

Within channels 0 to 7 and 8 to 15, two priority control modes are supported: fixed priority mode and round robin mode. Only round robin mode is supported for priority control between the group of channels 0 to 7 and the group of channels 8 to 15. To select a priority control mode, use the PR bit of the DMA control register (DCTRL register). The fixed priority mode is selected when 0 is set in the PR bit, and the round robin mode is selected when 1 is set.
Read priority and write priority are controlled independently.
The DMAC issues transfer requests to different channels concurrently without waiting for the completion of any particular transfer and processes responses in the order it receives them. Therefore, the order in which the channels start transactions is not necessarily consistent with the order in which the transactions end.

Table 9.17 Priority Control Setting

Mode	PR (DCTRL)	Function	Purpose
Fixed priority	0	Requests are controlled based on the fixed order of priority for channels 0 to 7 and 8 to 15 (High: CH0 (CH8) > CH1 (CH9) > CH2 (CH10) > CH3 (CH11) > CH4 (CH12) > CH5 (CH13) > CH6 (CH14) > CH7 (CH15): Low).	Use this mode when the channels have a specific order of priority.
Round robin	1	Requests are controlled in a round robin fashion.	Use this mode when you want requests evenly executed.

(1) Fixed Priority Mode

In fixed priority mode, the channels have a fixed order of priority in channels 0 to 7 and 8 to 15. Round robin mode is used to determine the priority between the group of channels 0 to 7 and the group of channels 8 to 15.
Immediately after a reset, the order of priority is as follows.
High CH0 > CH8 > CH1 > CH9 > CH2 > CH10 > CH3 > CH11 > CH4 > CH12 > CH5 > CH13 > CH6 > CH14 > CH7 > CH15 Low
If there is a transfer request from DMA channel 0 in this state, a transfer is executed on DMA channel 0. After the transfer is completed, the order of priority is as follows.
High CH8 > CH0 > CH9 > CH1 > CH10 > CH2 > CH11 > CH3 > CH12 > CH4 > CH13 > CH5 > CH14 > CH6 > CH15 > CH7 Low
If a DMA transfer request occurs on multiple channels simultaneously, the DMA transfer request of the channel having the smallest channel number is given priority. The following figure shows an example where a DMA transfer request occurs on a channel having a higher priority while a DMA transfer is being executed in fixed priority mode.

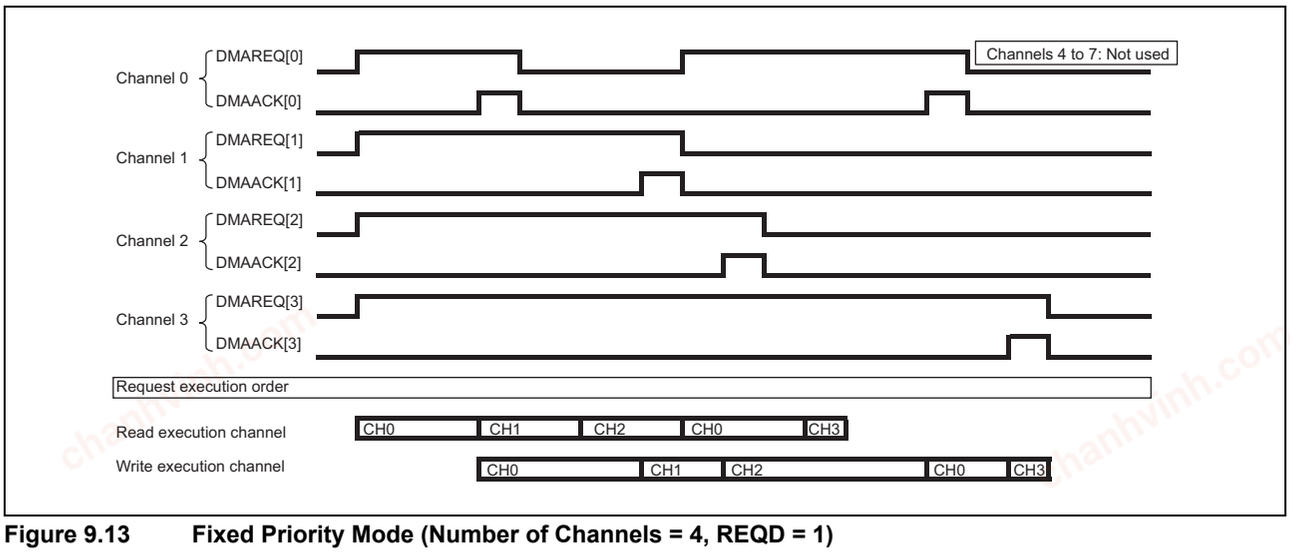


Figure 9.13 Fixed Priority Mode (Number of Channels = 4, REQD = 1)

chanhvinh.com

(2) Round Robin Mode

In round robin mode, each time a transfer request is received from a channel in the group of channels 0 to 7 and the group of channels 8 to 15, the order of priority is changed in such a way that the channel that executed a transfer last has the lowest priority.

Round robin mode is used to determine the priority between the group of channels 0 to 7 and the group of channels 8 to 15.

Immediately after a reset, the order of priority is the same as that of the fixed priority mode, which is as follows.

High CH0 > CH8 > CH1 > CH9 > CH2 > CH10 > CH3 > CH11 > CH4 > CH12 > CH5 > CH13 > CH6 > CH14 > CH7 > CH15 Low

If a transfer request is received from DMA channel 2 in this state, a transfer is executed on DMA channel 2. After the transfer is completed, the order of priority is as follows.

High CH8 > CH3 > CH9 > CH4 > CH10 > CH5 > CH11 > CH6 > CH12 > CH7 > CH13 > CH0 > CH14 > CH1 > CH15 > CH2 Low

The following figure shows an example where DMA transfers are executed in round robin mode.

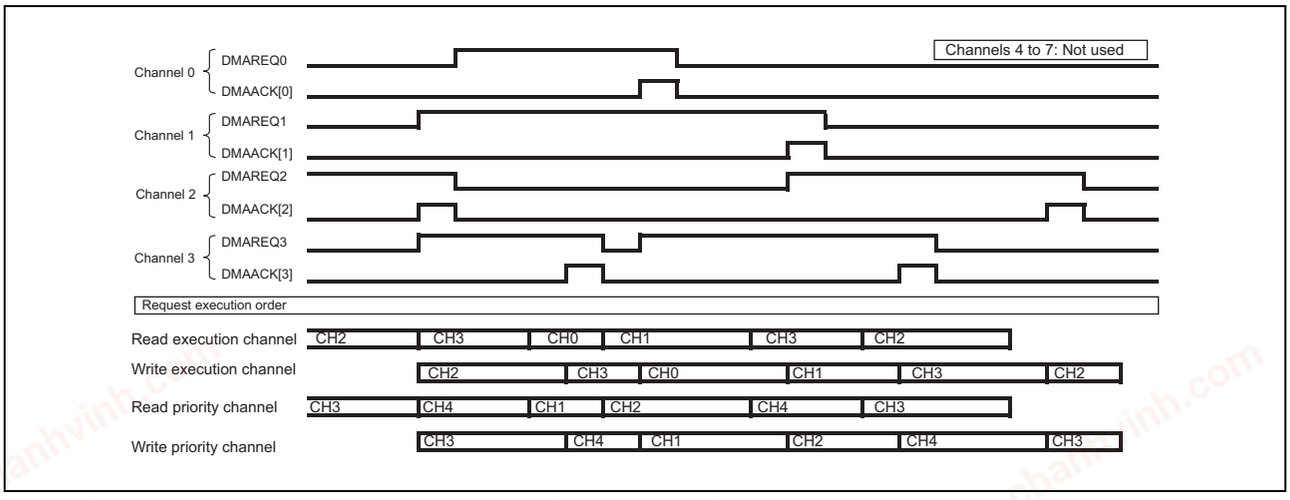


Figure 9.14 Round Robin Mode (Number of Channels = 4, REQD = 0)

The channel whose channel number is the number of the currently transferring channel + 1 gets to execute a DMA transfer next. If there is no transfer request from this channel, the channel whose channel number is the number of this channel + 1 gets to execute a DMA transfer.

chanhvinh.com

9.7.3 Number of States of an External Bus Cycle

When this module is the bus master, the number of states of an external bus cycle is controlled by the bus state controller as when the CPU is the bus master. For details, refer to section 8, Bus State Controller.

9.7.4 DMA Transfer Request

Edge detection or level detection can be selected using the LVL bit of the CHCFG_n register.
The HIEN and LOEN bits of the CHCFG_n register are used to select either the rising edge or falling edge in the case of edge detection or either the high level or low level in the case of level detection.
When the transfer request is by an on-chip peripheral module, set the CHCFG_n register according to Table 9.4, On-Chip Peripheral Module Requests.
When the transfer request is by the external pin (DREQ0), set the detection conditions (rising/falling edge and high/low level) according to Table 9.18, Setting for Detection of External Pin Request.

Table 9.18 Setting for Detection of External Pin Request

Mode	LVL (CHCFG_0)	HIEN (CHCFG_0)	LOEN (CHCFG_0)	Function
Edge detection	0	0	0	Specify this value when you use auto request triggers.
			1	Detects external pin request (DREQ0) at its falling edge.
		1	0	Detects external pin request (DREQ0) at its rising edge.
			1	Setting prohibited
Level detection	1	0	0	Setting prohibited
			1	Detects external pin request (DREQ0) in Low level mode.
		1	0	Detects external pin request (DREQ0) in High level mode.
			1	Setting prohibited

(1) Edge Detection

Setting 0 in the LVL bit of the CHCFG_n register enables edge detection.
When 1 is set in the HIEN bit of the CHCFG_n register, rising edge detection is enabled. When 1 is set in the LOEN bit, falling edge detection is enabled.
Wait for DACK0 to be detected, before issuing the next DREQ0 request.

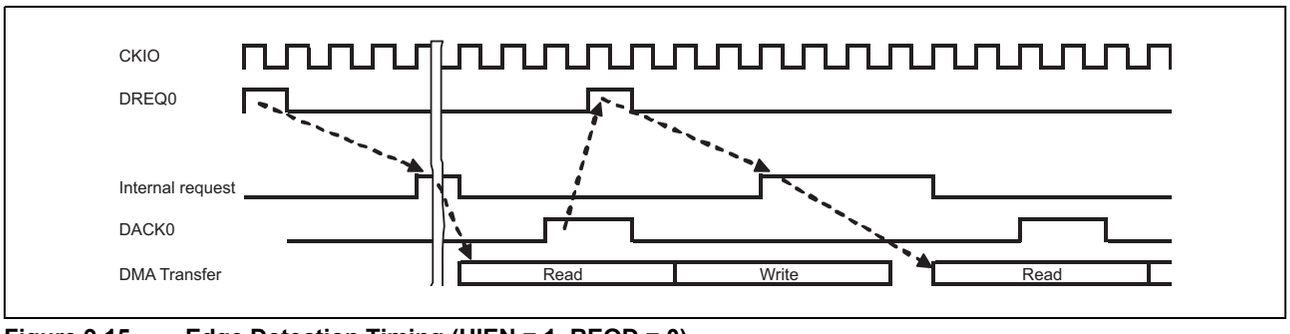


Figure 9.15 Edge Detection Timing (HIEN = 1, REQD = 0)

chanhvinh.com

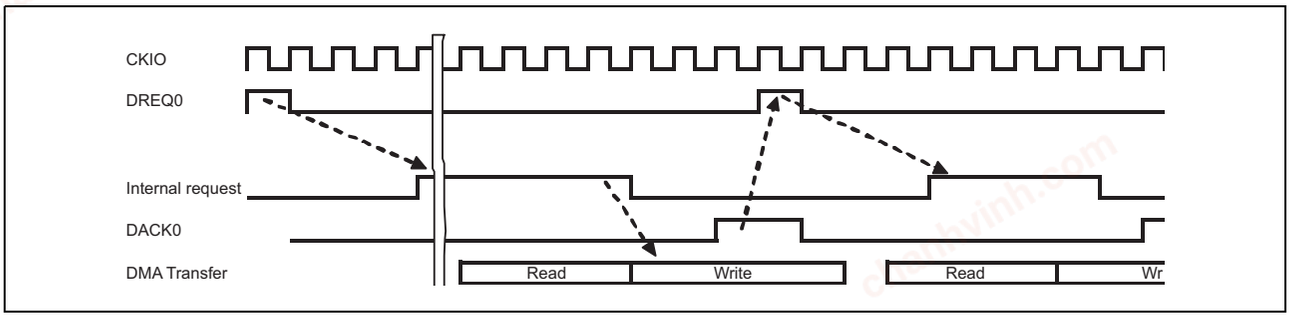


Figure 9.16 Edge Detection Timing (HIEN = 1, REQD = 1)

(2) Level Detection

Setting 1 in the LVL bit of the CHCFG_n register enables level detection.
DREQ0 is regarded as valid when it remains active for two consecutive clock cycles or more (depending on the HIEN and LOEN settings).
When the level mode is selected for DACK0, it remains at the High level until DREQ0 is deasserted.
When the next DMA transfer request is to be issued, DACK0 needs to be deasserted before that DREQ0 can be asserted.

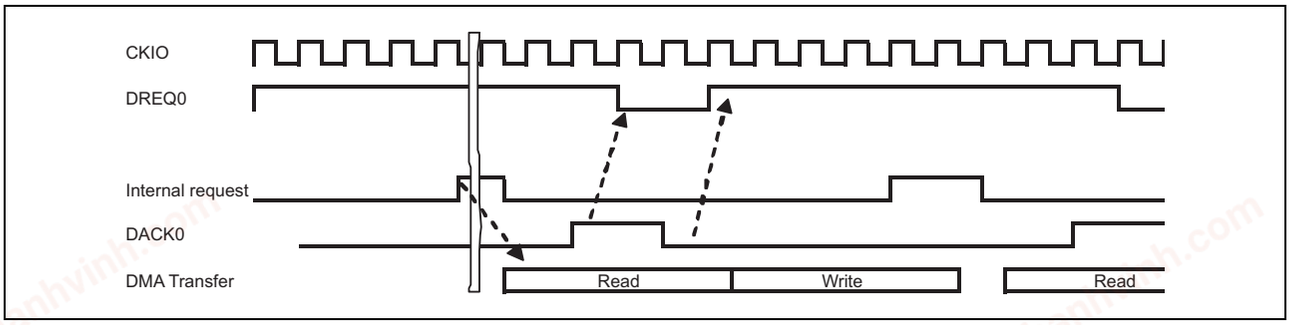


Figure 9.17 Level Detection Timing (HIEN = 1, REQD = 0, AM[2:0] = 001)

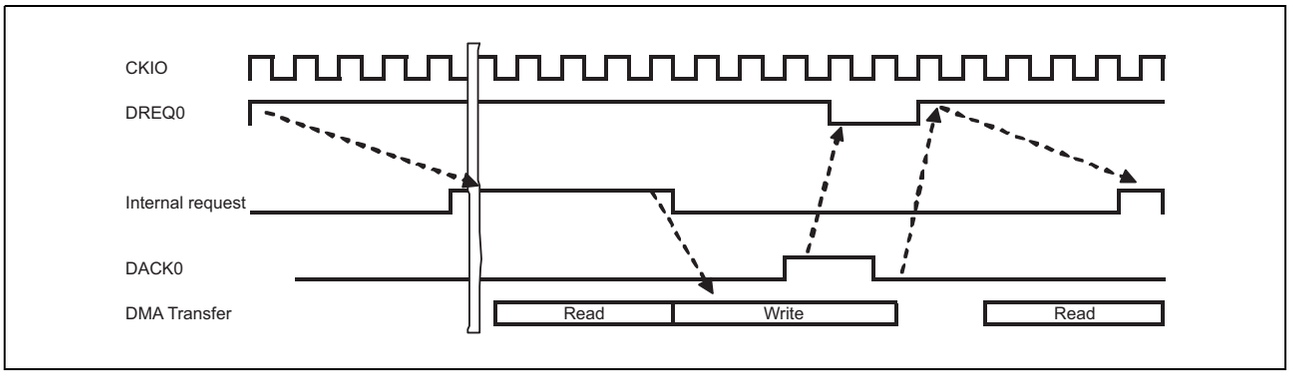


Figure 9.18 Level Detection Timing (HIEN = 1, REQD = 1, AM[2:0] = 001)

chanhvinh.com

9.7.5 DMA Acknowledge Output Function

DACK0 is an acknowledge signal that is sent to DREQ0. Level output and bus cycle output settings are supported as the DACK0 output mode. DACK0 is asserted at the same time as $\overline{CS}$ assertion except for the MPX-IO interface. For details, refer to section 8, Bus State Controller.

(1) DMA Acknowledge Signal Output Timing Setting

Upon receiving a DMA transfer request, the DACK0 pin becomes active (High level output). By using the REQD and AM[2:0] bits of the CHCFG_n register, the DACK0 output timing can be set as shown below.

Table 9.19 DACK0 Output Timing Setting

Mode	AM[2] (CHCFG_0)	AM[1:0] (CHCFG_0)	REQD (CHCFG_0)	Purpose
Pulse	0	00	0	Setting prohibited
			1	
Level	0	01	0 (Active during read)	DACK0 is output as a level. DACK0 remains asserted until DREQ0 is deasserted.
			1 (Active during write)	
Bus cycle	0	10 11	0 (Active during read)	DACK0 is output for the duration of a bus cycle. Use this mode when you want to keep DACK0 asserted until the end of the bus cycle.
			1 (Active during write)	
Mask	1	—	—	Make this setting when using auto request trigger.

(2) Level Output

Setting 001 in the AM bits of the CHCFG_n register enables level output. DACK0 remains asserted until DREQ0 is deasserted.

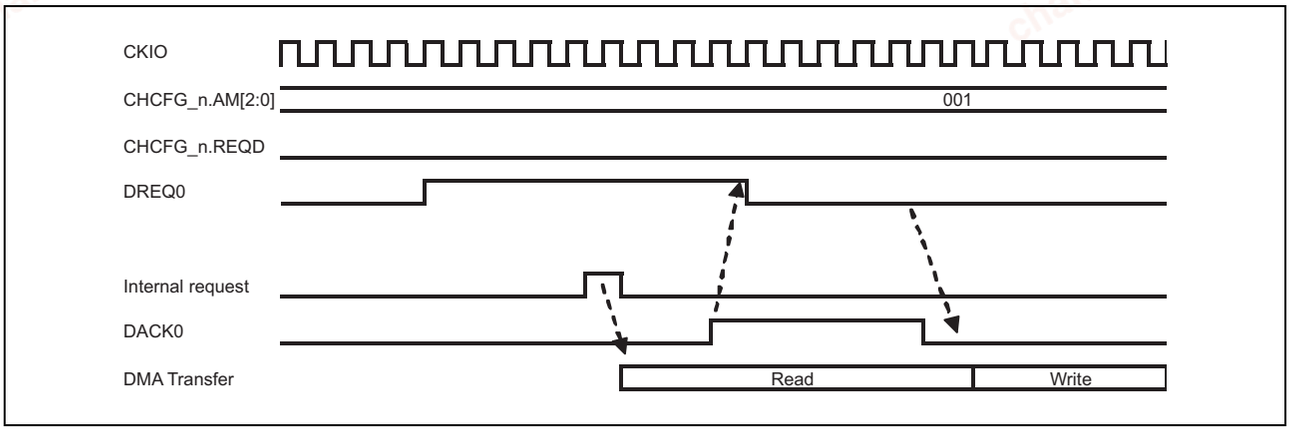


Figure 9.19 DACK0 Output Timing (AM[2:0] = 001, REQD = 0)

chanhvinh.com

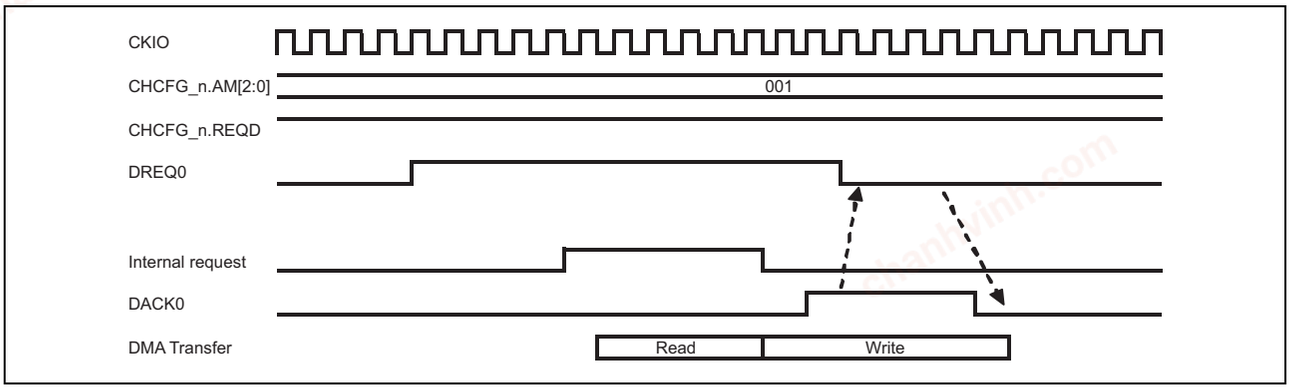


Figure 9.20 DACK0 Output Timing (AM[2:0] = 001, REQD = 1)

(3) Bus Cycle Output

Setting 010 in the AM bits of the CHCTRL_n register enables bus cycle output. DACK0 remains active for the duration of a bus cycle.

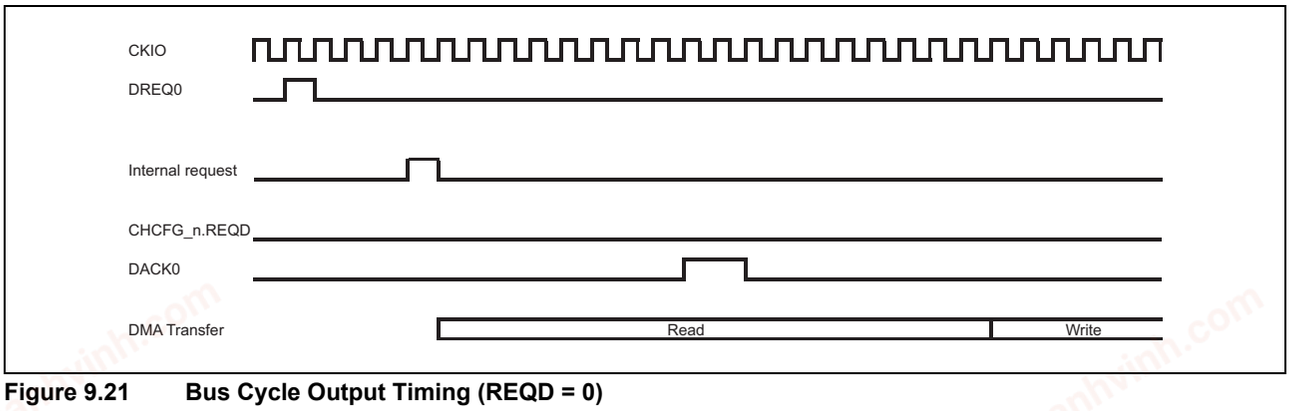


Figure 9.21 Bus Cycle Output Timing (REQD = 0)

- In the read active mode (REQD = 0), DACK0 remains active from the time when a read request is output on the bus until one cycle after the final read data.
- When level detection is selected for DREQ0, DREQ0 remains disabled until the cycle following the end of the bus cycle.

The following signals trigger the rise and fall of DACK0:

Rise: Transfer start (MARVALID = 1)

Fall: Transfer end (MRLAST & MRREADY = 1)

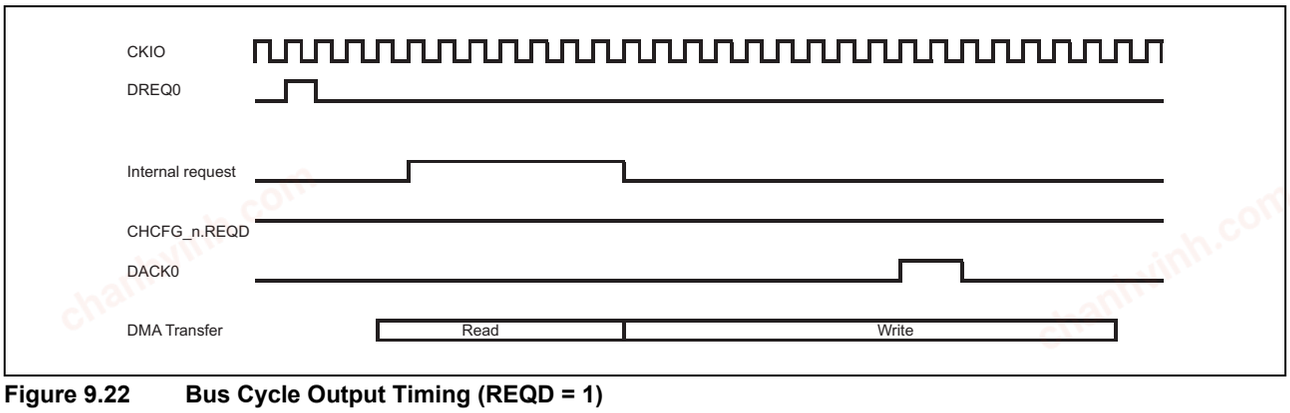


Figure 9.22 Bus Cycle Output Timing (REQD = 1)

chanhvinh.com

- In the write active mode (REQD = 1), DACK0 remains active from the time when a write request is output until one cycle after the response to the final data is returned.
- When level detection is selected for DREQ0, DREQ0 remains disabled until the cycle following the end of the bus cycle.

The following signals trigger the rise and fall of DACK0:
Rise: Transfer start (MAWVALID = 1)
Fall: Transfer end (MBVALID & MBREADY = 1)

9.7.6 DMA Transfer End Output Function

TEND0 is a transaction completion signal that is sent to the source of a DMA transfer request. TEND0 is asserted as the same time as DACK0 for the last transfer transaction. Figure 9.23 shows the TEND0 output timing.

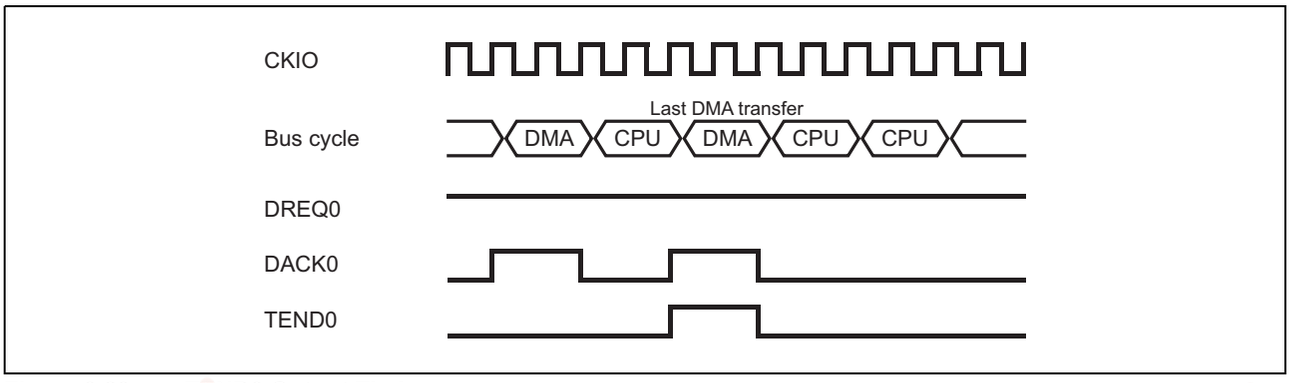


Figure 9.23 TEND0 Output Timing

9.7.7 DMA Transfer End Interrupt

The DMA transfer end interrupt is an interrupt request signal that indicates that a DMA transaction is completed.
There is an independent DMA transfer end interrupt for each channel.
When the transfer of data equivalent to the total transfer byte count loaded to the CRTB (Current Transaction Byte) is completed, 1 is set in END of the CHSTAT_n register. In this case, when 0 is set in DEM of the CHCFG_n register, the DMA transfer end interrupt is output (n = 0 to 15). (When writeback is performed in link mode, the signal is output after the writeback operation.)
When 0 is set in LV of the header in the read descriptor in link mode, 1 is set in DER of the CHSTAT_n register. In this case, when 0 is set in DIM of the header, the DMA transfer end interrupt is output.

Table 9.20 Assertion Conditions of DMA Transfer End Interrupt

Source	Condition	DMA Transfer End Interrupt Mask Signal
DMA transaction end	When the transfer of data equivalent to the total transfer byte count loaded to the CRTB (Current Transaction Byte) is completed with an OKAY response (or after the writeback operation when writeback is performed in link mode)	DEM bit of the CHCFG_n register
Descriptor invalid	When 0 is set in LV of the header in the read descriptor in link mode while 0 is set in DIM of the header	DIM bit of the header

chanhvinh.com

9.7.8 DMA Error Interrupt

If an error response is received for a DMA transfer or descriptor access, the DMAC regards it as an error and stops the transfer. Upon receiving an error response, the EN bit of the CHSTAT_n register of transferring channel n is cleared to 0 and 1 is set in the ER bit (n = 0 to 15). Also, the DMA error interrupt is output.
The DMA error interrupt cannot be masked.
Once an error occurs, the data of the whole transfer cannot be guaranteed. Be sure to start the transaction again from the beginning by following the procedure below.

1. Set 1 in the SWRST bit of the CHCTRL_n register.
2. Set each register again.

9.7.9 Interval Count Function

The interval at which a DMA transfer is executed can be adjusted by setting the ITVL bit of the channel interval register (CHITVL_n). This function is intended to prevent the DMA controller from occupying the bus all the time.
When a read or write operation is completed, a countdown starts from the value set in CHITVL_n. The next internal request is not executed until the count value reaches 0.
The following figure shows an example of how this works.

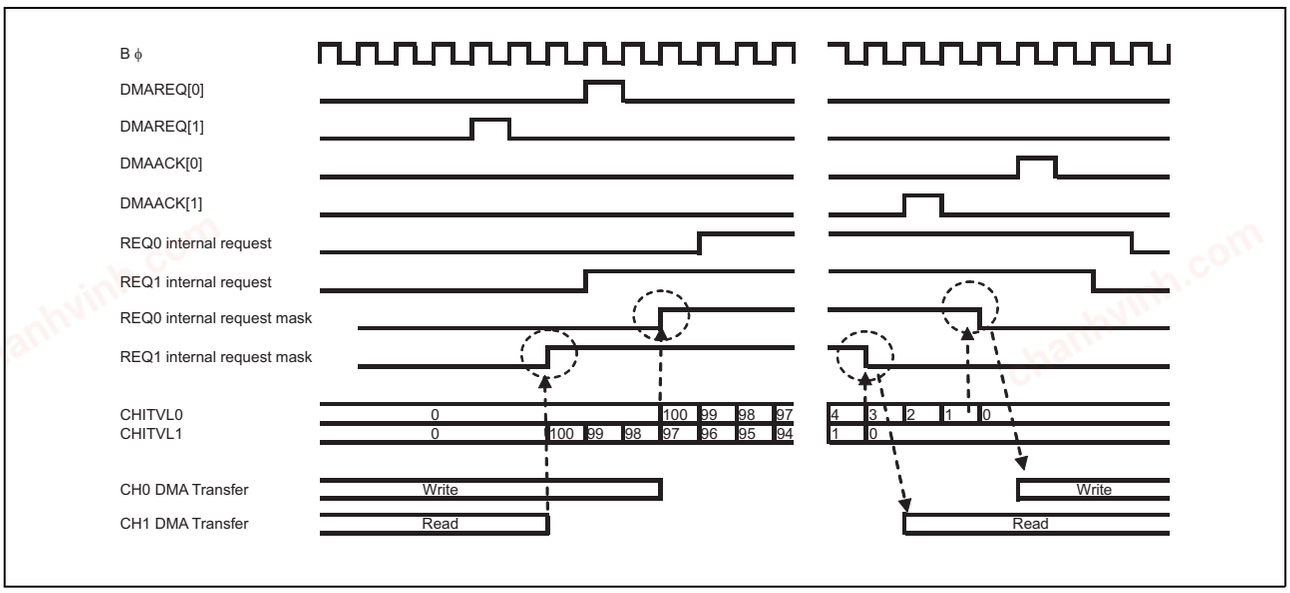


Figure 9.24 Interval Count

chanhvinh.com

9.7.10 Difference in Operation Due to the Transfer Size

(1) When the Source Transfer Size Is Smaller

When the read of data equivalent to the destination data size is completed, the data is written to the destination.
The following figure shows a timing chart where the source transfer size is 8 bits and the destination transfer size is 32 bits (in the case of rising edge detection).

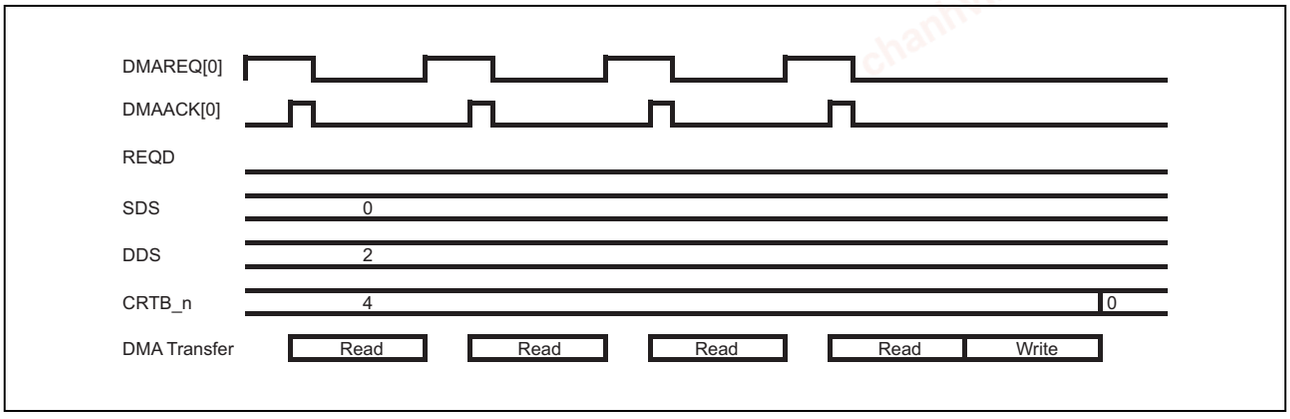


Figure 9.25 When the Source Transfer Size Is Smaller (LVL = 0, HIEN = 1, REQD = 0, SDS < DDS in CHCFG_n)

(2) When the Destination Transfer Size Is Smaller

Since the source transfer size is larger, multiple destination writes occur after a single source read. The following figure shows a timing chart where the source transfer size is 64 bits and the destination transfer size is 16 bits (in the case of rising edge detection) (1 is set in REQD of the CHCFG_n register).

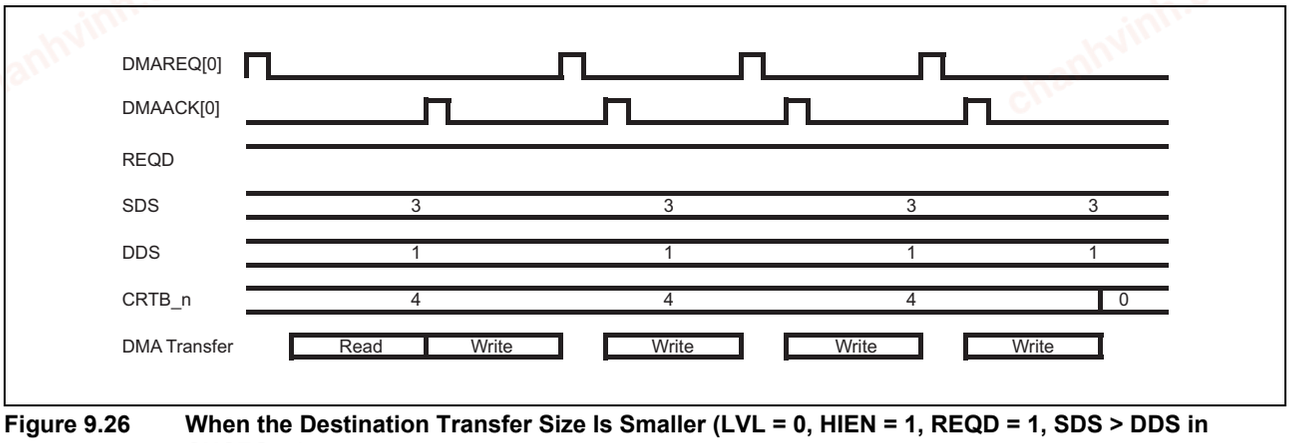


Figure 9.26 When the Destination Transfer Size Is Smaller (LVL = 0, HIEN = 1, REQD = 1, SDS > DDS in CHCFG_n)

chanhvinh.com

(3) When the Source Transfer Size Is the Same as the Destination Transfer Size

Every time a DMA transfer request is detected, a source read and a destination write occur.
The following figure shows a timing chart where the source transfer size and the destination transfer size are both 8 bits (in the case of rising edge detection, with 1 set in REQD of the CHCFG_n register).

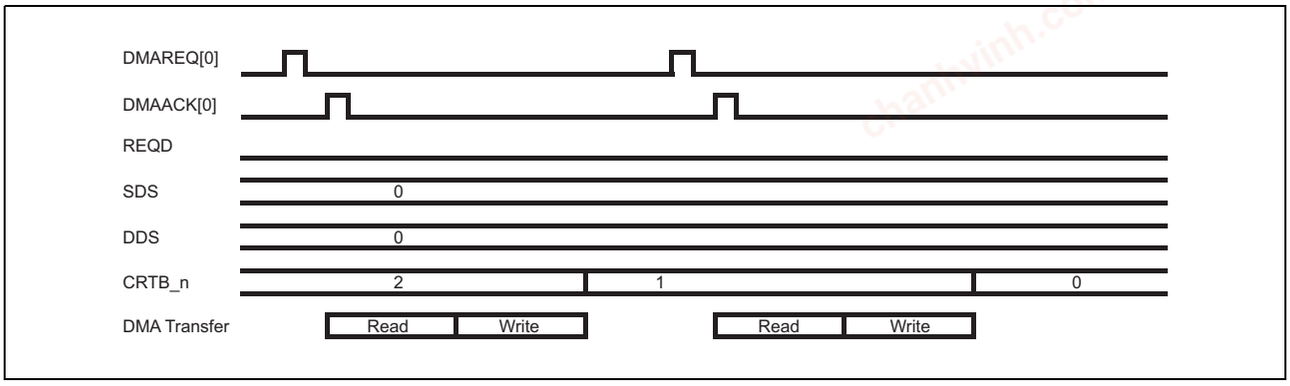


Figure 9.27 When the Source Transfer Size Is the Same as the Destination Transfer Size (LVL = 0, HIEN = 1, REQD = 0, SDS = DDS in CHCFG_n)

9.7.11 Transfer Status

The channel status register indicates the status of DMA transfer execution on a channel.

(1) Suspend

A DMA transfer can be suspended by using the SETSUS bit of CHCTRL_n. In this case, if an ongoing bus cycle exists, the DMAC waits for that cycle to end before suspending the transfer. Writing 1 in the CLRSUS bit restores the DMA transfer from the suspend status.

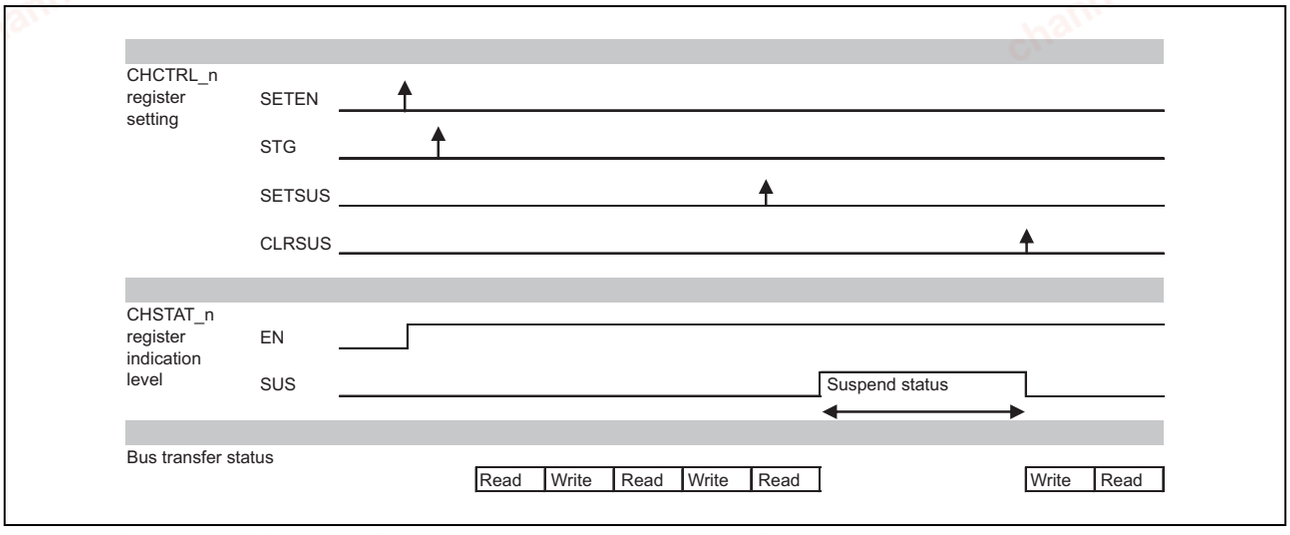


Figure 9.28 DMAC Suspend Status (Auto Request/Block Transfer)

In the above case, the DMA transfer is suspended after the read transfer is completed.
If there is any ongoing DMA transfer, the suspend status starts when that transfer is completed. To make sure that the transfer is suspended, read the CHSTAT or DSTAT_SUS register, after setting SETSUS, and check that 1 is set in the SUS bit for the relevant channel.

chanhvinh.com

(2) Transfer Stop

If you write 1 to CLREN while a DMA transaction is in progress, you can stop the DMA transaction for the corresponding channel. For the post-stop processing, two modes are supported: one sweeps out the data remaining in the buffer when the transaction is stopped (SBE = 1) and the other does not (SBE = 0). One of these modes can be selected using the SBE bit of the CHCFG_n register. By default, SBE is set to 0.
When this sweep mode is enabled and CLREN is set to 1, and if a DMA transaction is stopped with data remaining in the DMAC buffer, the transaction is completed after the DMAC sweeps the data.

(a) Transfer Stop (Buffer Sweep Disabled - SBE = 0)

If you set 1 in CLREN during a DMA transfer, the DMA transfer is stopped. The stop timing depends on the value set in REQD. After stopping a DMA transfer, be sure to set 1 in SWRST to clear the DMA internal status before setting the next transfer.

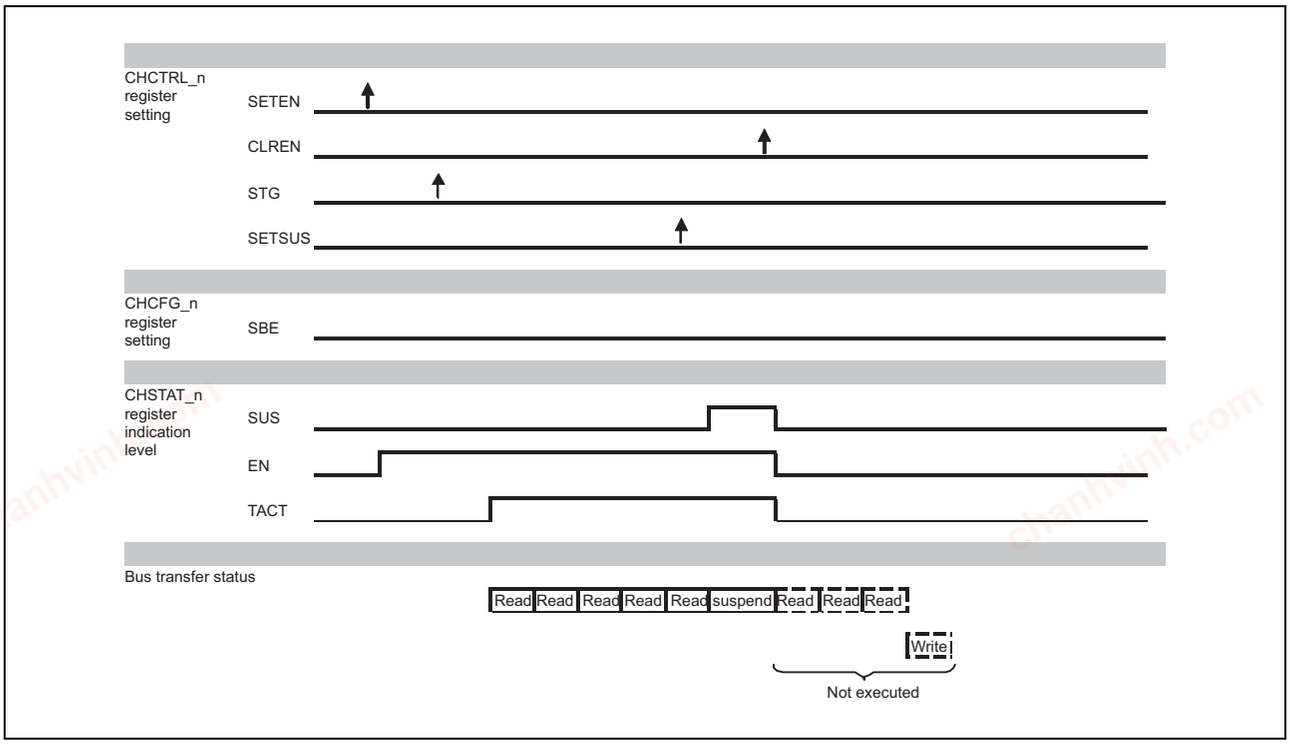


Figure 9.29 DMAC Transfer Stop

- When the TACT bit is set to 0, you can check that the channel has been brought to a complete stop.
- If you stop an ongoing DMA transfer before it is completed, the DMA transfer end interrupt is not asserted.
- If 0 is set in REQD, the DMA transfer is stopped when the next read is completed. (If the buffer contains any data that can be written, the DMA transfer is stopped after the data is written.)
- If 1 is set in REQD, the DMA transfer is stopped when the next write is completed.

chanhvinh.com

(b) Transfer Stop (Buffer Sweep Enabled - SBE = 1)

If 1 is set in CLREN during a DMA transfer, the DMA transfer is stopped. When 0 is set in REQD, the DMA transfer is stopped after the DMAC sweeps (writes) the already read data. If 1 is set in REQD to use hardware requests, do not use the sweep mode. After stopping a DMA transfer, be sure to set 1 in SWRST to clear the DMAC internal status before setting the next transfer.

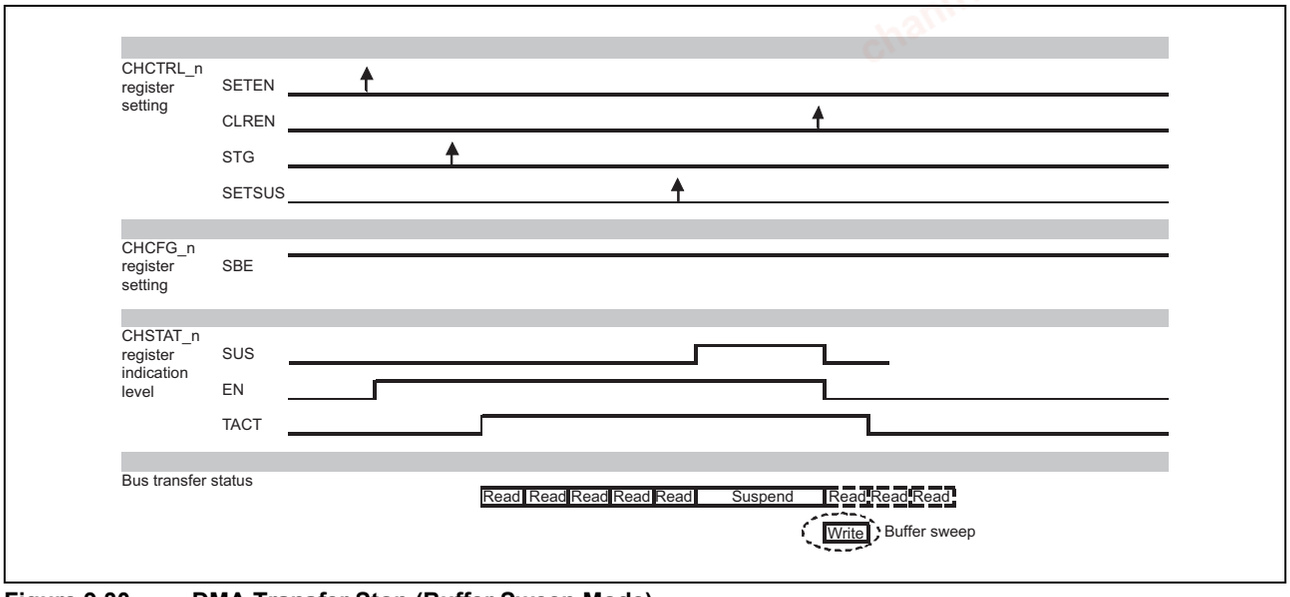


Figure 9.30 DMA Transfer Stop (Buffer Sweep Mode)

- When the TACT bit is set to 0, you can check that the channel has been brought to a complete stop.
- If you stop a transfer in sweep mode (SBE = 1) during the fifth read transfer by setting SETSUS and then CLREN, the read data is written before the DMA transfer is stopped.

(c) Channel Stop Check Method

Even when you clear the EN bit to 0, you cannot stop the DMA transfer immediately, if the bus is already executing the transfer. Therefore, in order to make sure that the DMAC has been brought to a complete stop, you need to check that the EN bit and TACT bit are both set to 0.

(d) Transfer Stop Procedure

The transfer stop procedure is described below.

1. Set 1 in SETSUS of CHCTRL_n.
2. Repeat polling until the SUS bit of CHSTAT_n is set to 1. (If EN is already set to 0, the DMAC has already been stopped. Go to step 6.)
3. Set 1 in CLREN of CHCTRL_n.
4. When 0 is set in SBE, the transfer is stopped according to the value of REQD. When 1 is set in SBE, the sweep mode is enabled. When 1 is set in SBE, set 0 in REQD.
5. Read CHSTAT_n to check that 0 is set in the TACT bit. When TACT is set to 0, it means that the DMAC has been brought to a complete stop. When TACT is set to 1, repeat polling until this bit is set to 0.
6. To execute the next DMA transfer after stopping a transfer, be sure to set 1 in the SWRST (software reset) bit of CHCTRL_n before the next DMA transfer starts.

chanhvinh.com

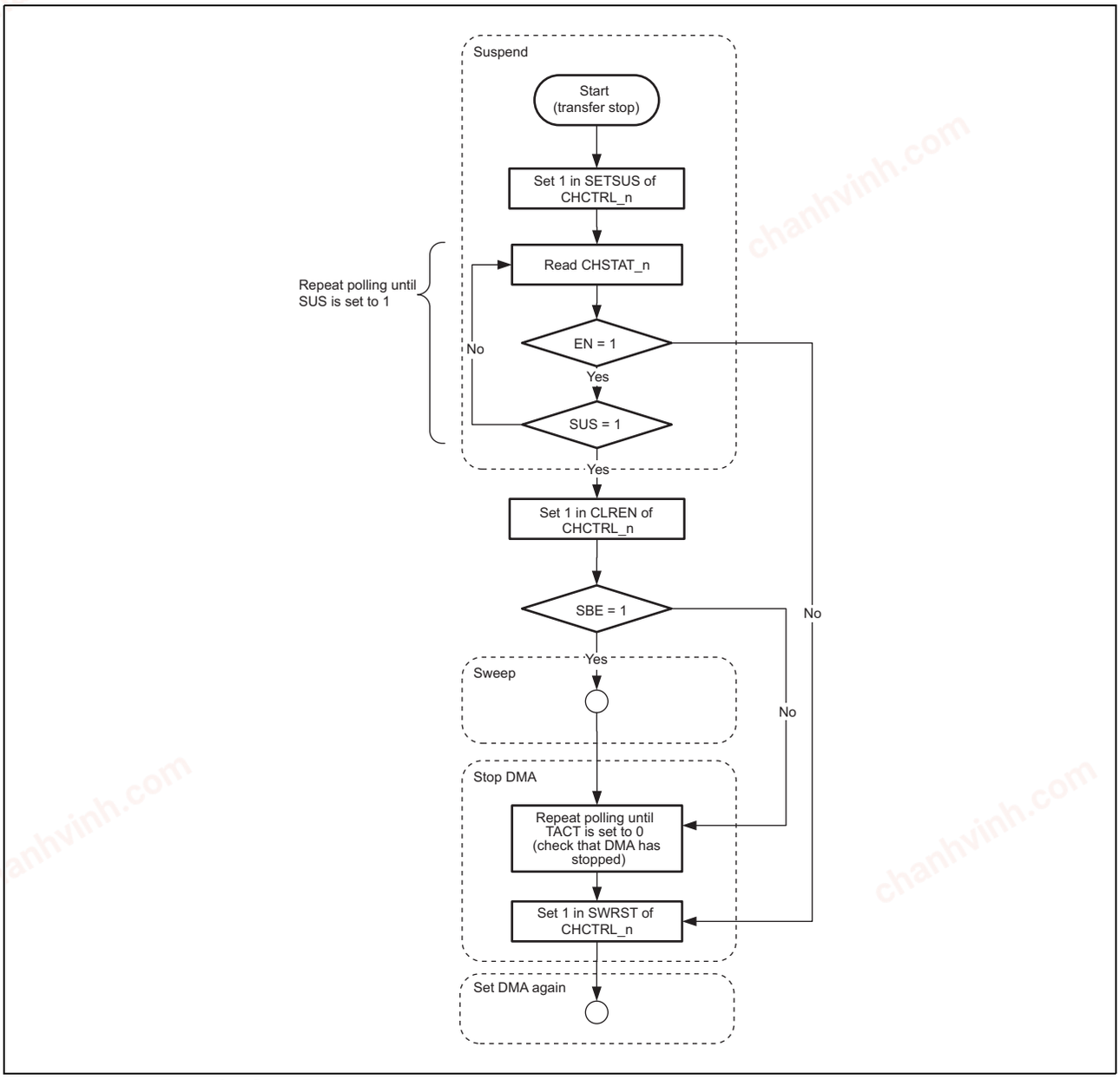


Figure 9.31 Transfer Stop Flow

chanhvinh.com

9.8 DMA Setting Examples

Setting examples applicable when DMA transfer is executed using the direct memory access controller are shown in the following.

The transfer conditions for these setting examples are as follows.

Table 9.21 Transfer Condition List for DMA Transfer Setting Examples

	DMA Mode	Transfer Mode	Transfer Request
Setting example 1	Register	Single	Hardware
Setting example 2	Register	Block	Software
Setting example 3	Register (continuous execution)	Block	Software
Setting example 4	Link	Block	Software

For details of the settings, see the individual setting examples.

9.8.1 Setting Example 1 (Register Mode/Hardware Request)

The following table shows a setting example applicable when DMA transfer is executed using the settings shown below.

Table 9.22 DMA Transfer Setting Example 1

Item		Description	
Channel used		3	
DMA mode		Register	
Transfer mode		Single transfer	
Register set used		Next0	
Source/destination		Source	Destination
	Start address	1111_0000H	2222_0000H
	Address direction	Increment	Increment
	Data size	32 bits	32 bits
DMA transfer byte count		64 bytes	
DMA transfer request		Rising edge detection by hardware	
DMAACK signal		Level output during read	
DMA transfer end interrupt mask		Not masked	
CACHE setting		Default value	

Setting example 1

N0SA = 1111_0000H (source address)
N0DA = 2222_0000H (destination address)
N0TB = 0000_0040H (transfer byte count)

CHCFG = 0002_2123H (configuration)
CHITVL = 0000_0000H (interval)
CHEXT = 0000_0000H (CACHE setting)

chanhvinh.com

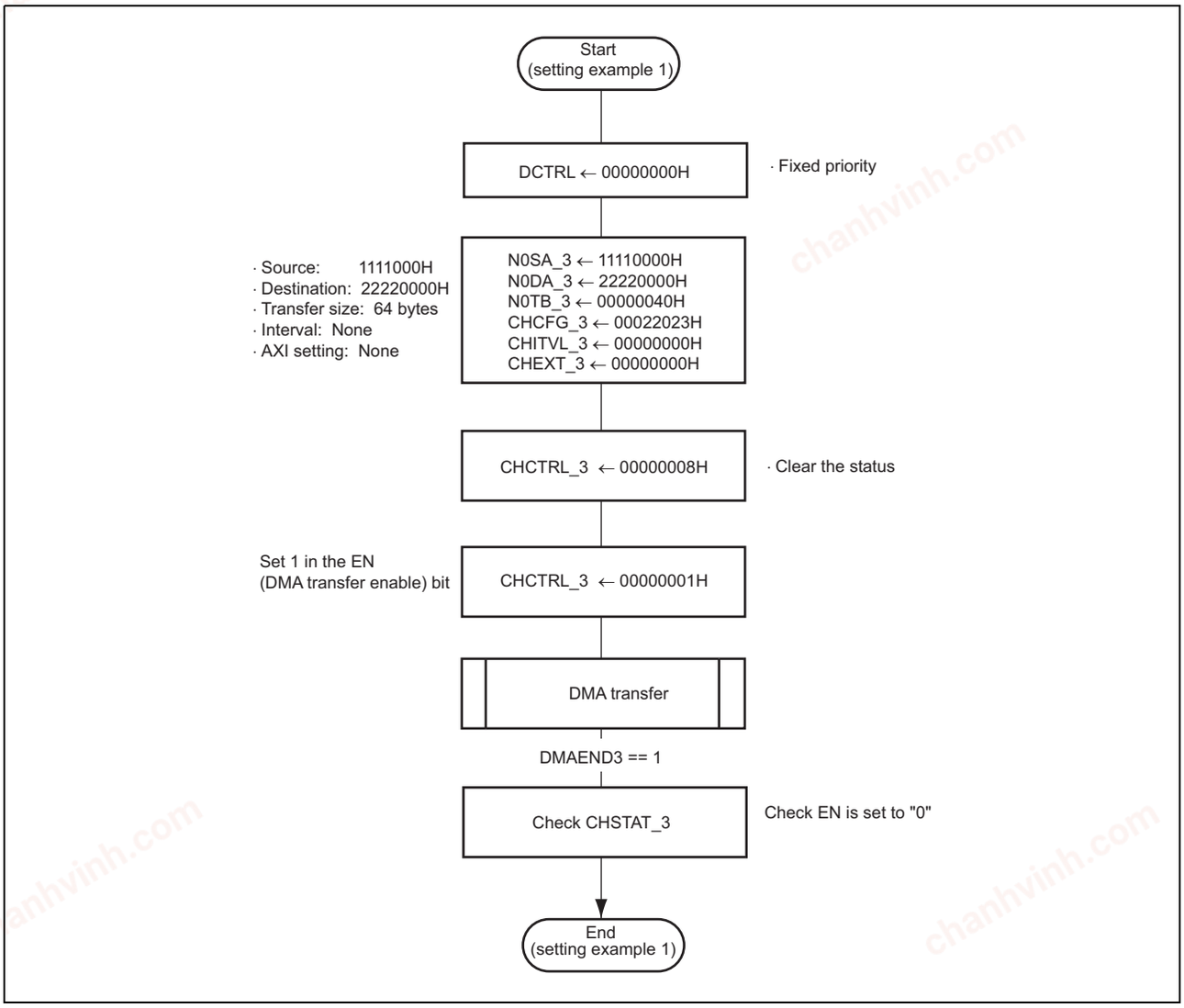


Figure 9.32 Setting Example 1

chanhvinh.com

9.8.2 Setting Example 2 (Register Mode/Software Request)

The following table shows a setting example applicable when DMA transfer is executed using the settings shown below.

Table 9.23 DMA Transfer Setting Example 2

Item		Description	
Channel used		2	
Priority control		Round robin	
DMA mode		Register	
Transfer mode		Block transfer	
Register set used		Next1	
Source/destination		Source	Destination
	Start address	0FFF_E000H	3333_0000H
	Address direction	Increment	Increment
	Data size	8 bits	256 bits
DMA transfer byte count		128 bytes	
DMA transfer request		Auto request	
DMAACK signal		Masked	
DMA transfer end interrupt mask		Not masked	
CACHE setting		Default value	

Setting example 2

DCTRL = 0000_0001H (DMA setting)

N1SA = 0FFF_E000H (source address)
N1DA = 3333_0000H (destination address)
N1TB = 0000_0080H (transfer byte count)

CHCFG = 1045_0402H (configuration)
CHITVL = 0000_0000H (interval)
CHEXT = 0000_0000H (CACHE setting)

chanhvinh.com

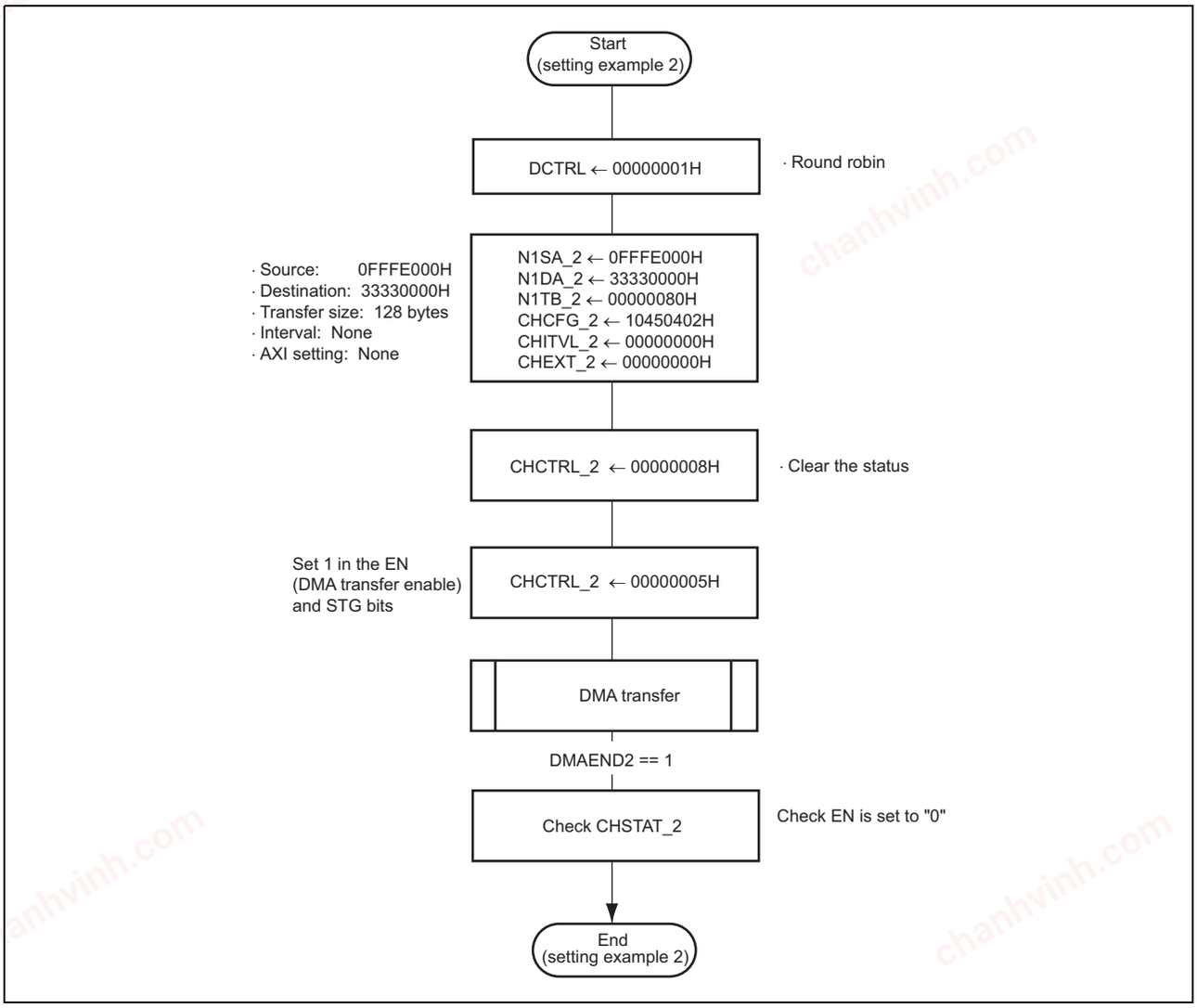


Figure 9.33 Setting Example 2

chanhvinh.com

9.8.3 Setting Example 3 (Register Mode/Continuous Execution)

The following table shows a setting example applicable when DMA transfer is executed using the settings shown below.

Table 9.24 DMA Transfer Setting Example 3

Item		Description	
Channel used		1	
Priority control		Round robin	
DMA mode		Register	
Transfer mode		Block transfer	
Register set used		Use Next0 and then Next1 continuously	
Next0		Source	Destination
	Start address	1111_0000H	3333_0000H
	Address direction	Fixed	Fixed
	Data size	32 bits	512 bits
	DMA transfer byte count	512 bytes	
Next1		Source	Destination
	Start address	2222_0000H	4444_0000H
	Address direction	Fixed	Fixed
	Data size	32 bits	512 bits
	DMA transfer byte count	2048 bytes	
DMA transfer request		Auto request	
DMAACK signal		Not output	
DMA transfer end interrupt mask		Mask the DMA transfer end interrupt upon completion of Next0	
CACHE setting		Default value	

Setting example 3

DCTRL = 0000_0001H (DMA setting)

N0SA = 1111_0000H (source address)
N0DA = 3333_0000H (destination address)
N0TB = 0000_0200H (transfer byte count)

N1SA = 2222_0000H (source address)
N1DA = 4444_0000H (destination address)
N1TB = 0000_0800H (transfer byte count)

CHCFG = 6176_2001H (configuration)
CHITVL = 0000_0000H (interval)
CHEXT = 0000_0000H (CACHE setting)

chanhvinh.com

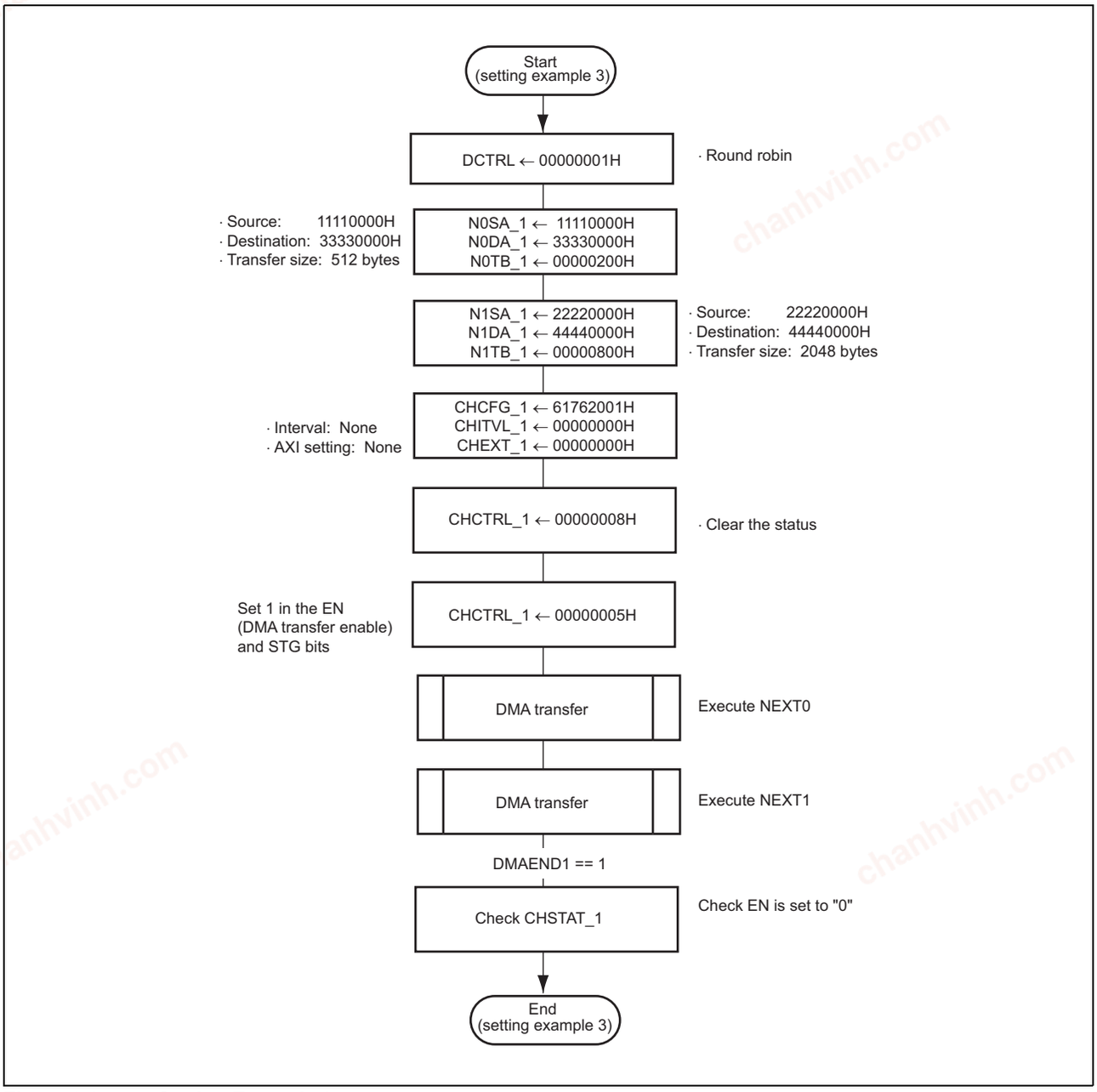


Figure 9.34 Setting Example 3

9.8.4 Setting Example 4 (Link Mode)

The following table shows a setting example applicable when DMA transfer is executed using the settings shown below.

Table 9.25 DMA Transfer Setting Example 4

Item	Description
Channel used	0
Priority control	Round robin
DMA mode	Link
Transfer mode	Block transfer
Register set used	—
Descriptor start address	0000_1000H

chanhvinh.com

Table 9.26 DMA Transfer Setting Example 4 (Descriptor 1)

Item		Description	
Descriptor start address		0000_1000H	
Next descriptor start address		0000_2000H	
Transfer mode		Block transfer	
Next0		Source	Destination
	Start address	1111_0000H	3333_0000H
	Address direction	Increment	Increment
	Data size	32 bits	32 bits
	DMA transfer byte count	2048 bytes	
DMA transfer request		Auto request trigger (STG)	
DMAACK signal		Not output	
DMA transfer end interrupt mask		Masked	
CACHE setting		Default value	
header			
	DMA interrupt when LV = 1	Issued (DIM = 0)	
	LV writeback	Done (WBD = 0)	
	Next link address	Available (LE = 0)	
	Descriptor valid	Valid (LV = 1)	

Table 9.27 DMA Transfer Setting Example 4 (Descriptor 2)

Item		Description	
Descriptor start address		0000_2000H	
Next descriptor start address		0000_5000H	
Transfer mode		Block transfer	
Next0		Source	Destination
	Start address	4444_0000H	5555_0000H
	Address direction	Increment	Increment
	Data size	64 bits	256 bits
	DMA transfer byte count	1024 bytes	
DMA transfer request		Auto request trigger (STG)	
DMAACK signal		Not output	
DMA transfer end interrupt mask		Masked	
CACHE setting		Default value	
header			
	DMA interrupt when LV = 1	Issued (DIM = 0)	
	LV writeback	Done (WBD = 0)	
	Next link address	Available (LE = 0)	
	Descriptor valid	Valid (LV = 1)	

chanhvinh.com

Table 9.28 DMA Transfer Setting Example 4 (Descriptor 3)

Item		Description	
Descriptor start address		0000_5000H	
Next descriptor start address		—	
Transfer mode		Block transfer	
Next0		Source	Destination
	Start address	7777_0000H	AAAA_0000H
	Address direction	Increment	Increment
	Data size	512 bits	512 bits
	DMA transfer byte count	4096 bytes	
DMA transfer request		Auto request trigger (STG)	
DMAACK signal		Not output	
DMA transfer end interrupt mask		Not masked	
CACHE setting		Default value	
header			
	DMA interrupt when LV = 1	Issued (DIM = 0)	
	LV writeback	Done (WBD = 0)	
	Next link address	Not available (LE = 1)	
	Descriptor valid	Valid (LV = 1)	

Setting example 4

DCTRL= 0000_0001H (DMA setting)
NXLA = 0000_1000H (descriptor start address)
CHCFG = 8000_0000H (configuration)

Table 9.29 Descriptor Setting

	Descriptor 1	Descriptor 2	Descriptor 3
header	0000_0001H	0000_0001H	0000_0003H
SA (Source Address)	1111_0000H	4444_0000H	7777_0000H
DA (Destination Address)	3333_0000H	5555_0000H	AAAA_0000H
TB (Transaction Byte)	0000_0800H	0000_0400H	0000_1000H
CFG (Configuration)	8142_2008H	8145_3008H	8046_6008H
ITVL (Interval)	0000_0000H	0000_0000H	0000_0000H
EXT (Extension)	0000_0000H	0000_0000H	0000_0000H
NXLA (Next Link Address)	0000_2000H	0000_5000H	0000_0000H

chanhvinh.com

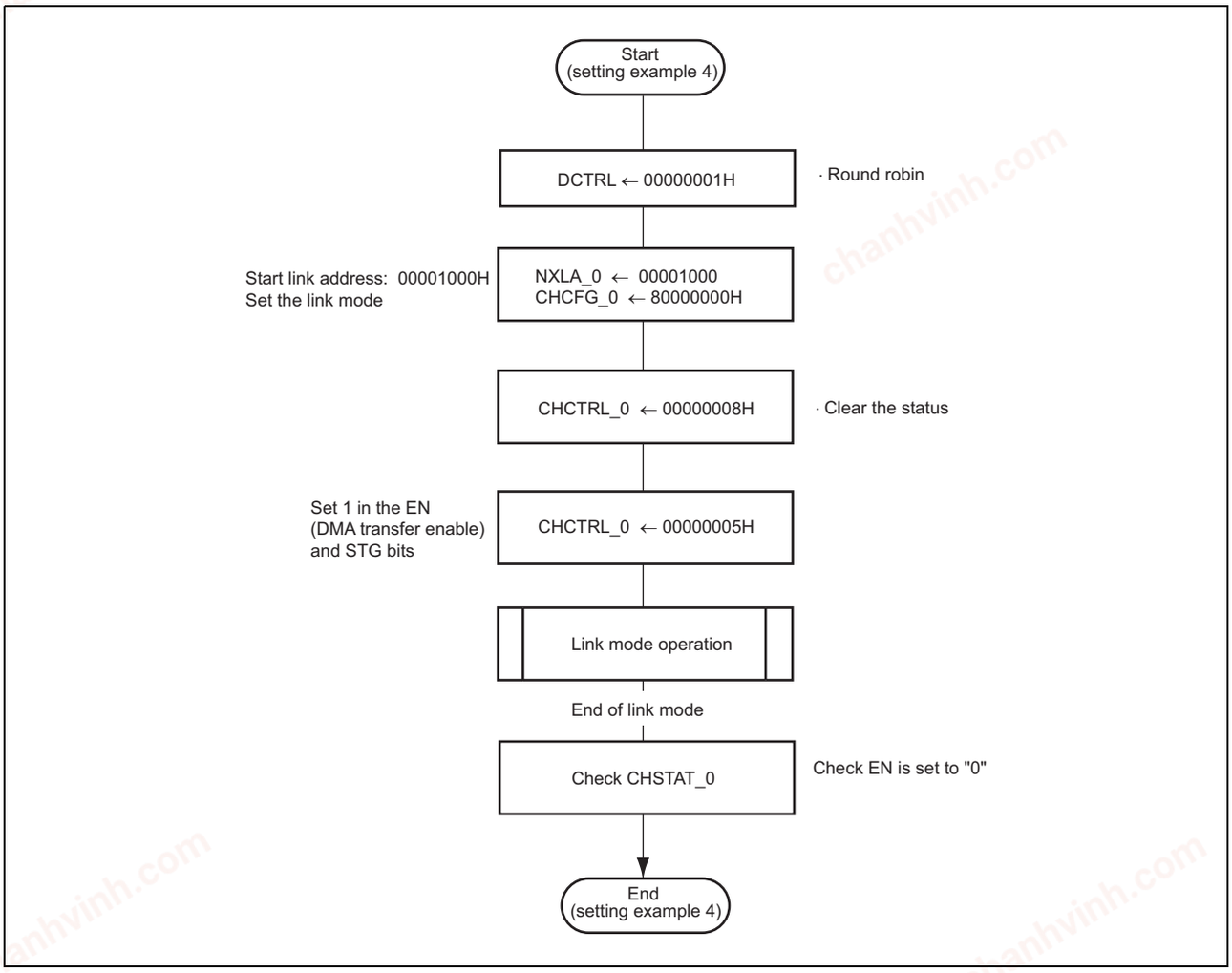


Figure 9.35 Setting Example 4

9.8.5 Next Register Set Continuous Execution Setting

The following figure shows the flowchart for executing DMA transfers continuously by using two Next Register Sets in register mode. While a DMA transaction is being executed using one Next Register Set, the other Next Register Set is set in order to continue to execute DMA transfers.

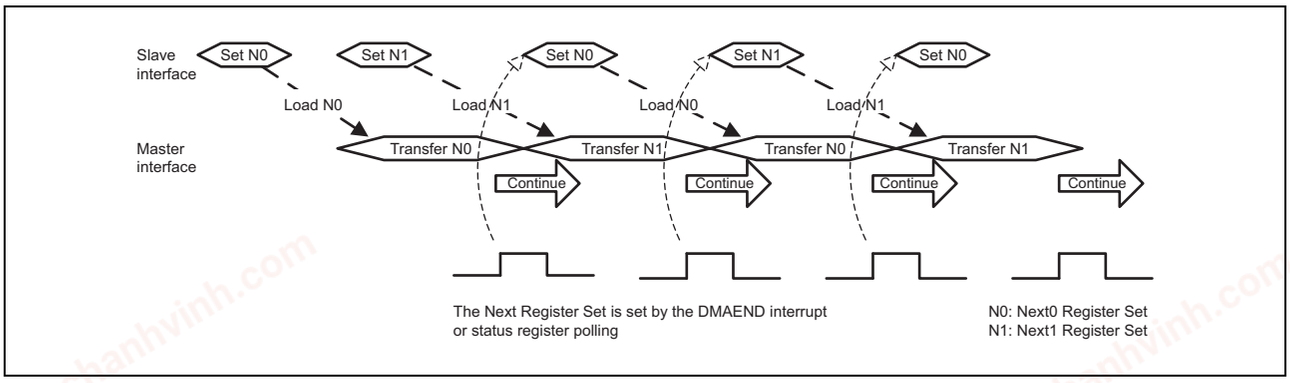


Figure 9.36 Image of Next Register Set Continuous Execution

chanhvinh.com

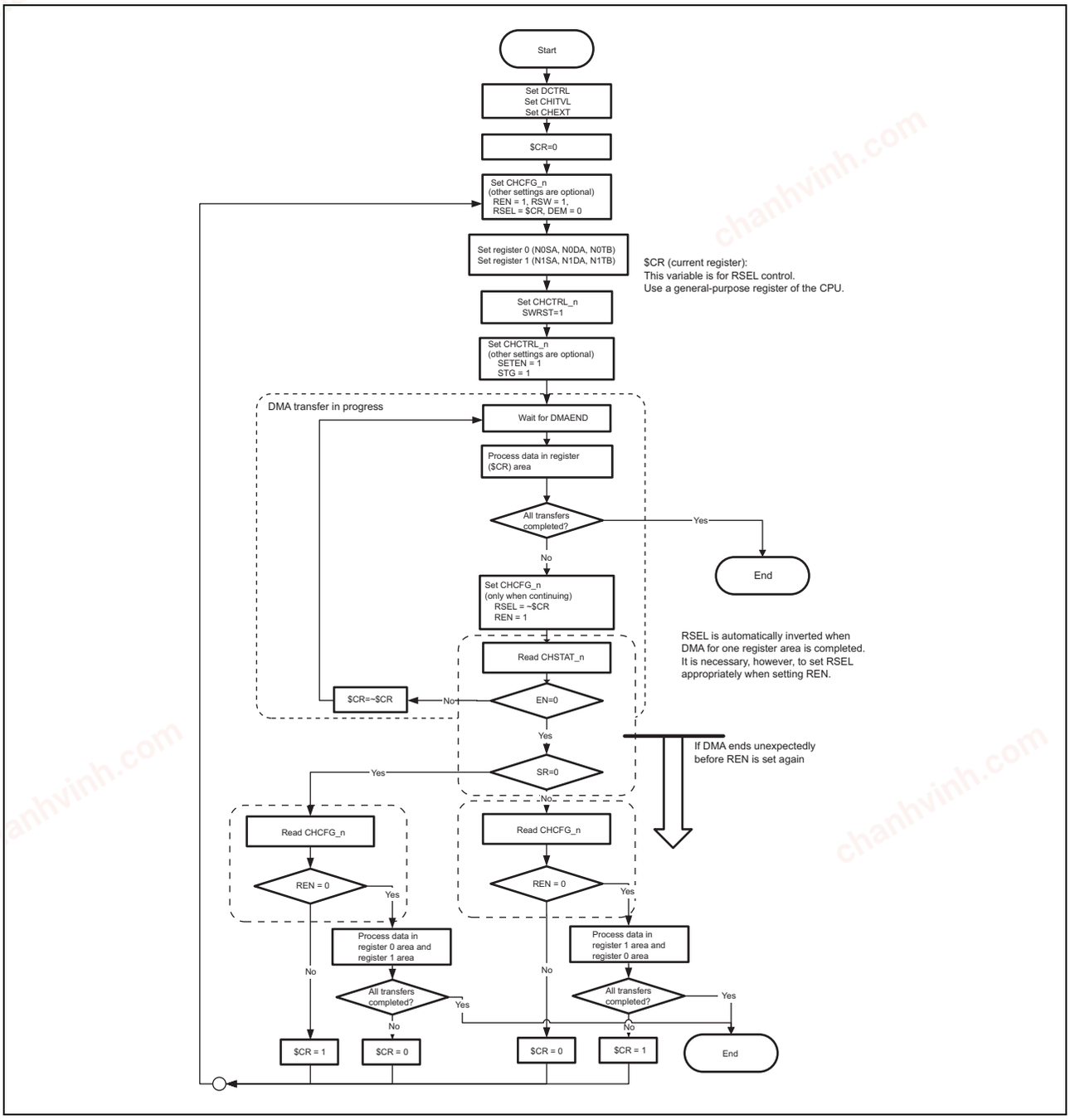


Figure 9.37 Example of Continuous DMA Execution by Using a Next Register Set

- Supplementary information

  First, save the data of the register sets to be used for DMA transfers (0 (N0SA, N0DA, and N0TB) and 1 (N1SA, N1DA, and N1TB)) to a general-purpose register of the CPU (the values of this register is referred to as $CR for the sake of convenience).

  Each time the DMA transfers for one register set are completed (the DMA transfer end interrupt is output), REN is automatically cleared to 0. In order to continue to execute DMA transfers, it is necessary to set REN of the CHCFG_n register every time the DMA transfer end interrupt is asserted. This register also contains the RSEL bit, and the value of this bit needs to be set appropriately as well. Therefore, use $CR.

  In this mode, two Next Register Sets are executed continuously. However, if CLREN is not set before the DMA transaction is completed (the next DMA transfer end interrupt is output), continuous execution stops. In this case, you can check how much of data has been transferred, by reading the SR and EN bits of the CHSTAT_n register and the REN bit of the CHCFG_n register. To restart the DMA transaction, follow the flowchart shown above.

chanhvinh.com

9.9 Note

9.9.1 Divided Output of DACK0 and TEND0

When transferring 4 bytes or more to an 8-bit or 16-bit external device or transferring 2 bytes or more to an 8-bit external device, each DMA transfer unit is divided into multiple bus cycles. Note that, if the setting is such that DMA transfer is divided into multiple bus cycles and $\overline{CS}$ is negated between bus cycles, the DACK0 output and the TEND0 output is divided to align data as with $\overline{CS}$. Figure 9.38 shows an example.

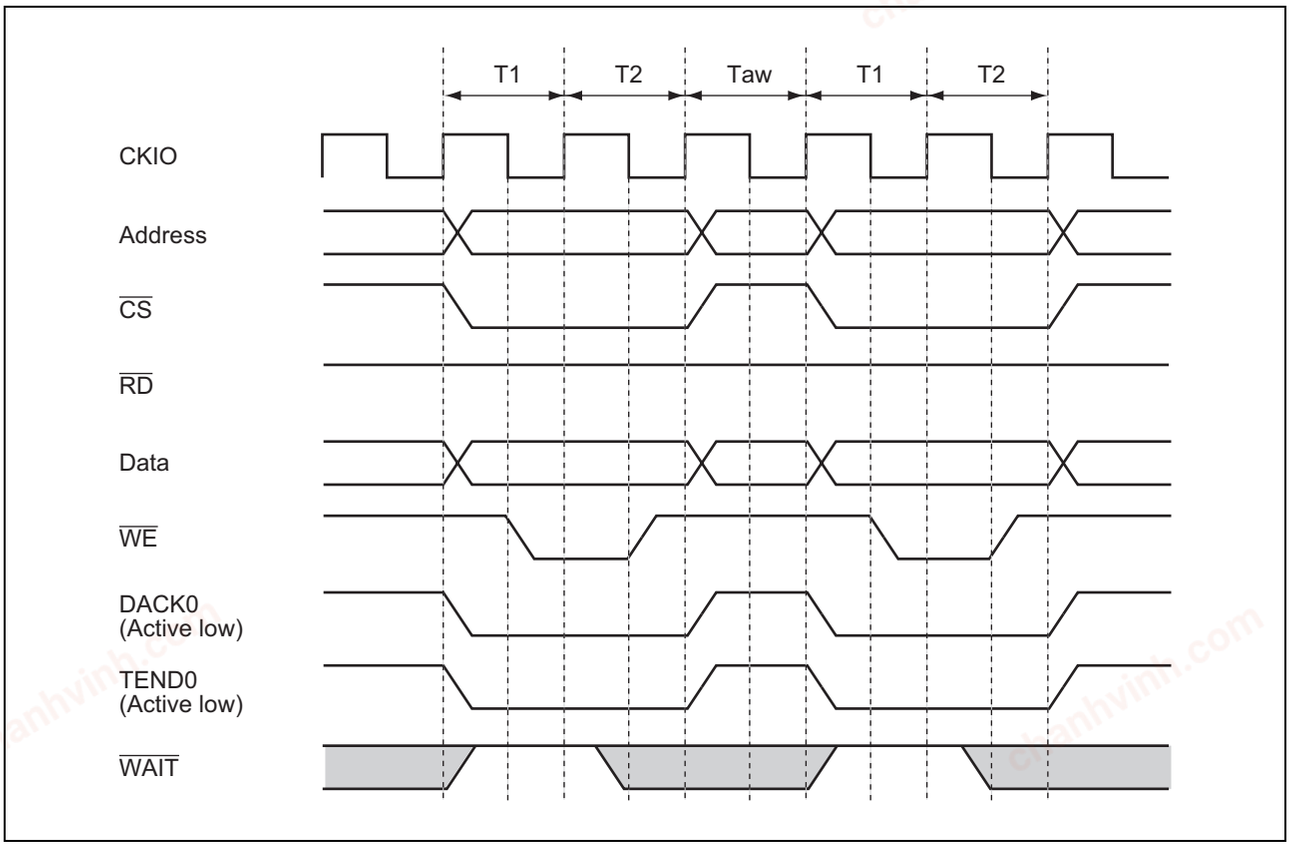


Figure 9.38 Example of TEND0 Divided Output Timing

chanhvinh.com

9.9.2 TEND0 Not Output

Note that TEND0 may not be output depending on the combination of the bits DDS[3:0], SDS[3:0] and REQD in the CHCFG_0 register.

Table 9.30 shows when TEND0 is not output and Figure 9.39 shows an operation example.

Table 9.30 Bit Combination when TEND0 Is Not Output

CHCFG_0 Register			
REQD	**DDS**	**SDS**	**TEND0 Output**
1	—	—	Output
0	DDS > SDS		Output
	DDS = SDS		Output
	DDS < SDS		Not output

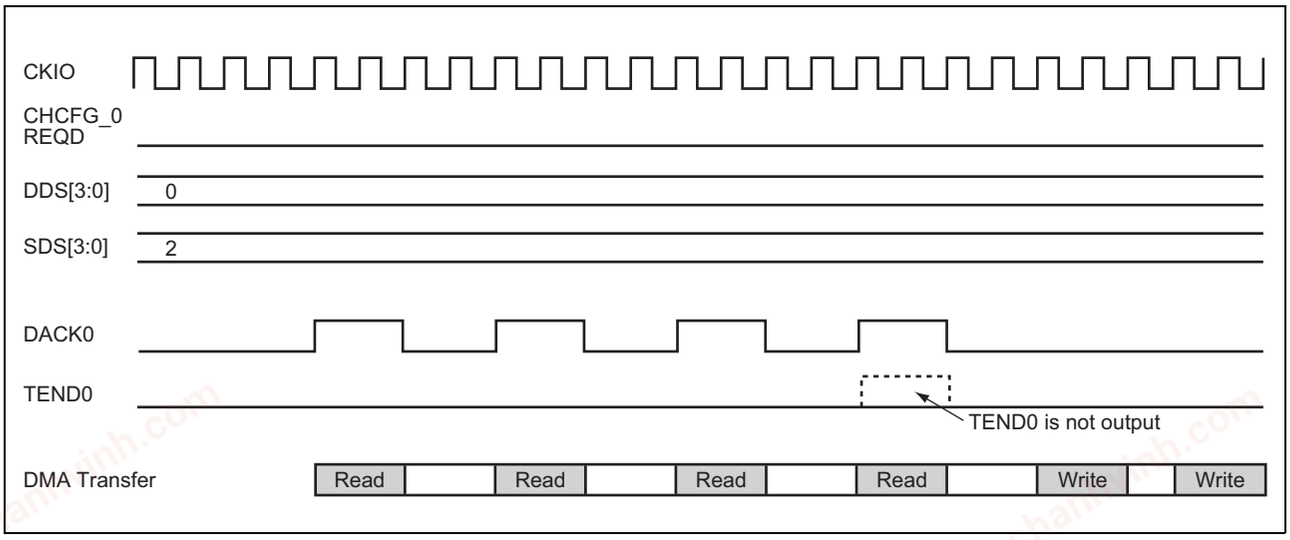


Figure 9.39 TEND0 Not Output

9.9.3 Atomic Access (ARLOCK[1:0] and AWLOCK[1:0])

This module does not support atomic (locked or exclusive) access, that is, it only supports normal access.

Signals ARLOCK[1:0] and AWLOCK[1:0] are fixed as follows and cannot be modified.

ARLOCK[1:0], AWLOCK[1:0]: 00 (normal access)

chanhvinh.com

10. Multi-Function Timer Pulse Unit 2

This LSI has an on-chip multi-function timer pulse unit 2 that comprises five 16-bit timer channels.

10.1 Features

- Maximum 16 pulse input/output lines
- Selection of eight counter input clocks for each channel
- The following operations can be set:
  - Waveform output at compare match
  - Input capture function
  - Counter clear operation
  - Multiple timer counters (TCNT) can be written to simultaneously
  - Simultaneous clearing by compare match and input capture is possible
  - Register simultaneous input/output is possible by synchronous counter operation
  - A maximum 12-phase PWM output is possible in combination with synchronous operation
- Buffer operation settable for channels 0, 3, and 4
- Phase counting mode settable independently for each of channels 1 and 2
- Cascade connection operation
- Fast access via internal 16-bit bus
- 25 interrupt sources
- Automatic transfer of register data
- A/D converter start trigger can be generated
- Module standby mode can be settable
- A total of six-phase waveform output, which includes complementary PWM output, and positive and negative phases of reset PWM output by interlocking operation of channels 3 and 4, is possible.
- AC synchronous motor (brushless DC motor) drive mode using complementary PWM output and reset PWM output is settable by interlocking operation of channels 0, 3, and 4, and the selection of two types of waveform outputs (chopping and level) is possible.
- In complementary PWM mode, interrupts at the crest and trough of the counter value and A/D converter start triggers can be skipped.

chanhvinh.com

Table 10.1 Functions of Multi-Function Timer Pulse Unit 2

Item		Channel 0	Channel 1	Channel 2	Channel 3	Channel 4
Count clock		Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 TCLKA TCLKB TCLKC TCLKD	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 Pϕ/256 TCLKA TCLKB	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 Pϕ/1024 TCLKA TCLKB TCLKC	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 Pϕ/256 Pϕ/1024 TCLKA TCLKB	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 Pϕ/256 Pϕ/1024 TCLKA TCLKB
General registers		TGRA_0 TGRB_0 TGRE_0	TGRA_1 TGRB_1	TGRA_2 TGRB_2	TGRA_3 TGRB_3	TGRA_4 TGRB_4
General registers/ buffer registers		TGRC_0 TGRD_0 TGRF_0	—	—	TGRC_3 TGRD_3	TGRC_4 TGRD_4
I/O pins		TIOC0A TIOC0B TIOC0C TIOC0D	TIOC1A TIOC1B	TIOC2A TIOC2B	TIOC3A TIOC3B TIOC3C TIOC3D	TIOC4A TIOC4B TIOC4C TIOC4D
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	√	√	√	√	√
	1 output	√	√	√	√	√
	Toggle output	√	√	√	√	√
Input capture function		√	√	√	√	√
Synchronous operation		√	√	√	√	√
PWM mode 1		√	√	√	√	√
PWM mode 2		√	√	√	—	—
Complementary PWM mode		—	—	—	√	√
Reset PWM mode		—	—	—	√	√
AC synchronous motor drive mode		√	—	—	√	√
Phase counting mode		—	√	√	—	—
Buffer operation		√	—	—	√	√
Activation of direct memory access controller		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture and TCNT overflow or underflow
A/D converter start trigger		TGRA_0 compare match or input capture TGRE_0 compare match	TGRA_1 compare match or input capture	TGRA_2 compare match or input capture	TGRA_3 compare match or input capture	TGRA_4 compare match or input capture TCNT_4 underflow (trough) in complementary PWM mode

chanhvinh.com

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4
Interrupt sources	7 sources • Compare match or input capture 0A • Compare match or input capture 0B • Compare match or input capture 0C • Compare match or input capture 0D • Compare match 0E • Compare match 0F • Overflow	4 sources • Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow	4 sources • Compare match or input capture 2A • Compare match or input capture 2B • Overflow • Underflow	5 sources • Compare match or input capture 3A • Compare match or input capture 3B • Compare match or input capture 3C • Compare match or input capture 3D • Overflow	5 sources • Compare match or input capture 4A • Compare match or input capture 4B • Compare match or input capture 4C • Compare match or input capture 4D • Overflow or underflow
A/D converter start request delaying function	—	—	—	—	• A/D converter start request at a match between TADCORA_4 and TCNT_4 • A/D converter start request at a match between TADCORB_4 and TCNT_4
Interrupt skipping function	—	—	—	• Skips TGRA_3 compare match interrupts	• Skips TCIV_4 interrupts

[Legend]
√: Available
—: Not available

chanhvinh.com

Figure 10.1 shows a block diagram.

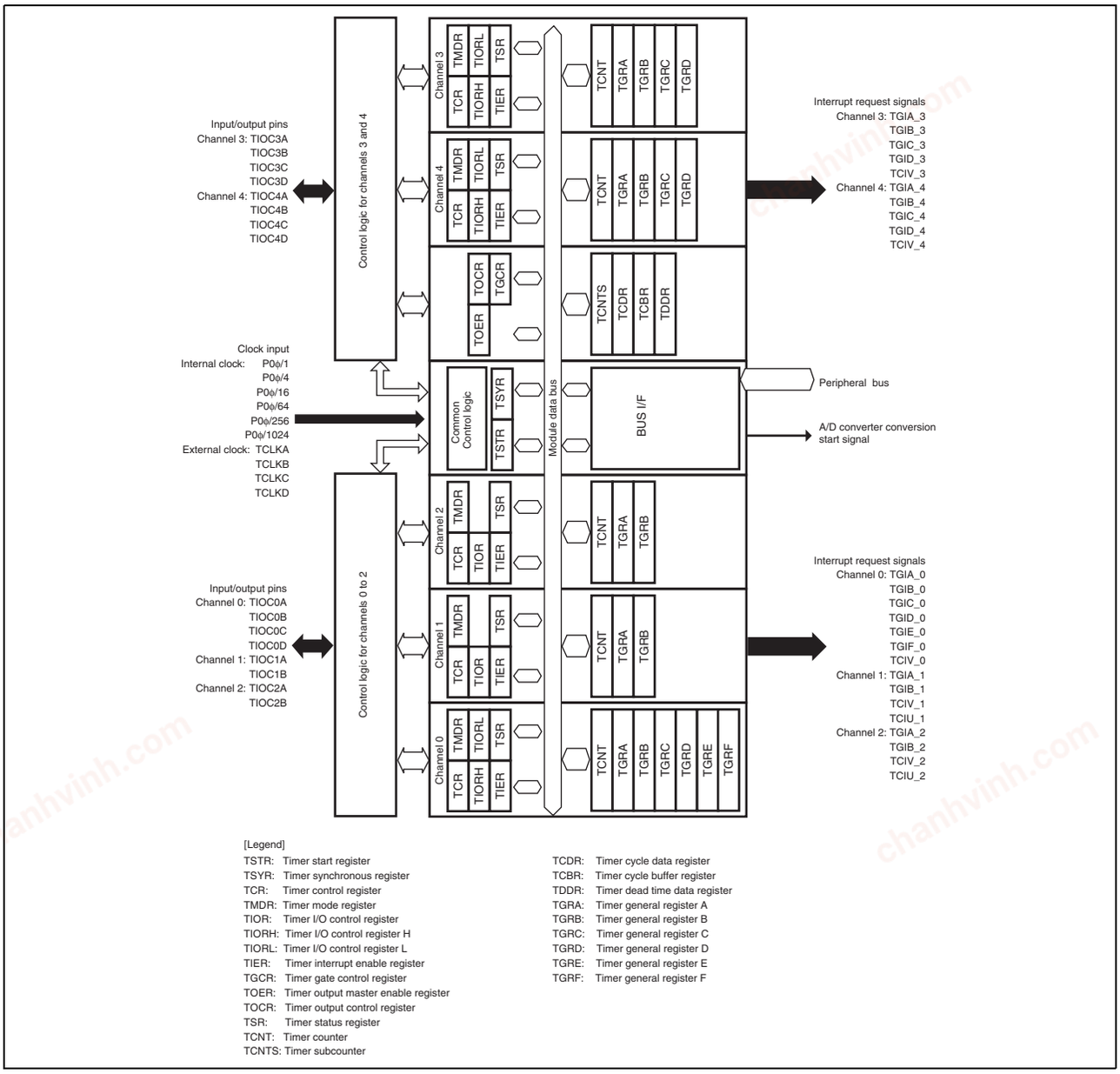


Figure 10.1 Block Diagram

chanhvinh.com

10.2 Input/Output Pins

Table 10.2 shows the pin configuration

Table 10.2 Pin Configuration

Channel	Pin Name	I/O	Function
Common	TCLKA	Input	External clock A input pin (Channel 1 phase counting mode A phase input)
	TCLKB	Input	External clock B input pin (Channel 1 phase counting mode B phase input)
	TCLKC	Input	External clock C input pin (Channel 2 phase counting mode A phase input)
	TCLKD	Input	External clock D input pin (Channel 2 phase counting mode B phase input)
0	TIOC0A	I/O	TGRA_0 input capture input/output compare output/PWM output pin
	TIOC0B	I/O	TGRB_0 input capture input/output compare output/PWM output pin
	TIOC0C	I/O	TGRC_0 input capture input/output compare output/PWM output pin
	TIOC0D	I/O	TGRD_0 input capture input/output compare output/PWM output pin
1	TIOC1A	I/O	TGRA_1 input capture input/output compare output/PWM output pin
	TIOC1B	I/O	TGRB_1 input capture input/output compare output/PWM output pin
2	TIOC2A	I/O	TGRA_2 input capture input/output compare output/PWM output pin
	TIOC2B	I/O	TGRB_2 input capture input/output compare output/PWM output pin
3	TIOC3A	I/O	TGRA_3 input capture input/output compare output/PWM output pin
	TIOC3B	I/O	TGRB_3 input capture input/output compare output/PWM output pin
	TIOC3C	I/O	TGRC_3 input capture input/output compare output/PWM output pin
	TIOC3D	I/O	TGRD_3 input capture input/output compare output/PWM output pin
4	TIOC4A	I/O	TGRA_4 input capture input/output compare output/PWM output pin
	TIOC4B	I/O	TGRB_4 input capture input/output compare output/PWM output pin
	TIOC4C	I/O	TGRC_4 input capture input/output compare output/PWM output pin
	TIOC4D	I/O	TGRD_4 input capture input/output compare output/PWM output pin

Note: For the pin configuration in complementary PWM mode, see Table 10.52 in section 10.4.8, Complementary PWM Mode.

chanhvinh.com

10.3 Register Descriptions

Table 10.3 shows the register configuration. To distinguish registers in each channel, an underscore and the channel number are added as a suffix to the register name; TCR for channel 0 is expressed as TCR_0.

Table 10.3 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access Size
0	Timer control register_0	TCR_0	R/W	H'00	H'FCFF0300	8
	Timer mode register_0	TMDR_0	R/W	H'00	H'FCFF0301	8
	Timer I/O control register H_0	TIORH_0	R/W	H'00	H'FCFF0302	8
	Timer I/O control register L_0	TIORL_0	R/W	H'00	H'FCFF0303	8
	Timer interrupt enable register_0	TIER_0	R/W	H'00	H'FCFF0304	8
	Timer status register_0	TSR_0	R/W	H'C0	H'FCFF0305	8
	Timer counter_0	TCNT_0	R/W	H'0000	H'FCFF0306	16
	Timer general register A_0	TGRA_0	R/W	H'FFFF	H'FCFF0308	16
	Timer general register B_0	TGRB_0	R/W	H'FFFF	H'FCFF030A	16
	Timer general register C_0	TGRC_0	R/W	H'FFFF	H'FCFF030C	16
	Timer general register D_0	TGRD_0	R/W	H'FFFF	H'FCFF030E	16
	Timer general register E_0	TGRE_0	R/W	H'FFFF	H'FCFF0320	16
	Timer general register F_0	TGRF_0	R/W	H'FFFF	H'FCFF0322	16
	Timer interrupt enable register 2_0	TIER2_0	R/W	H'00	H'FCFF0324	8
	Timer status register 2_0	TSR2_0	R/W	H'C0	H'FCFF0325	8
	Timer buffer operation transfer mode register_0	TBTM_0	R/W	H'00	H'FCFF0326	8
1	Timer control register_1	TCR_1	R/W	H'00	H'FCFF0380	8
	Timer mode register_1	TMDR_1	R/W	H'00	H'FCFF0381	8
	Timer I/O control register_1	TIOR_1	R/W	H'00	H'FCFF0382	8
	Timer interrupt enable register_1	TIER_1	R/W	H'00	H'FCFF0384	8
	Timer status register_1	TSR_1	R/W	H'C0	H'FCFF0385	8
	Timer counter_1	TCNT_1	R/W	H'0000	H'FCFF0386	16
	Timer general register A_1	TGRA_1	R/W	H'FFFF	H'FCFF0388	16
	Timer general register B_1	TGRB_1	R/W	H'FFFF	H'FCFF038A	16
	Timer input capture control register	TICCR	R/W	H'00	H'FCFF0390	8
2	Timer control register_2	TCR_2	R/W	H'00	H'FCFF0000	8
	Timer mode register_2	TMDR_2	R/W	H'00	H'FCFF0001	8
	Timer I/O control register_2	TIOR_2	R/W	H'00	H'FCFF0002	8
	Timer interrupt enable register_2	TIER_2	R/W	H'00	H'FCFF0004	8
	Timer status register_2	TSR_2	R/W	H'C0	H'FCFF0005	8
	Timer counter_2	TCNT_2	R/W	H'0000	H'FCFF0006	16
	Timer general register A_2	TGRA_2	R/W	H'FFFF	H'FCFF0008	16
	Timer general register B_2	TGRB_2	R/W	H'FFFF	H'FCFF000A	16
3	Timer control register_3	TCR_3	R/W	H'00	H'FCFF0200	8
	Timer mode register_3	TMDR_3	R/W	H'00	H'FCFF0202	8
	Timer I/O control register H_3	TIORH_3	R/W	H'00	H'FCFF0204	8
	Timer I/O control register L_3	TIORL_3	R/W	H'00	H'FCFF0205	8
	Timer interrupt enable register_3	TIER_3	R/W	H'00	H'FCFF0208	8
	Timer status register_3	TSR_3	R/W	H'C0	H'FCFF022C	8
	Timer counter_3	TCNT_3	R/W	H'0000	H'FCFF0210	16
	Timer general register A_3	TGRA_3	R/W	H'FFFF	H'FCFF0218	16
	Timer general register B_3	TGRB_3	R/W	H'FFFF	H'FCFF021A	16

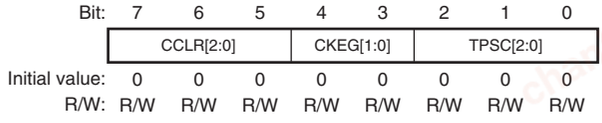
chanhvinh.com

Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access Size
3	Timer general register C_3	TGRC_3	R/W	H'FFFF	H'FCFF0224	16
	Timer general register D_3	TGRD_3	R/W	H'FFFF	H'FCFF0226	16
	Timer buffer operation transfer mode register_3	TBTM_3	R/W	H'00	H'FCFF0238	8
4	Timer control register_4	TCR_4	R/W	H'00	H'FCFF0201	8
	Timer mode register_4	TMDR_4	R/W	H'00	H'FCFF0203	8
	Timer I/O control register H_4	TIORH_4	R/W	H'00	H'FCFF0206	8
	Timer I/O control register L_4	TIORL_4	R/W	H'00	H'FCFF0207	8
	Timer interrupt enable register_4	TIER_4	R/W	H'00	H'FCFF0209	8
	Timer status register_4	TSR_4	R/W	H'C0	H'FCFF022D	8
	Timer counter_4	TCNT_4	R/W	H'0000	H'FCFF0212	16
	Timer general register A_4	TGRA_4	R/W	H'FFFF	H'FCFF021C	16
	Timer general register B_4	TGRB_4	R/W	H'FFFF	H'FCFF021E	16
	Timer general register C_4	TGRC_4	R/W	H'FFFF	H'FCFF0228	16
	Timer general register D_4	TGRD_4	R/W	H'FFFF	H'FCFF022A	16
	Timer buffer operation transfer mode register_4	TBTM_4	R/W	H'00	H'FCFF0239	8
	Timer A/D converter start request control register	TADCR	R/W	H'0000	H'FCFF0240	16
	Timer A/D converter start request cycle set register A_4	TADCORA_4	R/W	H'FFFF	H'FCFF0244	16
	Timer A/D converter start request cycle set register B_4	TADCORB_4	R/W	H'FFFF	H'FCFF0246	16
	Timer A/D converter start request cycle set buffer register A_4	TADCOBRA_4	R/W	H'FFFF	H'FCFF0248	16
	Timer A/D converter start request cycle set buffer register B_4	TADCOBRB_4	R/W	H'FFFF	H'FCFF024A	16
Common	Timer start register	TSTR	R/W	H'00	H'FCFF0280	8
	Timer synchronous register	TSYR	R/W	H'00	H'FCFF0281	8
	Timer read/write enable register	TRWER	R/W	H'01	H'FCFF0284	8
Common to 3 and 4	Timer output master enable register	TOER	R/W	H'C0	H'FCFF020A	8
	Timer output control register 1	TOCR1	R/W	H'00	H'FCFF020E	8
	Timer output control register 2	TOCR2	R/W	H'00	H'FCFF020F	8
	Timer gate control register	TGCR	R/W	H80	H'FCFF020D	8
	Timer cycle data register	TCDR	R/W	H'FFFF	H'FCFF0214	16
	Timer dead time data register	TDDR	R/W	H'FFFF	H'FCFF0216	16
	Timer subcounter	TCNTS	R	H'0000	H'FCFF0220	16
	Timer cycle buffer register	TCBR	R/W	H'FFFF	H'FCFF0222	16
	Timer interrupt skipping set register	TITCR	R/W	H'00	H'FCFF0230	8
	Timer interrupt skipping counter	TITCNT	R	H'00	H'FCFF0231	8
	Timer buffer transfer set register	TBTER	R/W	H'00	H'FCFF0232	8
	Timer dead time enable register	TDER	R/W	H'01	H'FCFF0234	8
	Timer waveform control register	TWCR	R/W	H'00	H'FCFF0260	8
	Timer output level buffer register	TOLBR	R/W	H'00	H'FCFF0236	8

chanhvinh.com

10.3.1 Timer Control Register (TCR)

The TCR registers are 8-bit readable/writable registers that control the TCNT operation for each channel. This module has a total of five TCR registers, one each for channels 0 to 4. TCR register settings should be conducted only when TCNT operation is stopped.



Bit	Bit Name	Initial Value	R/W	Description
7 to 5	CCLR[2:0]	000	R/W	Counter Clear 0 to 2 These bits select the TCNT counter clearing source. See Table 10.4 and Table 10.5 for details.
4, 3	CKEG[1:0]	00	R/W	Clock Edge 0 and 1 These bits select the input clock edge. When the input clock is counted using both edges, the input clock period is halved (e.g. Pϕ/4 both edges = Pϕ/2 rising edge). If phase counting mode is used on channels 1 and 2, this setting is ignored and the phase counting mode setting has priority. Internal clock edge selection is valid when the input clock is Pϕ/4 or slower. When Pϕ/1, or the overflow/underflow of another channel is selected for the input clock, although values can be written, counter operation compiles with the initial value. 00: Count at rising edge 01: Count at falling edge 1x: Count at both edges
2 to 0	TPSC[2:0]	000	R/W	Time Prescaler 0 to 2 These bits select the TCNT counter clock. The clock source can be selected independently for each channel. See Table 10.6 to Table 10.9 for details.

[Legend]
x: Don't care

Table 10.4 CCLR0 to CCLR2 (Channels 0, 3, and 4)

Channel	Bit 7 CCLR2	Bit 6 CCLR1	Bit 5 CCLR0	Description
0, 3, 4	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*1
	1	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRC compare match/input capture*2
		1	0	TCNT cleared by TGRD compare match/input capture*2
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*1

Note 1. Synchronous operation is set by setting the SYNC bit in TSYR to 1.
Note 2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority, and compare match/input capture does not occur.

chanhvinh.com

Table 10.5 CCLR0 to CCLR2 (Channels 1 and 2)

Channel	Bit 7 Reserved*2	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*1

Note 1. Synchronous operation is selected by setting the SYNC bit in TSYR to 1.
Note 2. Bit 7 is reserved in channels 1 and 2. It is always read as 0 and cannot be modified.

Table 10.6 TPSC0 to TPSC2 (Channel 0)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
0	0	0	0	Internal clock: counts on P0ϕ/1
			1	Internal clock: counts on P0ϕ/4
		1	0	Internal clock: counts on P0ϕ/16
			1	Internal clock: counts on P0ϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	External clock: counts on TCLKC pin input
			1	External clock: counts on TCLKD pin input

Table 10.7 TPSC0 to TPSC2 (Channel 1)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
1	0	0	0	Internal clock: counts on P0ϕ/1
			1	Internal clock: counts on P0ϕ/4
		1	0	Internal clock: counts on P0ϕ/16
			1	Internal clock: counts on P0ϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	Internal clock: counts on P0ϕ/256
			1	Counts on TCNT_2 overflow/underflow

Note: This setting is ignored when channel 1 is in phase counting mode.

Table 10.8 TPSC0 to TPSC2 (Channel 2)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
2	0	0	0	Internal clock: counts on P0ϕ/1
			1	Internal clock: counts on P0ϕ/4
		1	0	Internal clock: counts on P0ϕ/16
			1	Internal clock: counts on P0ϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	External clock: counts on TCLKC pin input
			1	Internal clock: counts on P0ϕ/1024

Note: This setting is ignored when channel 2 is in phase counting mode.

chanhvinh.com

Table 10.9 TPSC0 to TPSC2 (Channels 3 and 4)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
3, 4	0	0	0	Internal clock: counts on P0ϕ/1
			1	Internal clock: counts on P0ϕ/4
		1	0	Internal clock: counts on P0ϕ/16
			1	Internal clock: counts on P0ϕ/64
	1	0	0	Internal clock: counts on P0ϕ/256
			1	Internal clock: counts on P0ϕ/1024
		1	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input

chanhvinh.com

10.3.2 Timer Mode Register (TMDR)

The TMDR registers are 8-bit readable/writable registers that are used to set the operating mode of each channel. This module has five TMDR registers, one each for channels 0 to 4. TMDR register settings should be changed only when TCNT operation is stopped.

Bit:	7	6	5	4	3	2	1	0
	-	BFE	BFB	BFA	MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	BFE	0	R/W	Buffer Operation E Specifies whether TGRE_0 and TGRF_0 are to operate in the normal way or to be used together for buffer operation. TGRF compare match is generated when TGRF is used as the buffer register. In channels 1 to 4, this bit is reserved. It is always read as 0 and the write value should always be 0. 0: TGRE_0 and TGRF_0 operate normally 1: TGRE_0 and TGRF_0 used together for buffer operation
5	BFB	0	R/W	Buffer Operation B Specifies whether TGRB is to operate in the normal way, or TGRB and TGRD are to be used together for buffer operation. When TGRD is used as a buffer register, TGRD input capture/output compare is not generated in a mode other than complementary PWM. In channels 1 and 2, which have no TGRD, bit 5 is reserved. It is always read as 0 and cannot be modified. 0: TGRB and TGRD operate normally 1: TGRB and TGRD used together for buffer operation
4	BFA	0	R/W	Buffer Operation A Specifies whether TGRA is to operate in the normal way, or TGRA and TGRC are to be used together for buffer operation. When TGRC is used as a buffer register, TGRC input capture/output compare is not generated in a mode other than complementary PWM. TGRC compare match is generated when in complementary PWM mode. When compare match for channel 4 occurs during the Tb period in complementary PWM mode, TGFC is set. Therefore, set the TGIEC bit in the timer interrupt enable register 4 (TIER_4) to 0. In channels 1 and 2, which have no TGRC, bit 4 is reserved. It is always read as 0 and cannot be modified. 0: TGRA and TGRC operate normally 1: TGRA and TGRC used together for buffer operation
3 to 0	MD[3:0]	0000	R/W	Modes 0 to 3 These bits are used to set the timer operating mode. See Table 10.10 for details.

chanhvinh.com

Table 10.10 Setting of Operation Mode by Bits MD0 to MD3

Bit 3 MD3	Bit 2 MD2	Bit 1 MD1	Bit 0 MD0	Description
0	0	0	0	Normal operation
			1	Setting prohibited
		1	0	PWM mode 1
			1	PWM mode 2*1
	1	0	0	Phase counting mode 1*2
			1	Phase counting mode 2*2
		1	0	Phase counting mode 3*2
			1	Phase counting mode 4*2
1	0	0	0	Reset synchronous PWM mode*3
			1	Setting prohibited
		1	X	Setting prohibited
	1	0	0	Setting prohibited
			1	Complementary PWM mode 1 (transmit at crest)*3
		1	0	Complementary PWM mode 2 (transmit at trough)*3
			1	Complementary PWM mode 3 (transmit at crest and trough)*3

[Legend]
X: Don't care
Note 1. PWM mode 2 cannot be set for channels 3 and 4.
Note 2. Phase counting mode cannot be set for channels 0, 3, and 4.
Note 3. Reset synchronous PWM mode, complementary PWM mode can only be set for channel 3. When channel 3 is set to reset synchronous PWM mode or complementary PWM mode, the channel 4 settings become ineffective and automatically conform to the channel 3 settings. However, do not set channel 4 to reset synchronous PWM mode or complementary PWM mode. Reset synchronous PWM mode and complementary PWM mode cannot be set for channels 0, 1, and 2.

chanhvinh.com

10.3.3 Timer I/O Control Register (TIOR)

The TIOR registers are 8-bit readable/writable registers that control the TGR registers. This module has a total of eight TIOR registers, two each for channels 0, 3, and 4, one each for channels 1 and 2.

TIOR should be set while TMDR is set in normal operation, PWM mode, or phase counting mode.

The initial output specified by TIOR is valid when the counter is stopped (the CST bit in TSTR is cleared to 0). Note also that, in PWM mode 2, the output at the point at which the counter is cleared to 0 is specified.

When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

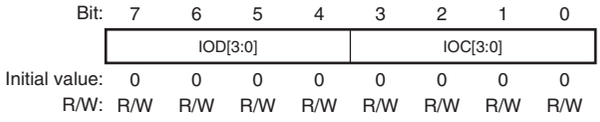
- TIORH_0, TIOR_1, TIOR_2, TIORH_3, TIORH_4

Bit:	7	6	5	4	3	2	1	0
	IOB[3:0]				IOA[3:0]			
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	IOB[3:0]	0000	R/W	I/O Control B0 to B3 Specify the function of TGRB. See the following tables. TIORH_0: Table 10.11 TIOR_1: Table 10.13 TIOR_2: Table 10.14 TIORH_3: Table 10.15 TIORH_4: Table 10.17
3 to 0	IOA[3:0]	0000	R/W	I/O Control A0 to A3 Specify the function of TGRA. See the following tables. TIORH_0: Table 10.19 TIOR_1: Table 10.21 TIOR_2: Table 10.22 TIORH_3: Table 10.23 TIORH_4: Table 10.25

chanhvinh.com

- TIORL_0, TIORL_3, TIORL_4



Bit	Bit Name	Initial Value	R/W	Description
7 to 4	IOD[3:0]	0000	R/W	I/O Control D0 to D3 Specify the function of TGRD. See the following tables. TIORL_0: Table 10.12 TIORL_3: Table 10.16 TIORL_4: Table 10.18
3 to 0	IOC[3:0]	0000	R/W	I/O Control C0 to C3 Specify the function of TGRC. See the following tables. TIORL_0: Table 10.20 TIORL_3: Table 10.24 TIORL_4: Table 10.26

chanhvinh.com

Table 10.11 TIORH_0 (Channel 0)

Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description	
				TGRB_0 Function	TIOC0B Pin Function
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

[Legend]
X: Don't care
Note: * After power-on reset, 0 is output until TIOR is set.

chanhvinh.com

Table 10.12 TIORL_0 (Channel 0)

Bit 7 IOD3	Bit 6 IOD2	Bit 5 IOD1	Bit 4 IOD0	Description	
				TGRD_0 Function	TIOC0D Pin Function
0	0	0	0	Output compare register*2	Output retained*1
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

[Legend]
X: Don't care
Note 1. After power-on reset, 0 is output until TIOR is set.
Note 2. When the BFB bit in TMDR_0 is set to 1 and TGRD_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

chanhvinh.com

Table 10.13 TIOR_1 (Channel 1)

Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description	
				TGRB_1 Function	TIOC1B Pin Function
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Input capture at generation of TGRC_0 compare match/input capture

[Legend]
X: Don't care
Note: * After power-on reset, 0 is output until TIOR is set.

chanhvinh.com

Table 10.14 TIOR_2 (Channel 2)

Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description	
				TGRB_2 Function	TIOC2B Pin Function
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]
X: Don't care
Note: * After power-on reset, 0 is output until TIOR is set.

chanhvinh.com

Table 10.15 TIORH_3 (Channel 3)

Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description	
				TGRB_3 Function	TIOC3B Pin Function
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]
X: Don't care
Note: * After power-on reset, 0 is output until TIOR is set.

chanhvinh.com

Table 10.16 TIORL_3 (Channel 3)

Bit 7 IOD3	Bit 6 IOD2	Bit 5 IOD1	Bit 4 IOD0	Description	
				TGRD_3 Function	TIOC3D Pin Function
0	0	0	0	Output compare register*2	Output retained*1
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]
X: Don't care
Note 1. After power-on reset, 0 is output until TIOR is set.
Note 2. When the BFB bit in TMDR_3 is set to 1 and TGRD_3 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

chanhvinh.com

Table 10.17 TIORH_4 (Channel 4)

Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description	
				TGRB_4 Function	TIOC4B Pin Function
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]
X: Don't care
Note: * After power-on reset, 0 is output until TIOR is set.

chanhvinh.com

Table 10.18 TIORL_4 (Channel 4)

Bit 7 IOD3	Bit 6 IOD2	Bit 5 IOD1	Bit 4 IOD0	Description	
				TGRD_4 Function	TIOC4D Pin Function
0	0	0	0	Output compare register*2	Output retained*1
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]
X: Don't care
Note 1. After power-on reset, 0 is output until TIOR is set.
Note 2. When the BFB bit in TMDR_4 is set to 1 and TGRD_4 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

chanhvinh.com

Table 10.19 TIORH_0 (Channel 0)

Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description	
				TGRA_0 Function	TIOC0A Pin Function
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

[Legend]
X: Don't care
Note: * After power-on reset, 0 is output until TIOR is set.

chanhvinh.com

Table 10.20 TIORL_0 (Channel 0)

Bit 3 IOC3	Bit 2 IOC2	Bit 1 IOC1	Bit 0 IOC0	Description	
				TGRC_0 Function	TIOC0C Pin Function
0	0	0	0	Output compare register*2	Output retained*1
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

[Legend]
X: Don't care
Note 1. After power-on reset, 0 is output until TIOR is set.
Note 2. When the BFA bit in TMDR_0 is set to 1 and TGRC_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

chanhvinh.com

Table 10.21 TIOR_1 (Channel 1)

Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description	
				TGRA_1 Function	TIOC1A Pin Function
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Input capture at generation of channel 0/TGRA_0 compare match/input capture

[Legend]
X: Don't care
Note: * After power-on reset, 0 is output until TIOR is set.

chanhvinh.com

Table 10.22 TIOR_2 (Channel 2)

Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description	
				TGRA_2 Function	TIOC2A Pin Function
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]
X: Don't care
Note: * After power-on reset, 0 is output until TIOR is set.

chanhvinh.com

Table 10.23 TIORH_3 (Channel 3)

Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description	
				TGRA_3 Function	TIOC3A Pin Function
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]
X: Don't care
Note: * After power-on reset, 0 is output until TIOR is set.

chanhvinh.com

Table 10.24 TIORL_3 (Channel 3)

Bit 3 IOC3	Bit 2 IOC2	Bit 1 IOC1	Bit 0 IOC0	Description	
				TGRC_3 Function	TIOC3C Pin Function
0	0	0	0	Output compare register*2	Output retained*1
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]
X: Don't care
Note 1. After power-on reset, 0 is output until TIOR is set.
Note 2. When the BFA bit in TMDR_3 is set to 1 and TGRC_3 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

chanhvinh.com

Table 10.25 TIORH_4 (Channel 4)

Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description	
				TGRA_4 Function	TIOC4A Pin Function
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]
X: Don't care
Note: * After power-on reset, 0 is output until TIOR is set.

chanhvinh.com

Table 10.26 TIORL_4 (Channel 4)

Bit 3 IOC3	Bit 2 IOC2	Bit 1 IOC1	Bit 0 IOC0	Description	
				TGRC_4 Function	TIOC4C Pin Function
0	0	0	0	Output compare register*2	Output retained*1
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]
X: Don't care
Note 1. After power-on reset, 0 is output until TIOR is set.
Note 2. When the BFA bit in TMDR_4 is set to 1 and TGRC_4 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

chanhvinh.com

10.3.4 Timer Interrupt Enable Register (TIER)

The TIER registers are 8-bit readable/writable registers that control enabling or disabling of interrupt requests for each channel. This module has six TIER registers, two for channel 0 and one each for channels 1 to 4.

- TIER_0, TIER_1, TIER_2, TIER_3, TIER_4

Bit:	7	6	5	4	3	2	1	0
	TTGE	TTGE2	TCIEU	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	TTGE	0	R/W	A/D Conversion Start Request Enable Enables or disables generation of A/D conversion start requests by TGRA input capture/compare match. 0: A/D converter start request generation disabled 1: A/D converter start request generation enabled
6	TTGE2	0	R/W	A/D Conversion Start Request Enable 2 Enables or disables generation of A/D conversion start requests by TCNT_4 underflow (trough) in complementary PWM mode. In channels 0 to 3, bit 6 is reserved. It is always read as 0 and the write value should always be 0. 0: A/D conversion start request generation by TCNT_4 underflow (trough) disabled 1: A/D conversion start request generation by TCNT_4 underflow (trough) enabled
5	TCIEU	0	R/W	Underflow Interrupt Enable Enables or disables interrupt requests (TCIU) by the TCFU flag when the TCFU flag in TSR is set to 1 in channels 1 and 2. In channels 0, 3, and 4, bit 5 is reserved. It is always read as 0 and the write value should always be 0. 0: Interrupt requests (TCIU) by TCFU disabled 1: Interrupt requests (TCIU) by TCFU enabled
4	TCIEV	0	R/W	Overflow Interrupt Enable Enables or disables interrupt requests (TCIV) by the TCFV flag when the TCFV flag in TSR is set to 1. 0: Interrupt requests (TCIV) by TCFV disabled 1: Interrupt requests (TCIV) by TCFV enabled
3	TGIED	0	R/W	TGR Interrupt Enable D Enables or disables interrupt requests (TGID) by the TGFD bit when the TGFD bit in TSR is set to 1 in channels 0, 3, and 4. In channels 1 and 2, bit 3 is reserved. It is always read as 0 and the write value should always be 0. 0: Interrupt requests (TGID) by TGFD bit disabled 1: Interrupt requests (TGID) by TGFD bit enabled
2	TGIEC	0	R/W	TGR Interrupt Enable C Enables or disables interrupt requests (TGIC) by the TGFC bit when the TGFC bit in TSR is set to 1 in channels 0, 3, and 4. In channels 1 and 2, bit 2 is reserved. It is always read as 0 and the write value should always be 0. 0: Interrupt requests (TGIC) by TGFC bit disabled 1: Interrupt requests (TGIC) by TGFC bit enabled
1	TGIEB	0	R/W	TGR Interrupt Enable B Enables or disables interrupt requests (TGIB) by the TGFB bit when the TGFB bit in TSR is set to 1. 0: Interrupt requests (TGIB) by TGFB bit disabled 1: Interrupt requests (TGIB) by TGFB bit enabled

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
0	TGIEA	0	R/W	TGR Interrupt Enable A Enables or disables interrupt requests (TGIA) by the TGFA bit when the TGFA bit in TSR is set to 1. 0: Interrupt requests (TGIA) by TGFA bit disabled 1: Interrupt requests (TGIA) by TGFA bit enabled

- TIER2_0

Bit:	7	6	5	4	3	2	1	0
	TTGE2	-	-	-	-	-	TGIEF	TGIEE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	TTGE2	0	R/W	A/D Converter Start Request Enable 2 Enables or disables generation of A/D converter start requests by compare match between TCNT_0 and TGRE_0. 0: A/D converter start request generation by compare match between TCNT_0 and TGRE_0 disabled 1: A/D converter start request generation by compare match between TCNT_0 and TGRE_0 enabled
6 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	TGIEF	0	R/W	TGR Interrupt Enable F Enables or disables interrupt requests by compare match between TCNT_0 and TGRF_0. 0: Interrupt requests (TGIF) by TGFE bit disabled 1: Interrupt requests (TGIF) by TGFE bit enabled
0	TGIEE	0	R/W	TGR Interrupt Enable E Enables or disables interrupt requests by compare match between TCNT_0 and TGRE_0. 0: Interrupt requests (TGIE) by TGEE bit disabled 1: Interrupt requests (TGIE) by TGEE bit enabled

chanhvinh.com

10.3.5 Timer Status Register (TSR)

The TSR registers are 8-bit readable/writable registers that indicate the status of each channel. This module has six TSR registers, two for channel 0 and one each for channels 1 to 4.

- TSR_0, TSR_1, TSR_2, TSR_3, TSR_4

Bit:	7	6	5	4	3	2	1	0
	TCFD	-	TCFU	TCFV	TGFD	TGFC	TGFB	TGFA
Initial value:	1	1	0	0	0	0	0	0
R/W:	R	R	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1

Note: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

Bit	Bit Name	Initial Value	R/W	Description
7	TCFD	1	R	Count Direction Flag Status flag that shows the direction in which TCNT counts in channels 1 to 4. In channel 0, bit 7 is reserved. It is always read as 1 and the write value should always be 1. 0: TCNT counts down 1: TCNT counts up
6	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
5	TCFU	0	R/(W)*1	Underflow Flag Status flag that indicates that TCNT underflow has occurred when channels 1 and 2 are set to phase counting mode. Only 0 can be written, for flag clearing. In channels 0, 3, and 4, bit 5 is reserved. It is always read as 0 and the write value should always be 0. [Clearing condition] • When 0 is written to TCFU after reading TCFU = 1*2 [Setting condition] • When the TCNT value underflows (changes from H'0000 to H'FFFF)
4	TCFV	0	R/(W)*1	Overflow Flag Status flag that indicates that TCNT overflow has occurred. Only 0 can be written, for flag clearing. [Clearing condition] • When 0 is written to TCFV after reading TCFV = 1*2 [Setting condition] • When the TCNT value overflows (changes from H'FFFF to H'0000) In channel 4, when the TCNT_4 value underflows (changes from H'0001 to H'0000) in complementary PWM mode, this flag is also set.
3	TGFD	0	R/(W)*1	Input Capture/Output Compare Flag D Status flag that indicates the occurrence of TGRD input capture or compare match in channels 0, 3, and 4. Only 0 can be written, for flag clearing. In channels 1 and 2, bit 3 is reserved. It is always read as 0 and the write value should always be 0. [Clearing condition] • When 0 is written to TGFD after reading TGFD = 1*2 [Setting conditions] • When TCNT = TGRD and TGRD is functioning as output compare register • When TCNT value is transferred to TGRD by input capture signal and TGRD is functioning as input capture register

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
2	TGFC	0	R/(W)*1	Input Capture/Output Compare Flag C Status flag that indicates the occurrence of TGRC input capture or compare match in channels 0, 3, and 4. Only 0 can be written, for flag clearing. In channels 1 and 2, bit 2 is reserved. It is always read as 0 and the write value should always be 0. [Clearing condition] • When 0 is written to TGFC after reading TGFC = 1*2 [Setting conditions] • When TCNT = TGRC and TGRC is functioning as output compare register • When TCNT value is transferred to TGRC by input capture signal and TGRC is functioning as input capture register
1	TGFB	0	R/(W)*1	Input Capture/Output Compare Flag B Status flag that indicates the occurrence of TGRB input capture or compare match. Only 0 can be written, for flag clearing. [Clearing condition] • When 0 is written to TGFB after reading TGFB = 1*2 [Setting conditions] • When TCNT = TGRB and TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal and TGRB is functioning as input capture register
0	TGFA	0	R/(W)*1	Input Capture/Output Compare Flag A Status flag that indicates the occurrence of TGRA input capture or compare match. Only 0 can be written, for flag clearing. [Clearing conditions] • When the direct memory access controller is activated by TGIA interrupt • When 0 is written to TGFA after reading TGFA = 1*2 [Setting conditions] • When TCNT = TGRA and TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal and TGRA is functioning as input capture register

Note 1. Writing 0 to this bit after reading it as 1 clears the flag.
Note 2. If the next flag is set before TGFA is cleared to 0 after reading TGFA = 1, TGFA remains 1 even when 0 is written to. In this case, read TGFA = 1 again to clear TGFA to 0.

chanhvinh.com

- TSR2_0

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	TGFF	TGFE
Initial value:	1	1	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/(W)*1	R/(W)*1

Note: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	TGFF	0	R/(W)*1	Compare Match Flag F Status flag that indicates the occurrence of compare match between TCNT_0 and TGRF_0. [Clearing condition] • When 0 is written to TGFF after reading TGFF = 1*2 [Setting condition] • When TCNT_0 = TGRF_0 and TGRF_0 is functioning as compare register
0	TGFE	0	R/(W)*1	Compare Match Flag E Status flag that indicates the occurrence of compare match between TCNT_0 and TGRE_0. [Clearing condition] • When 0 is written to TGFE after reading TGFE = 1*2 [Setting condition] • When TCNT_0 = TGRE_0 and TGRE_0 is functioning as compare register

Note 1. Writing 0 to this bit after reading it as 1 clears the flag.
Note 2. If the next flag is set before TGFA is cleared to 0 after reading TGFA = 1, TGFA remains 1 even when 0 is written to. In this case, read TGFA = 1 again to clear TGFA to 0.

chanhvinh.com

10.3.6 Timer Buffer Operation Transfer Mode Register (TBTM)

The TBTM registers are 8-bit readable/writable registers that specify the timing for transferring data from the buffer register to the timer general register in PWM mode. This module has three TBTM registers, one each for channels 0, 3, and 4.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	TTSE	TTSB	TTSA
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	TTSE	0	R/W	Timing Select E Specifies the timing for transferring data from TGRF_0 to TGRE_0 when they are used together for buffer operation. In channels 3 and 4, bit 2 is reserved. It is always read as 0 and the write value should always be 0. Do not set this bit to 1 when PWM mode is not selected in channel 0. 0: When compare match E occurs in channel 0 1: When TCNT_0 is cleared
1	TTSB	0	R/W	Timing Select B Specifies the timing for transferring data from TGRD to TGRB in each channel when they are used together for buffer operation. Do not set this bit to 1 when PWM mode is not selected in any channels. 0: When compare match B occurs in each channel 1: When TCNT is cleared in each channel
0	TTSA	0	R/W	Timing Select A Specifies the timing for transferring data from TGRC to TGRA in each channel when they are used together for buffer operation. Do not set this bit to 1 when PWM mode is not selected in any channels. 0: When compare match A occurs in each channel 1: When TCNT is cleared in each channel

chanhvinh.com

10.3.7 Timer Input Capture Control Register (TICCR)

TICCR is an 8-bit readable/writable register that specifies input capture conditions when TCNT_1 and TCNT_2 are cascaded. This module has one TICCR in channel 1.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	I2BE	I2AE	I1BE	I1AE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	I2BE	0	R/W	Input Capture Enable Specifies whether to include the TIOC2B pin in the TGRB_1 input capture conditions. 0: Does not include the TIOC2B pin in the TGRB_1 input capture conditions 1: Includes the TIOC2B pin in the TGRB_1 input capture conditions
2	I2AE	0	R/W	Input Capture Enable Specifies whether to include the TIOC2A pin in the TGRA_1 input capture conditions. 0: Does not include the TIOC2A pin in the TGRA_1 input capture conditions 1: Includes the TIOC2A pin in the TGRA_1 input capture conditions
1	I1BE	0	R/W	Input Capture Enable Specifies whether to include the TIOC1B pin in the TGRB_2 input capture conditions. 0: Does not include the TIOC1B pin in the TGRB_2 input capture conditions 1: Includes the TIOC1B pin in the TGRB_2 input capture conditions
0	I1AE	0	R/W	Input Capture Enable Specifies whether to include the TIOC1A pin in the TGRA_2 input capture conditions. 0: Does not include the TIOC1A pin in the TGRA_2 input capture conditions 1: Includes the TIOC1A pin in the TGRA_2 input capture conditions

chanhvinh.com

10.3.8 Timer A/D Converter Start Request Control Register (TADCR)

TADCR is a 16-bit readable/writable register that enables or disables A/D converter start requests and specifies whether to link A/D converter start requests with interrupt skipping operation. This module has one TADCR in channel 4.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BF[1:0]		-	-	-	-	-	-	UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE
Initial value:	0	0	0	0	0	0	0	0	0	0*	0	0*	0*	0*	0*	0*
R/W:	R/W	R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	BF[1:0]	00	R/W	TADCOBRA_4/TADCOBRB_4 Transfer Timing Select Select the timing for transferring data from TADCOBRA_4 and TADCOBRB_4 to TADCORA_4 and TADCORB_4. For details, see Table 10.27.
13 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	UT4AE	0	R/W	Up-Count TRG4AN Enable Enables or disables A/D converter start requests (TRG4AN) during TCNT_4 up-count operation. 0: A/D converter start requests (TRG4AN) disabled during TCNT_4 up-count operation 1: A/D converter start requests (TRG4AN) enabled during TCNT_4 up-count operation
6	DT4AE	0*	R/W	Down-Count TRG4AN Enable Enables or disables A/D converter start requests (TRG4AN) during TCNT_4 down-count operation. 0: A/D converter start requests (TRG4AN) disabled during TCNT_4 down-count operation 1: A/D converter start requests (TRG4AN) enabled during TCNT_4 down-count operation
5	UT4BE	0	R/W	Up-Count TRG4BN Enable Enables or disables A/D converter start requests (TRG4BN) during TCNT_4 up-count operation. 0: A/D converter start requests (TRG4BN) disabled during TCNT_4 up-count operation 1: A/D converter start requests (TRG4BN) enabled during TCNT_4 up-count operation
4	DT4BE	0*	R/W	Down-Count TRG4BN Enable Enables or disables A/D converter start requests (TRG4BN) during TCNT_4 down-count operation. 0: A/D converter start requests (TRG4BN) disabled during TCNT_4 down-count operation 1: A/D converter start requests (TRG4BN) enabled during TCNT_4 down-count operation
3	ITA3AE	0*	R/W	TGIA_3 Interrupt Skipping Link Enable Select whether to link A/D converter start requests (TRG4AN) with TGIA_3 interrupt skipping operation. 0: Does not link with TGIA_3 interrupt skipping 1: Links with TGIA_3 interrupt skipping
2	ITA4VE	0*	R/W	TCIV_4 Interrupt Skipping Link Enable Select whether to link A/D converter start requests (TRG4AN) with TCIV_4 interrupt skipping operation. 0: Does not link with TCIV_4 interrupt skipping 1: Links with TCIV_4 interrupt skipping

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
1	ITB3AE	0*	R/W	TGIA_3 Interrupt Skipping Link Enable Select whether to link A/D converter start requests (TRG4BN) with TGIA_3 interrupt skipping operation. 0: Does not link with TGIA_3 interrupt skipping 1: Links with TGIA_3 interrupt skipping
0	ITB4VE	0*	R/W	TCIV_4 Interrupt Skipping Link Enable Select whether to link A/D converter start requests (TRG4BN) with TCIV_4 interrupt skipping operation. 0: Does not link with TCIV_4 interrupt skipping 1: Links with TCIV_4 interrupt skipping

Note 1. TADCR must not be accessed in eight bits; it should always be accessed in 16 bits.
Note 2. When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) are cleared to 0 or the skipping count set bits (3ACOR and 4VCOR) in TITCR are cleared to 0), do not link A/D converter start requests with interrupt skipping operation (clear the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR) to 0).
Note 3. If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D converter start requests will not be issued.
Note: * Do not set to 1 when complementary PWM mode is not selected.

Table 10.27 Setting of Transfer Timing by Bits BF1 and BF0

Bit 7	Bit 6	
BF1	BF0	Description
0	0	Does not transfer data from the cycle set buffer register to the cycle set register.
0	1	Transfers data from the cycle set buffer register to the cycle set register at the crest of the TCNT_4 count.*1
1	0	Transfers data from the cycle set buffer register to the cycle set register at the trough of the TCNT_4 count.*2
1	1	Transfers data from the cycle set buffer register to the cycle set register at the crest and trough of the TCNT_4 count.*2

Note 1. Data is transferred from the cycle set buffer register to the cycle set register when the crest of the TCNT_4 count is reached in complementary PWM mode, when compare match occurs between TCNT_3 and TGRA_3 in reset-synchronized PWM mode, or when compare match occurs between TCNT_4 and TGRA_4 in PWM mode 1 or normal operation mode.
Note 2. These settings are prohibited when complementary PWM mode is not selected.

chanhvinh.com

10.3.9 Timer A/D Converter Start Request Cycle Set Registers (TADCORA_4 and TADCORB_4)

TADCORA_4 and TADCORB_4 are 16-bit readable/writable registers. When the TCNT_4 count reaches the value in TADCORA_4 or TADCORB_4, a corresponding A/D converter start request will be issued.

TADCORA_4 and TADCORB_4 are initialized to H'FFFF.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: TADCORA_4 and TADCORB_4 must not be accessed in eight bits; they should always be accessed in 16 bits.

10.3.10 Timer A/D Converter Start Request Cycle Set Buffer Registers (TADCOBRA_4 and TADCOBRB_4)

TADCOBRA_4 and TADCOBRB_4 are 16-bit readable/writable registers. When the crest or trough of the TCNT_4 count is reached, these register values are transferred to TADCORA_4 and TADCORB_4, respectively.

TADCOBRA_4 and TADCOBRB_4 are initialized to H'FFFF.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: TADCOBRA_4 and TADCOBRB_4 must not be accessed in eight bits; they should always be accessed in 16 bits.

10.3.11 Timer Counter (TCNT)

The TCNT counters are 16-bit readable/writable counters. This module has five TCNT counters, one each for channels 0 to 4.

The TCNT counters must not be accessed in eight bits; they should always be accessed in 16 bits.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: The TCNT counters must not be accessed in eight bits; they should always be accessed in 16 bits.

chanhvinh.com

10.3.12 Timer General Register (TGR)

The TGR registers are 16-bit readable/writable registers. This module has eighteen TGR registers, six for channel 0, two each for channels 1 and 2, four each for channels 3 and 4.

TGRA, TGRB, TGRC, and TGRD function as either output compare or input capture registers. TGRC and TGRD for channels 0, 3, and 4 can also be designated for operation as buffer registers. TGR buffer register combinations are TGRA and TGRC, and TGRB and TGRD.

TGRE_0 and TGRF_0 function as compare registers. When the TCNT_0 count matches the TGRE_0 value, an A/D converter start request can be issued. TGRF can also be designated for operation as a buffer register. TGR buffer register combination is TGRE and TGRF.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: The TGR registers must not be accessed in eight bits; they should always be accessed in 16 bits.
TGR registers are initialized to H'FFFF.

chanhvinh.com

10.3.13 Timer Start Register (TSTR)

TSTR is an 8-bit readable/writable register that selects operation/stoppage of TCNT for channels 0 to 4.

When setting the operating mode in TMDR or setting the count clock in TCR, first stop the TCNT counter.

Bit:	7	6	5	4	3	2	1	0
	CST4	CST3	-	-	-	CST2	CST1	CST0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	CST4	0	R/W	Counter Start 4 and 3 These bits select operation or stoppage for TCNT. If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the TIOC pin output compare output level is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value. 0: TCNT_4 and TCNT_3 count operation is stopped 1: TCNT_4 and TCNT_3 performs count operation
6	CST3	0	R/W	
5 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	CST2	0	R/W	Counter Start 2 to 0 These bits select operation or stoppage for TCNT. If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the TIOC pin output compare output level is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value. 0: TCNT_2 to TCNT_0 count operation is stopped 1: TCNT_2 to TCNT_0 performs count operation
1	CST1	0	R/W	
0	CST0	0	R/W	

chanhvinh.com

10.3.14 Timer Synchronous Register (TSYR)

TSYR is an 8-bit readable/writable register that selects independent operation or synchronous operation for the channel 0 to 4 TCNT counters. A channel performs synchronous operation when the corresponding bit in TSYR is set to 1.

Bit:	7	6	5	4	3	2	1	0
	SYNC4	SYNC3	-	-	-	SYNC2	SYNC1	SYNC0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	SYNC4	0	R/W	Timer Synchronous operation 4 and 3 These bits are used to select whether operation is independent of or synchronized with other channels. When synchronous operation is selected, the TCNT synchronous presetting of multiple channels, and synchronous clearing by counter clearing on another channel, are possible. To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit , the TCNT clearing source must also be set by means of bits CCLR0 to CCLR2 in TCR. 0: TCNT_4 and TCNT_3 operate independently (TCNT presetting/clearing is unrelated to other channels) 1: TCNT_4 and TCNT_3 performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible
6	SYNC3	0	R/W	
5 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	SYNC2	0	R/W	Timer Synchronous operation 2 to 0 These bits are used to select whether operation is independent of or synchronized with other channels. When synchronous operation is selected, the TCNT synchronous presetting of multiple channels, and synchronous clearing by counter clearing on another channel, are possible. To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of bits CCLR0 to CCLR2 in TCR. 0: TCNT_2 to TCNT_0 operates independently (TCNT presetting /clearing is unrelated to other channels) 1: TCNT_2 to TCNT_0 performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible
1	SYNC1	0	R/W	
0	SYNC0	0	R/W	

chanhvinh.com

10.3.15 Timer Read/Write Enable Register (TRWER)

TRWER is an 8-bit readable/writable register that enables or disables access to the registers and counters which have write-protection capability against accidental modification in channels 3 and 4.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	RWE
Initial value:	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	RWE	1	R/W	Read/Write Enable Enables or disables access to the registers which have write-protection capability against accidental modification. 0: Disables read/write access to the registers 1: Enables read/write access to the registers [Clearing condition] • When 0 is written to the RWE bit after reading RWE = 1

- Registers and counters having write-protection capability against accidental modification
  22 registers: TCR_3, TCR_4, TMDR_3, TMDR_4, TIORH_3, TIORH_4, TIORL_3, TIORL_4, TIER_3, TIER_4, TGRA_3, TGRA_4, TGRB_3, TGRB_4, TOER, TOCR1, TOCR2, TGCR, TCDR, TDDR, TCNT_3, and TCNT_4.

chanhvinh.com

10.3.16 Timer Output Master Enable Register (TOER)

TOER is an 8-bit readable/writable register that enables/disables output settings for output pins TIOC4D, TIOC4C, TIOC3D, TIOC4B, TIOC4A, and TIOC3B. These pins do not output correctly if the TOER bits have not been set. Set TOER of CH3 and CH4 prior to setting TIOR of CH3 and CH4.

Make settings of the TOER while counting by the TCNT registers of channels 3 and 4 is stopped.

Bit:	7	6	5	4	3	2	1	0
	-	-	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
Initial value:	1	1	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
5	OE4D	0	R/W	Master Enable TIOC4D This bit enables/disables the TIOC4D pin output for this module. 0: Output for this module is disabled (inactive level)* 1: Output for this module is enabled
4	OE4C	0	R/W	Master Enable TIOC4C This bit enables/disables the TIOC4C pin output for this module. 0: Output for this module is disabled (inactive level)* 1: Output for this module is enabled
3	OE3D	0	R/W	Master Enable TIOC3D This bit enables/disables the TIOC3D pin output for this module. 0: Output for this module is disabled (inactive level)* 1: Output for this module is enabled
2	OE4B	0	R/W	Master Enable TIOC4B This bit enables/disables the TIOC4B pin output for this module. 0: Output for this module is disabled (inactive level)* 1: Output for this module is enabled
1	OE4A	0	R/W	Master Enable TIOC4A This bit enables/disables the TIOC4A pin output for this module. 0: Output for this module is disabled (inactive level)* 1: Output for this module is enabled
0	OE3B	0	R/W	Master Enable TIOC3B This bit enables/disables the TIOC3B pin output for this module. 0: Output for this module is disabled (inactive level)* 1: Output for this module is enabled

Note: * The inactive level is determined by the settings in timer output control registers 1 and 2 (TOCR1 and TOCR2). For details, refer to section 10.3.17, Timer Output Control Register 1 (TOCR1), and section 10.3.18, Timer Output Control Register 2 (TOCR2). Set these bits to 1 to enable output for this module in other than complementary PWM or reset-synchronized PWM mode. When these bits are set to 0, low level is output.

chanhvinh.com

10.3.17 Timer Output Control Register 1 (TOCR1)

TOCR1 is an 8-bit readable/writable register that enables/disables PWM synchronized toggle output in complementary PWM mode/reset synchronized PWM mode, and controls output level inversion of PWM output.

Bit:	7	6	5	4	3	2	1	0
	-	PSYE	-	-	TOCL	TOCS	OLSN	OLSP
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R/(W)*3	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	PSYE	0	R/W	PWM Synchronous Output Enable This bit selects the enable/disable of toggle output synchronized with the PWM period. 0: Toggle output is disabled 1: Toggle output is enabled
5, 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	TOCL	0	R/(W)*3	TOC Register Write Protection*1 This bit selects the enable/disable of write access to the TOCS, OLSN, and OLSP bits in TOCR1. 0: Write access to the TOCS, OLSN, and OLSP bits is enabled 1: Write access to the TOCS, OLSN, and OLSP bits is disabled
2	TOCS	0	R/W	TOC Select This bit selects either the TOCR1 or TOCR2 setting to be used for the output level in complementary PWM mode and reset-synchronized PWM mode. 0: TOCR1 setting is selected 1: TOCR2 setting is selected
1	OLSN	0	R/W	Output Level Select N*2*4 This bit selects the negative phase output level in reset-synchronized PWM mode/complementary PWM mode. See Table 10.28.
0	OLSP	0	R/W	Output Level Select P*2 This bit selects the positive phase output level in reset-synchronized PWM mode/complementary PWM mode. See Table 10.29.

Note 1. Setting the TOCL bit to 1 prevents accidental modification when the CPU goes out of control.
Note 2. Clearing the TOCS bit to 0 makes this bit setting valid.
Note 3. After power-on reset, 1 can be written only once. After 1 has been written, 0 cannot be written.
Note 4. If the dead-time is not generated, the negative-phase output will be the exact inverse of the positive-phase output. Furthermore, set OLSP and OLSN to the same value.

chanhvinh.com

Table 10.28 Output Level Select Function

Bit 1	Function			
			Compare Match Output	
OLSN	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: The negative phase waveform initial output value changes to active level after elapse of the dead time after count start.

Table 10.29 Output Level Select Function

Bit 0	Function			
			Compare Match Output	
OLSP	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Figure 10.2 shows an example of complementary PWM mode output (1 phase) when OLSN = 1, OLSP = 1.

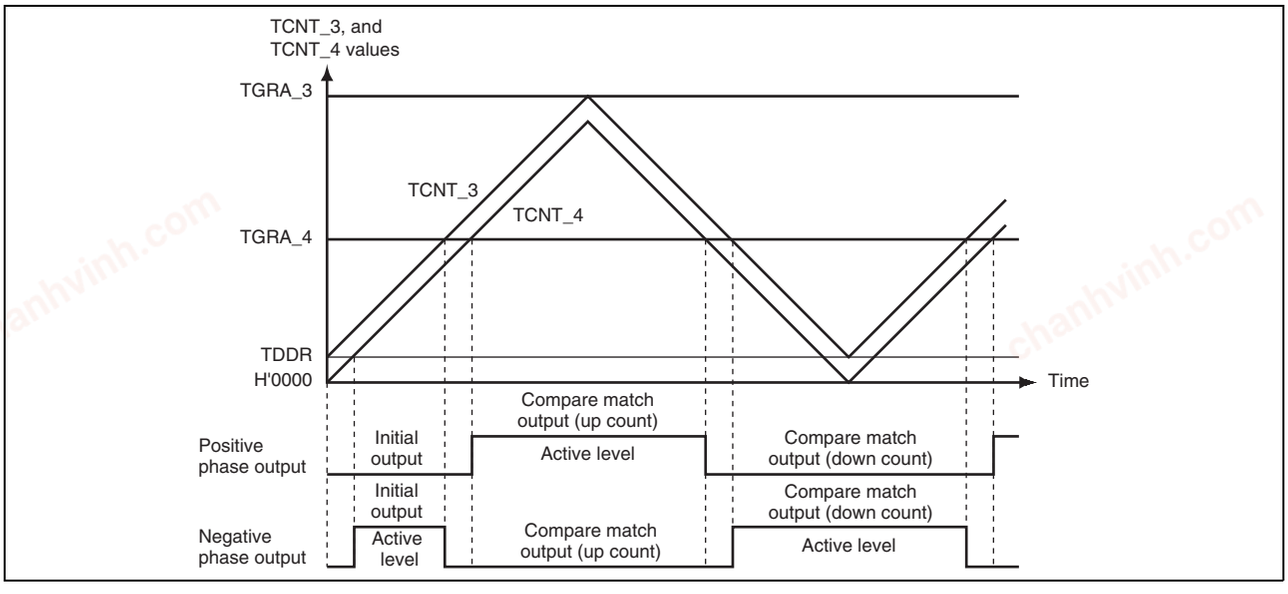


Figure 10.2 Complementary PWM Mode Output Level Example

chanhvinh.com

10.3.18 Timer Output Control Register 2 (TOCR2)

TOCR2 is an 8-bit readable/writable register that controls output level inversion of PWM output in complementary PWM mode and reset-synchronized PWM mode.

Bit:	7	6	5	4	3	2	1	0
	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7, 6	BF[1:0]	00	R/W	TOLBR Buffer Transfer Timing Select These bits select the timing for transferring data from TOLBR to TOCR2. For details, see Table 10.30.
5	OLS3N	0	R/W	Output Level Select 3N* This bit selects the output level on TIOC4D in reset-synchronized PWM mode/ complementary PWM mode. See Table 10.31.
4	OLS3P	0	R/W	Output Level Select 3P* This bit selects the output level on TIOC4B in reset-synchronized PWM mode/ complementary PWM mode. See Table 10.32.
3	OLS2N	0	R/W	Output Level Select 2N* This bit selects the output level on TIOC4C in reset-synchronized PWM mode/ complementary PWM mode. See Table 10.33.
2	OLS2P	0	R/W	Output Level Select 2P* This bit selects the output level on TIOC4A in reset-synchronized PWM mode/ complementary PWM mode. See Table 10.34.
1	OLS1N	0	R/W	Output Level Select 1N* This bit selects the output level on TIOC3D in reset-synchronized PWM mode/ complementary PWM mode. See Table 10.35.
0	OLS1P	0	R/W	Output Level Select 1P* This bit selects the output level on TIOC3B in reset-synchronized PWM mode/ complementary PWM mode. See Table 10.36.

Note: * Setting the TOCS bit in TOCR1 to 1 makes this bit setting valid.
If the dead-time is not generated, the negative-phase output will be the exact inverse of the positive-phase output. Furthermore, set OLSiP and OLSiN (i = 1, 2, 3) to the same value.

chanhvinh.com

Table 10.30 Setting of Bits BF1 and BF0

Bit 7	Bit 6	Description	
BF1	BF0	Complementary PWM Mode	Reset-Synchronized PWM Mode
0	0	Does not transfer data from the buffer register (TOLBR) to TOCR2.	Does not transfer data from the buffer register (TOLBR) to TOCR2.
0	1	Transfers data from the buffer register (TOLBR) to TOCR2 at the crest of the TCNT_4 count.	Transfers data from the buffer register (TOLBR) to TOCR2 when TCNT_3/TCNT_4 is cleared
1	0	Transfers data from the buffer register (TOLBR) to TOCR2 at the trough of the TCNT_4 count.	Setting prohibited
1	1	Transfers data from the buffer register (TOLBR) to TOCR2 at the crest and trough of the TCNT_4 count.	Setting prohibited

Table 10.31 TIOC4D Output Level Select Function

Bit 5	Function			
			Compare Match Output	
OLS3N	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: The negative phase waveform initial output value changes to the active level after elapse of the dead time after count start.

Table 10.32 TIOC4B Output Level Select Function

Bit 4	Function			
			Compare Match Output	
OLS3P	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Table 10.33 TIOC4C Output Level Select Function

Bit 3	Function			
			Compare Match Output	
OLS2N	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: The negative phase waveform initial output value changes to the active level after elapse of the dead time after count start.

Table 10.34 TIOC4A Output Level Select Function

Bit 2	Function			
			Compare Match Output	
OLS2P	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

chanhvinh.com

Table 10.35 TIOC3D Output Level Select Function

Bit 1	Function			
			Compare Match Output	
OLS1N	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: The negative phase waveform initial output value changes to the active level after elapse of the dead time after count start.

Table 10.36 TIOC4B Output Level Select Function

Bit 0	Function			
			Compare Match Output	
OLS1P	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

chanhvinh.com

10.3.19 Timer Output Level Buffer Register (TOLBR)

TOLBR is an 8-bit readable/writable register that functions as a buffer for TOCR2 and specifies the PWM output level in complementary PWM mode and reset-synchronized PWM mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	OLS3N	0	R/W	Specifies the buffer value to be transferred to the OLS3N bit in TOCR2.
4	OLS3P	0	R/W	Specifies the buffer value to be transferred to the OLS3P bit in TOCR2.
3	OLS2N	0	R/W	Specifies the buffer value to be transferred to the OLS2N bit in TOCR2.
2	OLS2P	0	R/W	Specifies the buffer value to be transferred to the OLS2P bit in TOCR2.
1	OLS1N	0	R/W	Specifies the buffer value to be transferred to the OLS1N bit in TOCR2.
0	OLS1P	0	R/W	Specifies the buffer value to be transferred to the OLS1P bit in TOCR2.

Figure 10.3 shows an example of the PWM output level setting procedure in buffer operation.

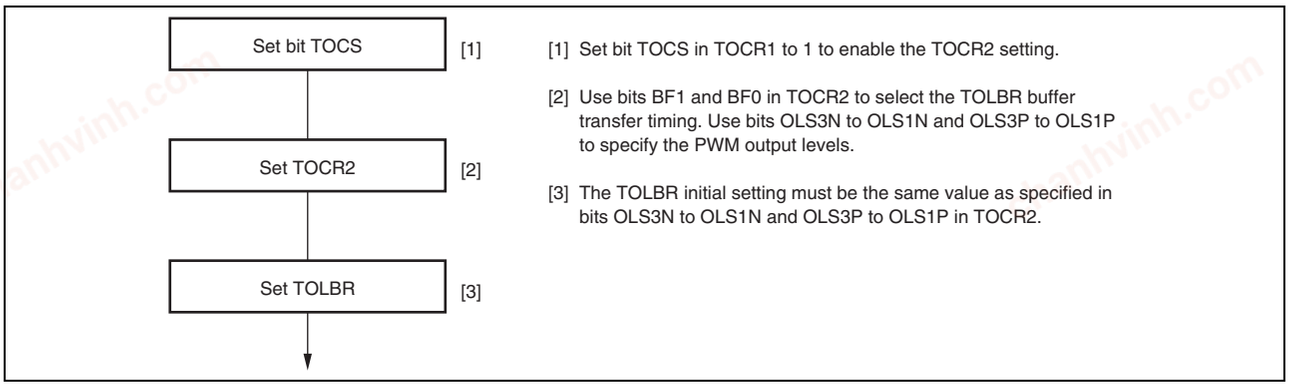


Figure 10.3 PWM Output Level Setting Procedure in Buffer Operation

chanhvinh.com

10.3.20 Timer Gate Control Register (TGCR)

TGCR is an 8-bit readable/writable register that controls the waveform output necessary for brushless DC motor control in reset-synchronized PWM mode/complementary PWM mode. These register settings are ineffective for anything other than complementary PWM mode/reset-synchronized PWM mode.

Bit:	7	6	5	4	3	2	1	0
	-	BDC	N	P	FB	WF	VF	UF
Initial value:	1	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
6	BDC	0	R/W	Brushless DC Motor This bit selects whether to make the functions of this register (TGCR) effective or ineffective. 0: Ordinary output 1: Functions of this register are made effective
5	N	0	R/W	Negative Phase Output (N) Control This bit selects whether the level output or the reset-synchronized PWM/ complementary PWM output while the reverse pins (TIOC3D, TIOC4C, and TIOC4D) are output. 0: Level output 1: Reset synchronized PWM/complementary PWM output
4	P	0	R/W	Positive Phase Output (P) Control This bit selects whether the level output or the reset-synchronized PWM/ complementary PWM output while the positive pins (TIOC3B, TIOC4A, and TIOC4B) are output. 0: Level output 1: Reset synchronized PWM/complementary PWM output
3	FB	0	R/W	External Feedback Signal Enable This bit selects whether the switching of the output of the positive/negative phase is carried out automatically with channel-0 TGRA, TGRB, TGRC input capture signals or by writing 0 or 1 to bits 2 to 0 in TGCR. 0: Output switching is external input (Input sources are channel 0 TGRA, TGRB, TGRC input capture signal) 1: Output switching is carried out by software (setting values of UF, VF, and WF in TGCR).
2	WF	0	R/W	Output Phase Switch 2 to 0 These bits set the positive phase/negative phase output phase on or off state. The setting of these bits is valid only when the FB bit in this register is set to 1. In this case, the setting of bits 2 to 0 is a substitute for external input. See Table 10.37.
1	VF	0	R/W	
0	UF	0	R/W	

chanhvinh.com

Table 10.37 Output level Select Function

			Function					
Bit 2	**Bit 1**	**Bit 0**	**TIOC3B**	**TIOC4A**	**TIOC4B**	**TIOC3D**	**TIOC4C**	**TIOC4D**
WF	**VF**	**UF**	**U Phase**	**V Phase**	**W Phase**	**U Phase**	**V Phase**	**W Phase**
0	0	0	OFF	OFF	OFF	OFF	OFF	OFF
		1	ON	OFF	OFF	OFF	OFF	ON
	1	0	OFF	ON	OFF	ON	OFF	OFF
		1	OFF	ON	OFF	OFF	OFF	ON
1	0	0	OFF	OFF	ON	OFF	ON	OFF
		1	ON	OFF	OFF	OFF	ON	OFF
	1	0	OFF	OFF	ON	ON	OFF	OFF
		1	OFF	OFF	OFF	OFF	OFF	OFF

10.3.21 Timer Subcounter (TCNTS)

TCNTS is a 16-bit read-only counter that is used only in complementary PWM mode.

The initial value of TCNTS is H'0000.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: Accessing the TCNTS in 8-bit units is prohibited. Always access in 16-bit units.

10.3.22 Timer Dead Time Data Register (TDDR)

TDDR is a 16-bit register, used only in complementary PWM mode that specifies the TCNT_3 and TCNT_4 counter offset values. In complementary PWM mode, when the TCNT_3 and TCNT_4 counters are cleared and then restarted, the TDDR register value is loaded into the TCNT_3 counter and the count operation starts.

The initial value of TDDR is H'FFFF.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: Accessing the TDDR in 8-bit units is prohibited. Always access in 16-bit units.

chanhvinh.com

10.3.23 Timer Cycle Data Register (TCDR)

TCDR is a 16-bit register used only in complementary PWM mode. Set half the PWM carrier cycle value (however, the value must also be at least twice the setting of TDDR plus 3) as the TCDR register value. This register is constantly compared with the TCNTS counter in complementary PWM mode, and when a match occurs, the TCNTS counter switches direction (decrement to increment).

The initial value of TCDR is H'FFFF.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: Accessing the TCDR in 8-bit units is prohibited. Always access in 16-bit units.

10.3.24 Timer Cycle Buffer Register (TCBR)

TCBR is a 16-bit register used only in complementary PWM mode. It functions as a buffer register for the TCDR register. The TCBR register values are transferred to the TCDR register with the transfer timing set in the TMDR register. The initial value of TCBR is H'FFFF.

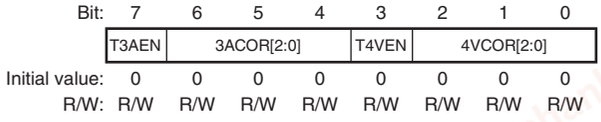
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: Accessing the TCBR in 8-bit units is prohibited. Always access in 16-bit units.

chanhvinh.com

10.3.25 Timer Interrupt Skipping Set Register (TITCR)

TITCR is an 8-bit readable/writable register that enables or disables interrupt skipping and specifies the interrupt skipping count. This module has one TITCR.



Bit	Bit Name	Initial value	R/W	Description
7	T3AEN	0	R/W	T3AEN Enables or disables TGIA_3 interrupt skipping. 0: TGIA_3 interrupt skipping disabled 1: TGIA_3 interrupt skipping enabled
6 to 4	3ACOR[2:0]	000	R/W	These bits specify the TGIA_3 interrupt skipping count within the range from 0 to 7.* For details, see Table 10.38.
3	T4VEN	0	R/W	T4VEN Enables or disables TCIV_4 interrupt skipping. 0: TCIV_4 interrupt skipping disabled 1: TCIV_4 interrupt skipping enabled
2 to 0	4VCOR[2:0]	000	R/W	These bits specify the TCIV_4 interrupt skipping count within the range from 0 to 7.* For details, see Table 10.39.

Note: * When 0 is specified for the interrupt skipping count, no interrupt skipping will be performed. Before changing the interrupt skipping count, be sure to clear the T3AEN and T4VEN bits to 0 to clear the skipping counter (TITCNT).

chanhvinh.com

Table 10.38 Setting of Interrupt Skipping Count by Bits 3ACOR2 to 3ACOR0

Bit 6	Bit 5	Bit 4	
3ACOR2	3ACOR1	3ACOR0	Description
0	0	0	Does not skip TGIA_3 interrupts.
0	0	1	Sets the TGIA_3 interrupt skipping count to 1.
0	1	0	Sets the TGIA_3 interrupt skipping count to 2.
0	1	1	Sets the TGIA_3 interrupt skipping count to 3.
1	0	0	Sets the TGIA_3 interrupt skipping count to 4.
1	0	1	Sets the TGIA_3 interrupt skipping count to 5.
1	1	0	Sets the TGIA_3 interrupt skipping count to 6.
1	1	1	Sets the TGIA_3 interrupt skipping count to 7.

Table 10.39 Setting of Interrupt Skipping Count by Bits 4VCOR2 to 4VCOR0

Bit 2	Bit 1	Bit 0	
4VCOR2	4VCOR1	4VCOR0	Description
0	0	0	Does not skip TCIV_4 interrupts.
0	0	1	Sets the TCIV_4 interrupt skipping count to 1.
0	1	0	Sets the TCIV_4 interrupt skipping count to 2.
0	1	1	Sets the TCIV_4 interrupt skipping count to 3.
1	0	0	Sets the TCIV_4 interrupt skipping count to 4.
1	0	1	Sets the TCIV_4 interrupt skipping count to 5.
1	1	0	Sets the TCIV_4 interrupt skipping count to 6.
1	1	1	Sets the TCIV_4 interrupt skipping count to 7.

chanhvinh.com

10.3.26 Timer Interrupt Skipping Counter (TITCNT)

TITCNT is an 8-bit readable counter. This module has one TITCNT. TITCNT retains its value even after stopping the count operation of TCNT_3 and TCNT_4.

Bit:	7	6	5	4	3	2	1	0
	-	3ACNT[2:0]			-	4VCNT[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0.
6 to 4	3ACNT[2:0]	000	R	TGIA_3 Interrupt Counter While the T3AEN bit in TITCR is set to 1, the count in these bits is incremented every time a TGIA_3 interrupt occurs. [Clearing conditions] • When the 3ACNT2 to 3ACNT0 value in TITCNT matches the 3ACOR2 to 3ACOR0 value in TITCR • When the T3AEN bit in TITCR is cleared to 0 • When the 3ACOR2 to 3ACOR0 bits in TITCR are cleared to 0
3	—	0	R	Reserved This bit is always read as 0.
2 to 0	4VCNT[2:0]	000	R	TCIV_4 Interrupt Counter While the T4VEN bit in TITCR is set to 1, the count in these bits is incremented every time a TCIV_4 interrupt occurs. [Clearing conditions] • When the 4VCNT2 to 4VCNT0 value in TITCNT matches the 4VCOR2 to 4VCOR0 value in TITCR • When the T4VEN bit in TITCR is cleared to 0 • When the 4VCOR2 to 4VCOR0 bits in TITCR are cleared to 0

Note: To clear the TITCNT, clear the bits T3AEN and T4VEN in TITCR to 0.

chanhvinh.com

10.3.27 Timer Buffer Transfer Set Register (TBTER)

TBTER is an 8-bit readable/writable register that enables or disables transfer from the buffer registers* used in complementary PWM mode to the temporary registers and specifies whether to link the transfer with interrupt skipping operation. This module has one TBTER.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	BTE[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	BTE[1:0]	00	R/W	These bits enable or disable transfer from the buffer registers* used in complementary PWM mode to the temporary registers and specify whether to link the transfer with interrupt skipping operation. For details, see Table 10.40.

Note: * Applicable buffer registers:
TGRC_3, TGRD_3, TGRC_4, TGRD_4, and TCBR

Table 10.40 Setting of Bits BTE1 and BTE0

Bit 1	Bit 0	
BTE1	BTE0	Description
0	0	Enables transfer from the buffer registers to the temporary registers*1 and does not link the transfer with interrupt skipping operation.
0	1	Disables transfer from the buffer registers to the temporary registers.
1	0	Links transfer from the buffer registers to the temporary registers with interrupt skipping operation.*2
1	1	Setting prohibited

Note 1. Data is transferred according to the MD3 to MD0 bit setting in TMDR. For details, refer to section 10.4.8, Complementary PWM Mode.

Note 2. When interrupt skipping is disabled (the T3AEN and T4VEN bits are cleared to 0 in the timer interrupt skipping set register (TITCR) or the skipping count set bits (3ACOR and 4VCOR) in TITCR are cleared to 0)), be sure to disable link of buffer transfer with interrupt skipping (clear the BTE1 bit in the timer buffer transfer set register (TBTER) to 0). If link with interrupt skipping is enabled while interrupt skipping is disabled, buffer transfer will not be performed.

chanhvinh.com

10.3.28 Timer Dead Time Enable Register (TDER)

TDER is an 8-bit readable/writable register that controls dead time generation in complementary PWM mode. This module has one TDER in channel 3. TDER must be modified only while TCNT stops.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	TDER
Initial value:	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R/(W)

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TDER	1	R/(W)	Dead Time Enable Specifies whether to generate dead time. 0: Does not generate dead time 1: Generates dead time* [Clearing condition] • When 0 is written to TDER after reading TDER = 1

Note: * TDDR must be set to 1 or a larger value.

chanhvinh.com

10.3.29 Timer Waveform Control Register (TWCR)

TWCR is an 8-bit readable/writable register that controls the waveform when synchronous counter clearing occurs in TCNT_3 and TCNT_4 in complementary PWM mode and specifies whether to clear the counters at TGRA_3 compare match. The CCE bit and WRE bit in TWCR must be modified only while TCNT stops.

Bit:	7	6	5	4	3	2	1	0
	CCE	-	-	-	-	-	-	WRE
Initial value:	0*	0	0	0	0	0	0	0
R/W:	R/(W)	R	R	R	R	R	R	R/(W)

Bit	Bit Name	Initial Value	R/W	Description
7	CCE	0*	R/(W)	Compare Match Clear Enable Specifies whether to clear counters at TGRA_3 compare match in complementary PWM mode. 0: Does not clear counters at TGRA_3 compare match 1: Clears counters at TGRA_3 compare match [Setting condition] • When 1 is written to CCE after reading CCE = 0
6 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	WRE	0	R/(W)	Initial Output Suppression Enable Selects the waveform output when synchronous counter clearing occurs in complementary PWM mode. The initial output is suppressed only when synchronous clearing occurs within the Tb interval at the trough in complementary PWM mode. When synchronous clearing occurs outside this interval, the initial value specified in TOCR is output regardless of the WRE bit setting. The initial value is also output when synchronous clearing occurs in the Tb interval at the trough immediately after TCNT_3 and TCNT_4 start operation. For the Tb interval at the trough in complementary PWM mode, see Figure 10.40. 0: Outputs the initial value specified in TOCR 1: Suppresses initial output [Setting condition] • When 1 is written to WRE after reading WRE = 0

Note: * Do not set to 1 when complementary PWM mode 1 is not selected.

10.3.30 Bus Master Interface

The timer counters (TCNT), general registers (TGR), timer subcounter (TCNTS), timer cycle buffer register (TCBR), timer dead time data register (TDDR), timer cycle data register (TCDR), timer A/D converter start request control register (TADCR), timer A/D converter start request cycle set registers (TADCOR), and timer A/D converter start request cycle set buffer registers (TADCOBR) are 16-bit registers. 16-bit read/writes is possible. 8-bit read/write is not possible. Always access in 16-bit units.

All registers other than the above registers are 8-bit registers. 8-bit read/writes is possible. 16-bit read/writes is not possible. Always access in 8-bit units.

chanhvinh.com

10.4 Operation

10.4.1 Basic Functions

Each channel has a TCNT and TGR register. TCNT performs up-counting, and is also capable of free-running operation, cycle counting, and external event counting.

Each TGR can be used as an input capture register or output compare register.

Always select functions for external pins of this module using the general I/O ports.

(1) Counter Operation

When one of bits CST0 to CST4 in TSTR is set to 1, the TCNT counter for the corresponding channel begins counting. TCNT can operate as a free-running counter, periodic counter, for example.

(a) Example of Count Operation Setting Procedure

Figure 10.4 shows an example of the count operation setting procedure.

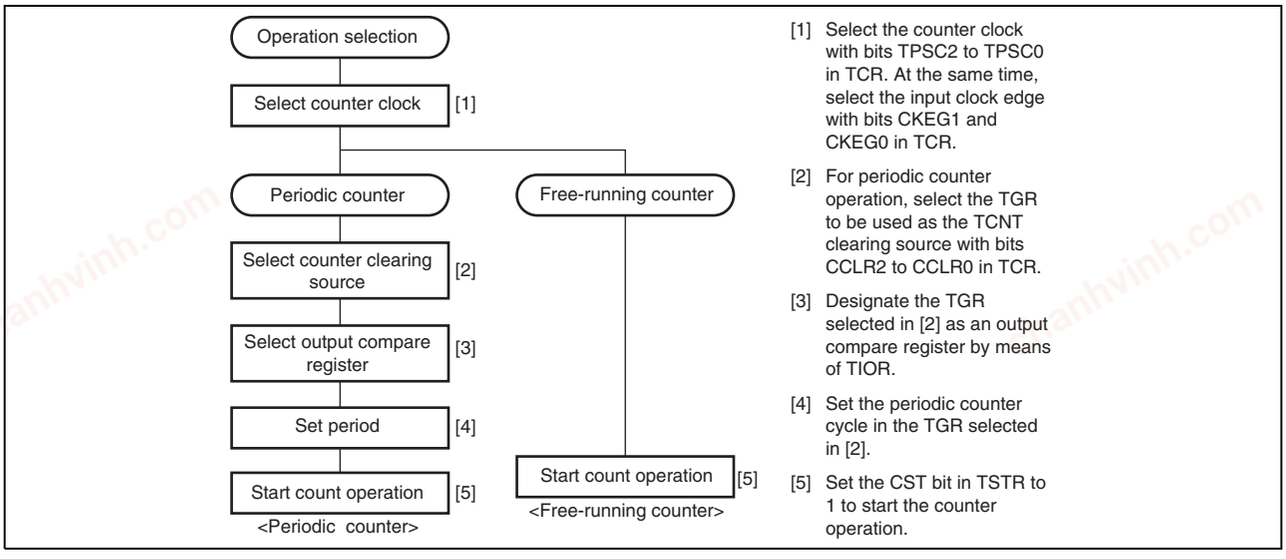


Figure 10.4 Example of Counter Operation Setting Procedure

chanhvinh.com

(b) Free-Running Count Operation and Periodic Count Operation:

Immediately after a reset, the TCNT counters of this module are all designated as free-running counters. When the relevant bit in TSTR is set to 1 the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (from H'FFFF to H'0000), the TCFV bit in TSR is set to 1. If the value of the corresponding TCIEV bit in TIER is 1 at this point, this module requests an interrupt. After overflow, TCNT starts counting up again from H'0000.

Figure 10.5 illustrates free-running counter operation.

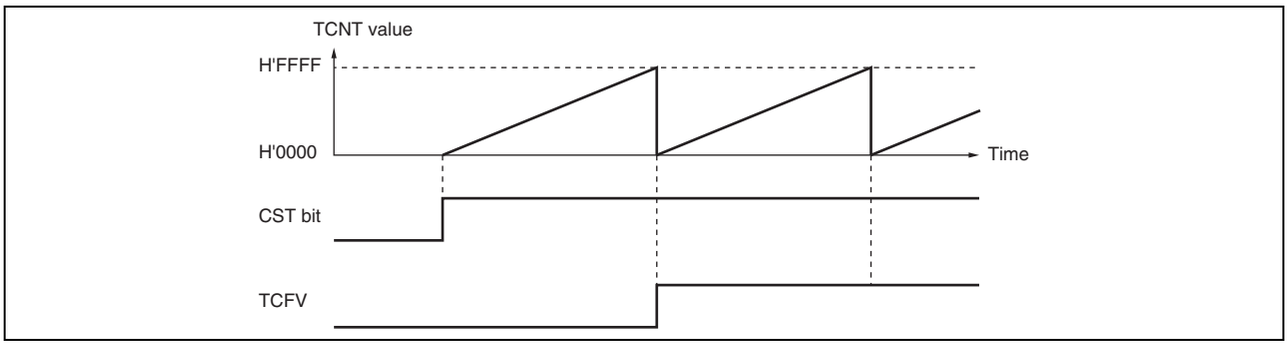


Figure 10.5 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for the relevant channel performs periodic count operation. The TGR register for setting the period is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR0 to CCLR2 in TCR. After the settings have been made, TCNT starts up-count operation as a periodic counter when the corresponding bit in TSTR is set to 1. When the count value matches the value in TGR, the TGF bit in TSR is set to 1 and TCNT is cleared to H'0000.

If the value of the corresponding TGIE bit in TIER is 1 at this point, this module requests an interrupt. After a compare match, TCNT starts counting up again from H'0000.

Figure 10.6 illustrates periodic counter operation.

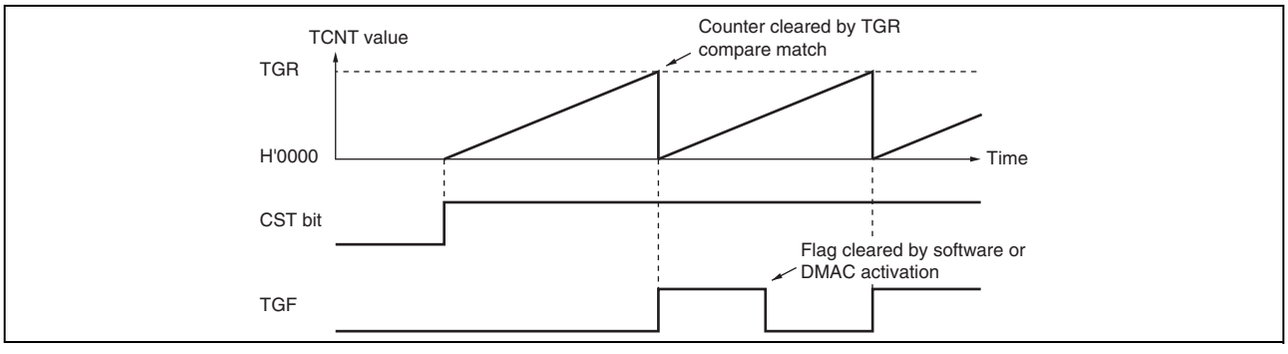


Figure 10.6 Periodic Counter Operation

chanhvinh.com

(2) Waveform Output by Compare Match

This module can perform 0, 1, or toggle output from the corresponding output pin using compare match.

(a) Example of Setting Procedure for Waveform Output by Compare Match

Figure 10.7 shows an example of the setting procedure for waveform output by compare match.

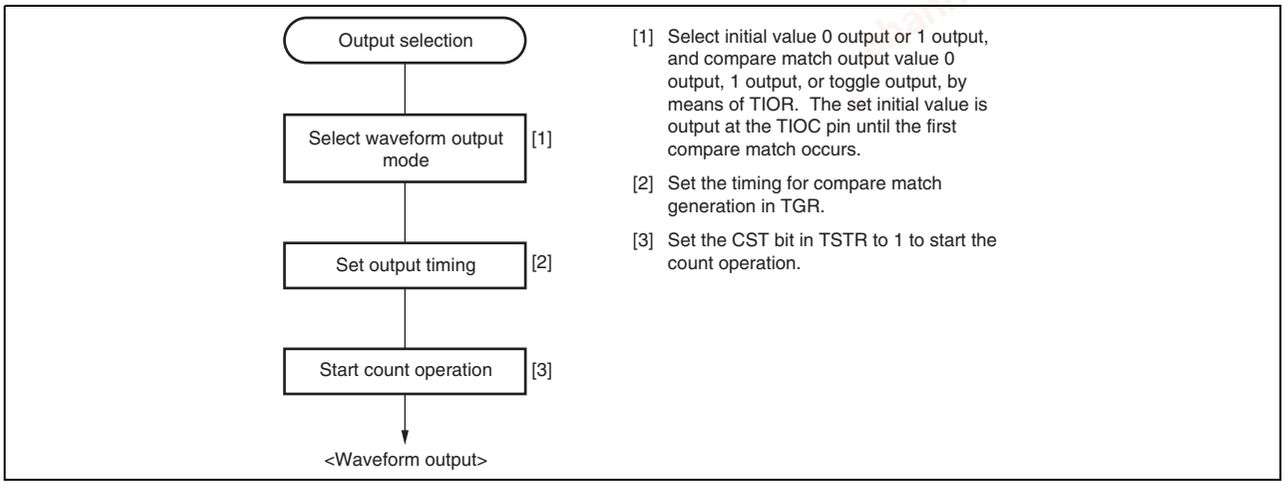


Figure 10.7 Example of Setting Procedure for Waveform Output by Compare Match

(b) Examples of Waveform Output Operation:

Figure 10.8 shows an example of 0 output/1 output.

In this example TCNT has been designated as a free-running counter, and settings have been made such that 1 is output by compare match A, and 0 is output by compare match B. When the set level and the pin level coincide, the pin level does not change.

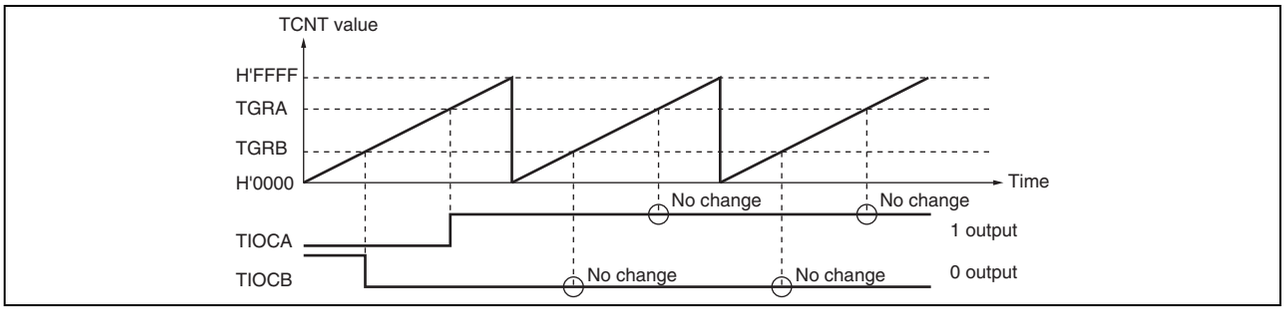


Figure 10.8 Example of 0 Output/1 Output Operation

chanhvinh.com

Figure 10.9 shows an example of toggle output.

In this example, TCNT has been designated as a periodic counter (with counter clearing on compare match B), and settings have been made such that the output is toggled by both compare match A and compare match B.

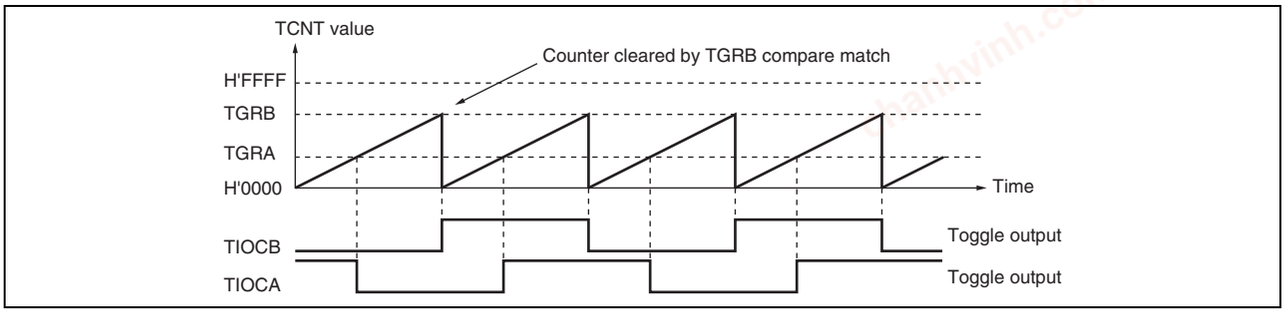


Figure 10.9 Example of Toggle Output Operation

chanhvinh.com

(3) Input Capture Function

The TCNT value can be transferred to TGR on detection of the TIOC pin input edge.

Rising edge, falling edge, or both edges can be selected as the detected edge. For channels 0 and 1, it is also possible to specify another channel's counter input clock or compare match signal as the input capture source.

Note: When another channel's counter input clock is used as the input capture input for channels 0 and 1, Pϕ/1 should not be selected as the counter input clock used for input capture input. Input capture will not be generated if Pϕ/1 is selected.

(a) Example of Input Capture Operation Setting Procedure

Figure 10.10 shows an example of the input capture operation setting procedure.

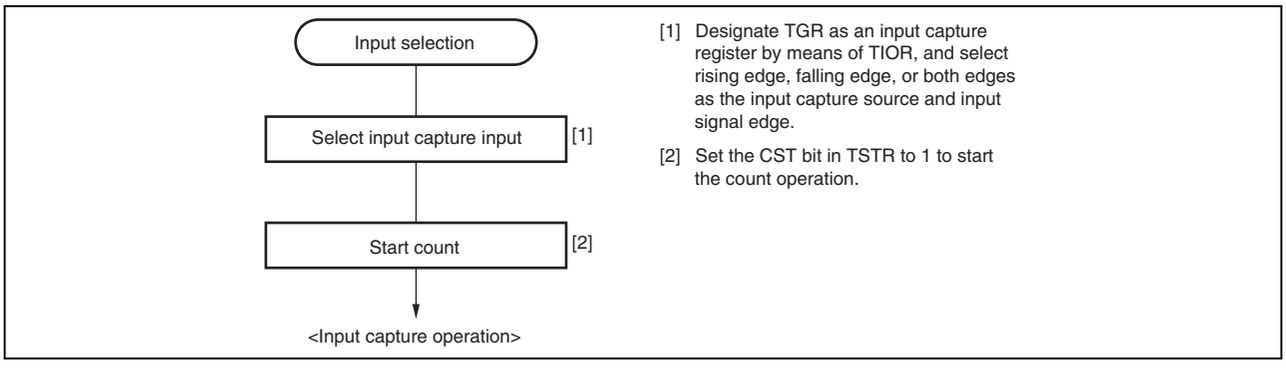


Figure 10.10 Example of Input Capture Operation Setting Procedure

chanhvinh.com

(b) Example of Input Capture Operation

Figure 10.11 shows an example of input capture operation.

In this example both rising and falling edges have been selected as the TIOCA pin input capture input edge, the falling edge has been selected as the TIOCB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT.

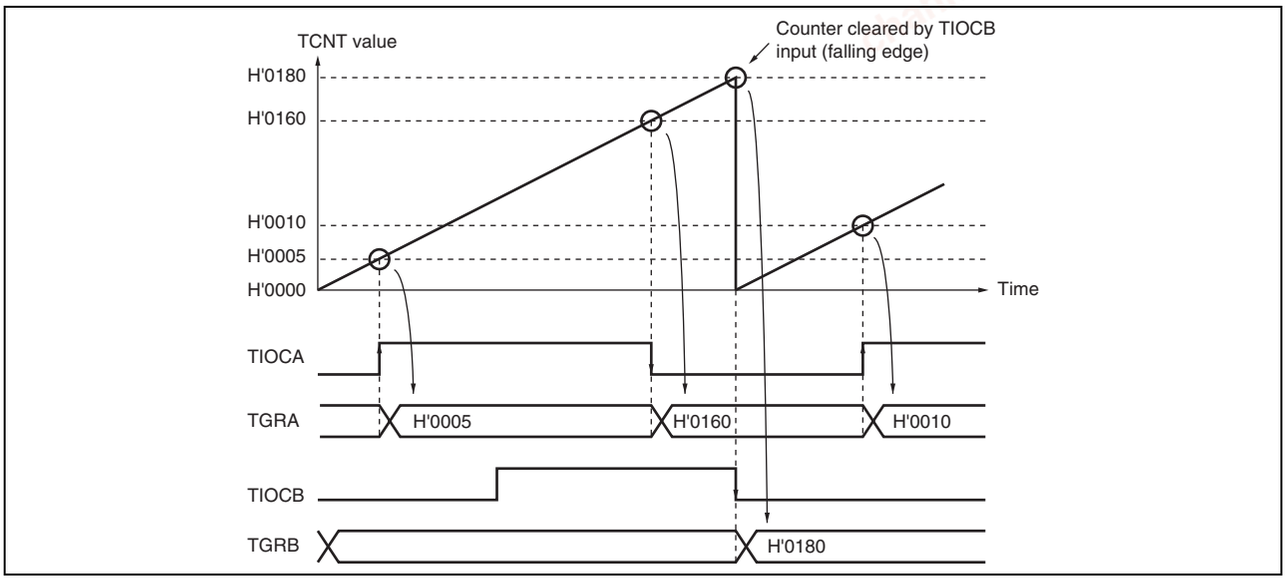


Figure 10.11 Example of Input Capture Operation

chanhvinh.com

10.4.2 Synchronous Operation

In synchronous operation, the values in a number of TCNT counters can be rewritten simultaneously (synchronous presetting). Also, a number of TCNT counters can be cleared simultaneously by making the appropriate setting in TCR (synchronous clearing).

Synchronous operation enables TGR to be incremented with respect to a single time base.

Channels 0 to 4 can all be designated for synchronous operation.

(1) Example of Synchronous Operation Setting Procedure

Figure 10.12 shows an example of the synchronous operation setting procedure.

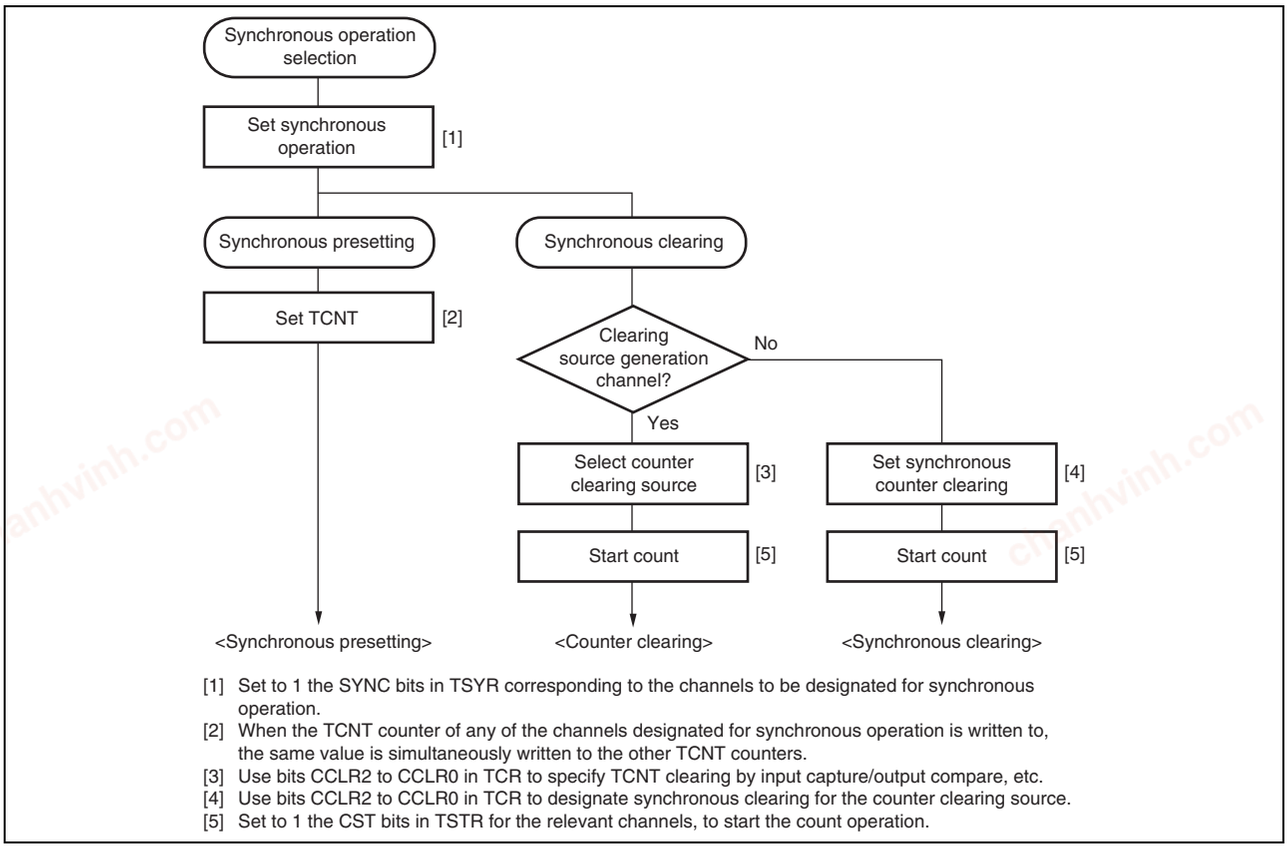


Figure 10.12 Example of Synchronous Operation Setting Procedure

chanhvinh.com

(2) Example of Synchronous Operation

Figure 10.13 shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been designated for channels 0 to 2, TGRB_0 compare match has been set as the channel 0 counter clearing source, and synchronous clearing has been set for the channel 1 and 2 counter clearing source.

Three-phase PWM waveforms are output from pins TIOC0A, TIOC1A, and TIOC2A. At this time, synchronous presetting, and synchronous clearing by TGRB_0 compare match, are performed for channel 0 to 2 TCNT counters, and the data set in TGRB_0 is used as the PWM cycle.

For details of PWM modes, see section 10.4.5, PWM Modes.

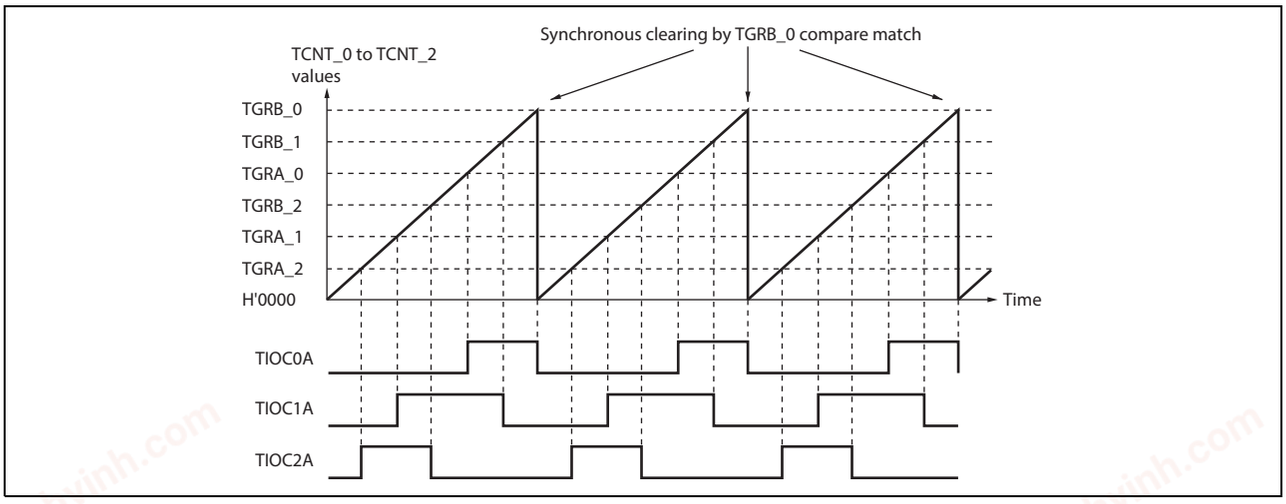


Figure 10.13 Example of Synchronous Operation

chanhvinh.com

10.4.3 Buffer Operation

Buffer operation, provided for channels 0, 3, and 4, enables TGRC and TGRD to be used as buffer registers. In channel 0, TGRF can also be used as a buffer register.

Buffer operation differs depending on whether TGR has been designated as an input capture register or as a compare match register.

Note: TGRE_0 cannot be designated as an input capture register and can only operate as a compare match register.

Table 10.41 shows the register combinations used in buffer operation.

Table 10.41 Register Combinations in Buffer Operation

Channel	Timer General Register	Buffer Register
0	TGRA_0	TGRC_0
	TGRB_0	TGRD_0
	TGRE_0	TGRF_0
3	TGRA_3	TGRC_3
	TGRB_3	TGRD_3
4	TGRA_4	TGRC_4
	TGRB_4	TGRD_4

- When TGR is an output compare register
  When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register.
  This operation is illustrated in Figure 10.14.

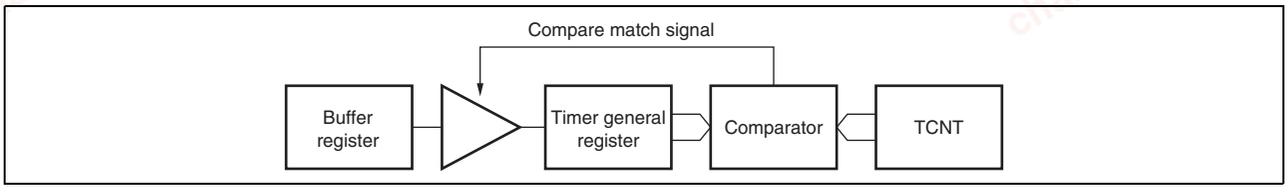


Figure 10.14 Compare Match Buffer Operation

- When TGR is an input capture register
  When input capture occurs, the value in TCNT is transferred to TGR and the value previously held in the timer general register is transferred to the buffer register.
  This operation is illustrated in Figure 10.15.

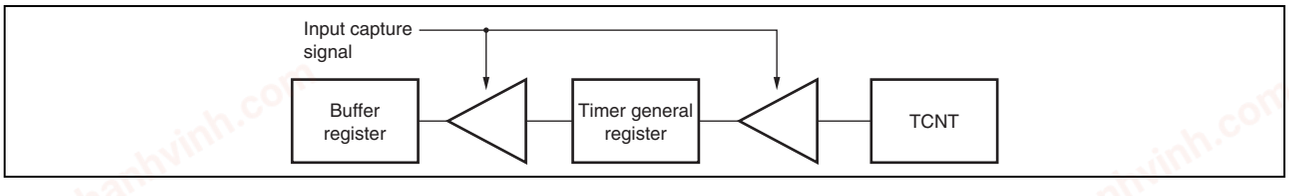


Figure 10.15 Input Capture Buffer Operation

chanhvinh.com

(1) Example of Buffer Operation Setting Procedure

Figure 10.16 shows an example of the buffer operation setting procedure.

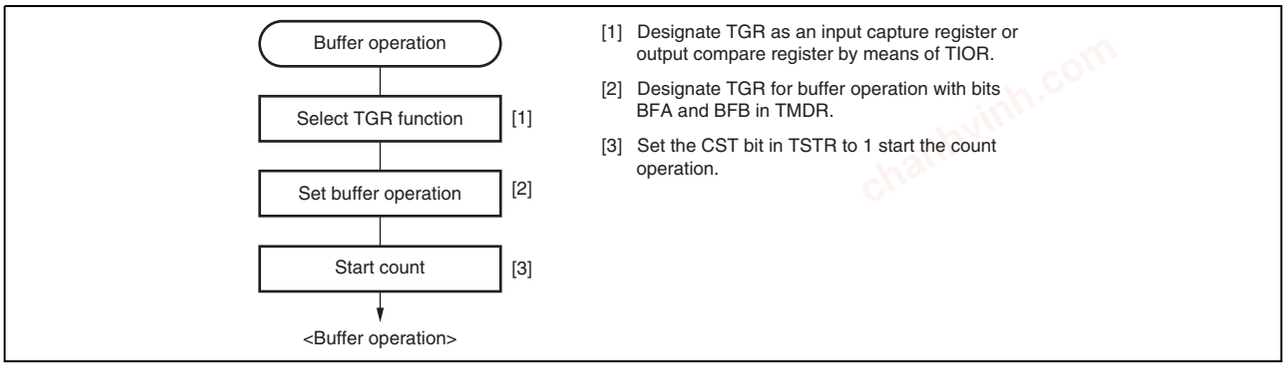


Figure 10.16 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TGR is an output compare register

Figure 10.17 shows an operation example in which PWM mode 1 has been designated for channel 0, and buffer operation has been designated for TGRA and TGRC. The settings used in this example are TCNT clearing by compare match B, 1 output at compare match A, and 0 output at compare match B. In this example, the TTSA bit in TBTM is cleared to 0.

As buffer operation has been set, when compare match A occurs the output changes and the value in buffer register TGRC is simultaneously transferred to timer general register TGRA. This operation is repeated each time that compare match A occurs.

For details of PWM modes, see section 10.4.5, PWM Modes.

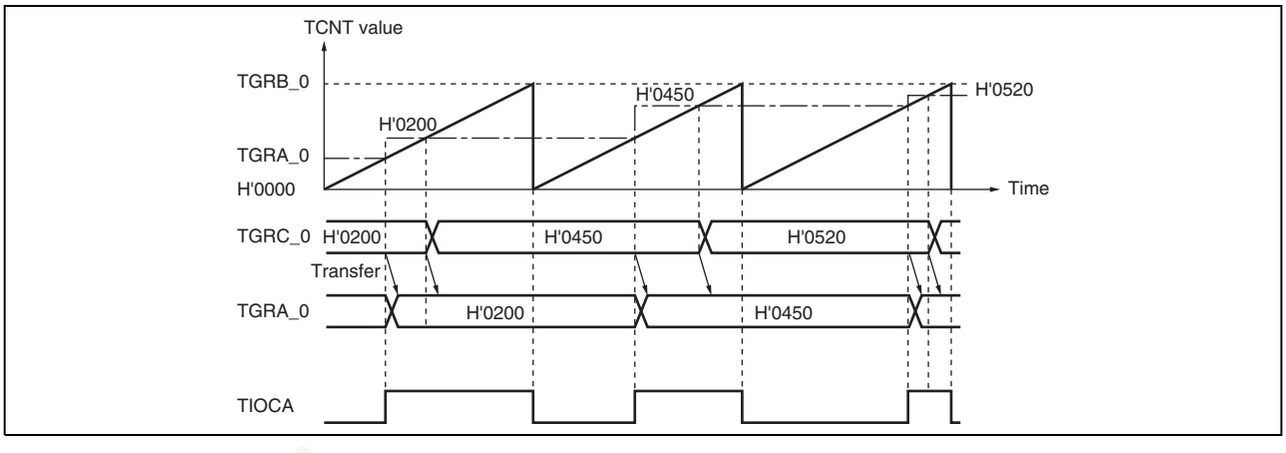


Figure 10.17 Example of Buffer Operation (1)

chanhvinh.com

(b) When TGR is an input capture register

Figure 10.18 shows an operation example in which TGRA has been designated as an input capture register, and buffer operation has been designated for TGRA and TGRC.

Counter clearing by TGRA input capture has been set for TCNT, and both rising and falling edges have been selected as the TIOCA pin input capture input edge.

As buffer operation has been set, when the TCNT value is stored in TGRA upon the occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

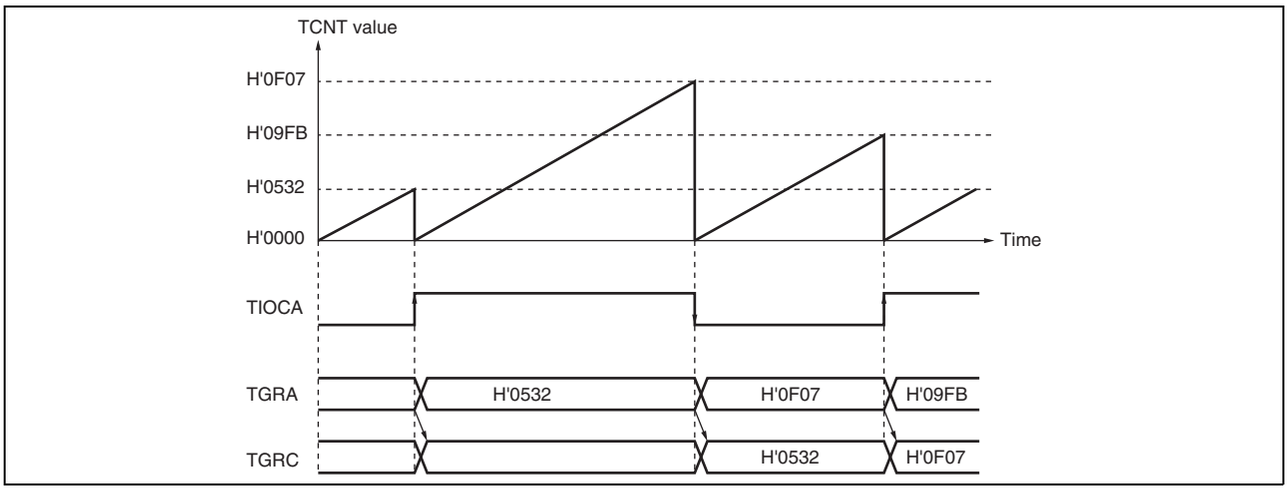


Figure 10.18 Example of Buffer Operation (2)

chanhvinh.com

(3) Selecting Timing for Transfer from Buffer Registers to Timer General Registers in Buffer Operation

The timing for transfer from buffer registers to timer general registers can be selected in PWM mode 1 or 2 for channel 0 or in PWM mode 1 for channels 3 and 4 by setting the buffer operation transfer mode registers (TBTM_0, TBTM_3, and TBTM_4). Either compare match (initial setting) or TCNT clearing can be selected for the transfer timing. TCNT clearing as transfer timing is one of the following cases.

- When TCNT overflows (H'FFFF to H'0000)
- When H'0000 is written to TCNT during counting
- When TCNT is cleared to H'0000 under the condition specified in the CCLR2 to CCLR0 bits in TCR

Note: TBTM must be modified only while TCNT stops.

Figure 10.19 shows an operation example in which PWM mode 1 is designated for channel 0 and buffer operation is designated for TGRA_0 and TGRC_0. The settings used in this example are TCNT_0 clearing by compare match B, 1 output at compare match A, and 0 output at compare match B. The TTSA bit in TBTM_0 is set to 1.

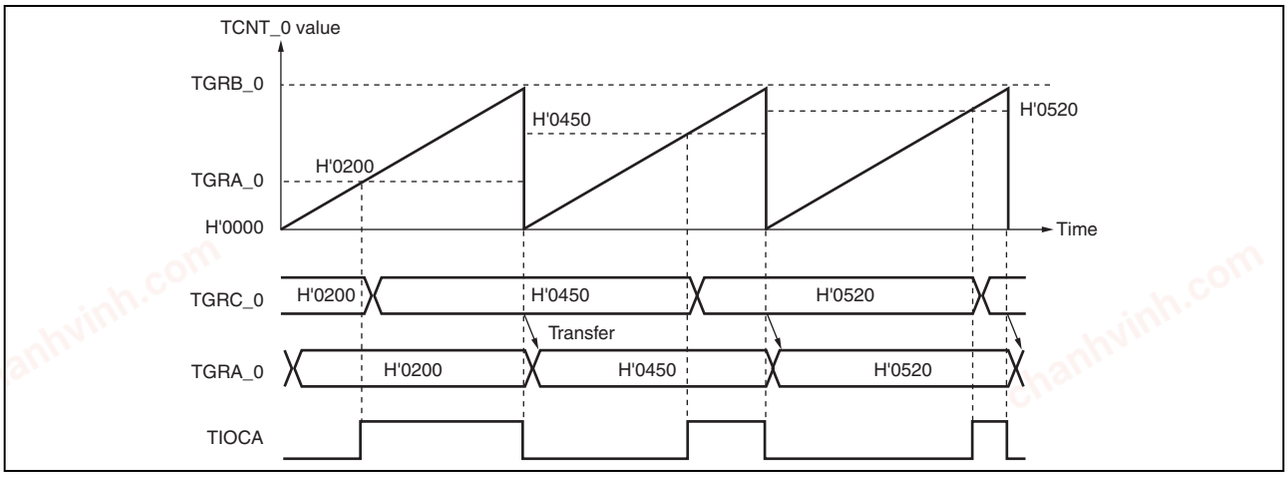


Figure 10.19 Example of Buffer Operation When TCNT_0 Clearing is Selected for TGRC_0 to TGRA_0 Transfer Timing

chanhvinh.com

10.4.4 Cascaded Operation

In cascaded operation, two 16-bit counters for different channels are used together as a 32-bit counter.

This function works by counting the channel 1 counter clock upon overflow/underflow of TCNT_2 as set in bits TPSC0 to TPSC2 in TCR.

Underflow occurs only when the lower 16-bit TCNT is in phase-counting mode.

Table 10.42 shows the register combinations used in cascaded operation.

Note: When phase counting mode is set for channel 1, the counter clock setting is invalid and the counters operates independently in phase counting mode.

Table 10.42 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
Channels 1 and 2	TCNT_1	TCNT_2

For simultaneous input capture of TCNT_1 and TCNT_2 during cascaded operation, additional input capture input pins can be specified by the input capture control register (TICCR). The condition for input capture is the detection of an edge in the signal obtained from the logical OR of the signal on the main input pin and the signal on the additional input pin. For details, see (4), Cascaded Operation Example (c). For input capture in cascade connection, refer to section 10.7.22, Simultaneous Capture of TCNT_1 and TCNT_2 in Cascade Connection.

Table 10.43 show the TICCR setting and input capture input pins.

Table 10.43 TICCR Setting and Input Capture Input Pins

Target Input Capture	TICCR Setting	Input Capture Input Pins
Input capture from TCNT_1 to TGRA_1	I2AE bit = 0 (initial value)	TIOC1A
	I2AE bit = 1	TIOC1A, TIOC2A
Input capture from TCNT_1 to TGRB_1	I2BE bit = 0 (initial value)	TIOC1B
	I2BE bit = 1	TIOC1B, TIOC2B
Input capture from TCNT_2 to TGRA_2	I1AE bit = 0 (initial value)	TIOC2A
	I1AE bit = 1	TIOC2A, TIOC1A
Input capture from TCNT_2 to TGRB_2	I1BE bit = 0 (initial value)	TIOC2B
	I1BE bit = 1	TIOC2B, TIOC1B

chanhvinh.com

(1) Example of Cascaded Operation Setting Procedure

Figure 10.20 shows an example of the setting procedure for cascaded operation.

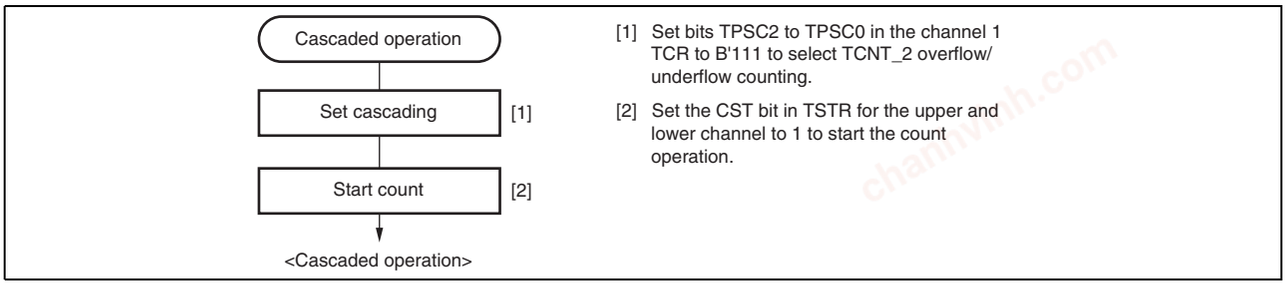


Figure 10.20 Cascaded Operation Setting Procedure

(2) Cascaded Operation Example (a)

Figure 10.21 illustrates the operation when TCNT_2 overflow/underflow counting has been set for TCNT_1 and phase counting mode has been designated for channel 2.

TCNT_1 is incremented by TCNT_2 overflow and decremented by TCNT_2 underflow.

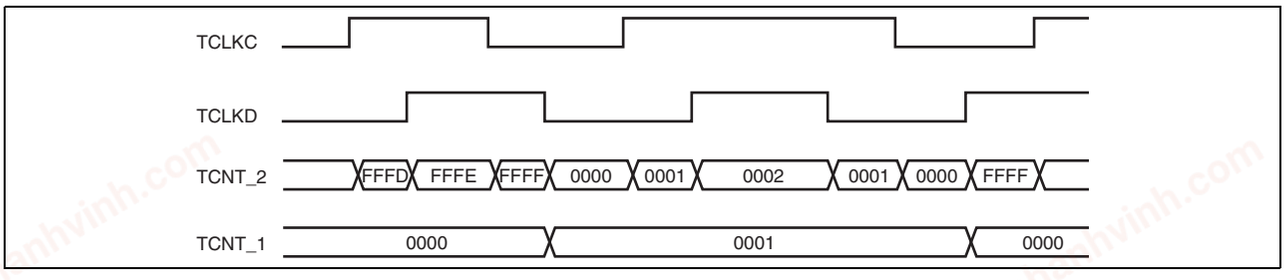


Figure 10.21 Cascaded Operation Example (a)

chanhvinh.com

(3) Cascaded Operation Example (b)

Figure 10.22 illustrates the operation when TCNT_1 and TCNT_2 have been cascaded and the I2AE bit in TICCR has been set to 1 to include the TIOC2A pin in the TGRA_1 input capture conditions. In this example, the IOA0 to IOA3 bits in TIOR_1 have selected the TIOC1A rising edge for the input capture timing while the IOA0 to IOA3 bits in TIOR_2 have selected the TIOC2A rising edge for the input capture timing.

Under these conditions, the rising edge of both TIOC1A and TIOC2A is used for the TGRA_1 input capture condition. For the TGRA_2 input capture condition, the TIOC2A rising edge is used.

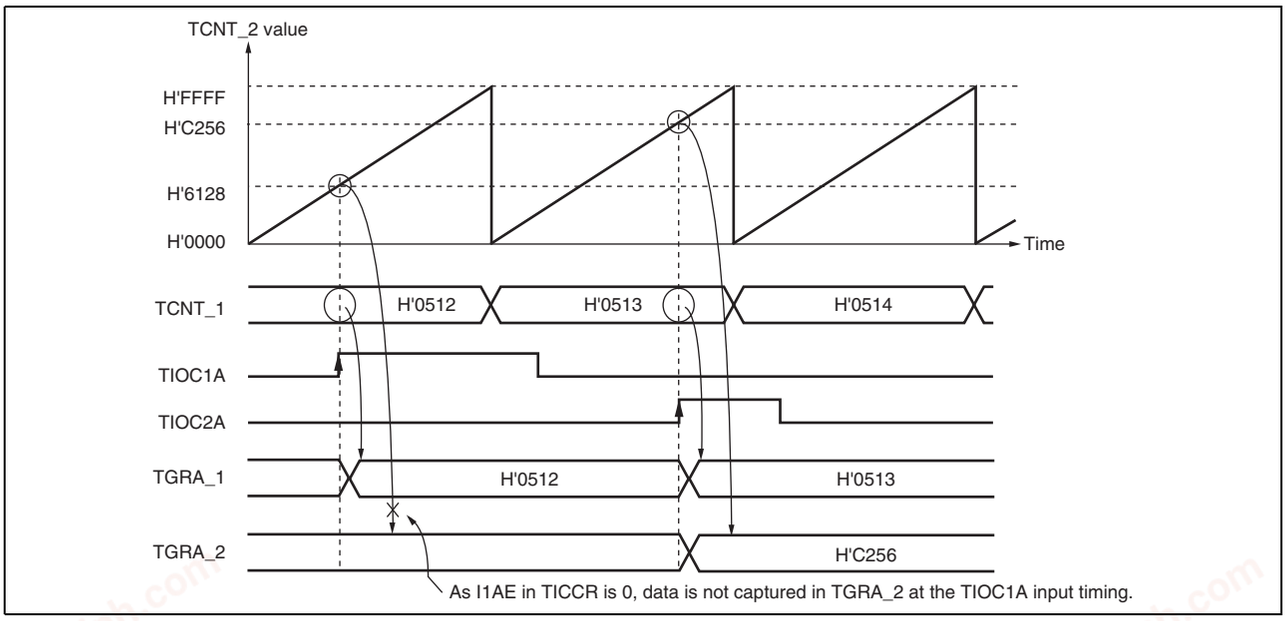


Figure 10.22 Cascaded Operation Example (b)

chanhvinh.com

(4) Cascaded Operation Example (c)

Figure 10.23 illustrates the operation when TCNT_1 and TCNT_2 have been cascaded and the I2AE and I1AE bits in TICCR have been set to 1 to include the TIOC2A and TIOC1A pins in the TGRA_1 and TGRA_2 input capture conditions, respectively. In this example, the IOA0 to IOA3 bits in both TIOR_1 and TIOR_2 have selected both the rising and falling edges for the input capture timing. Under these conditions, the ORed result of TIOC1A and TIOC2A input is used for the TGRA_1 and TGRA_2 input capture conditions.

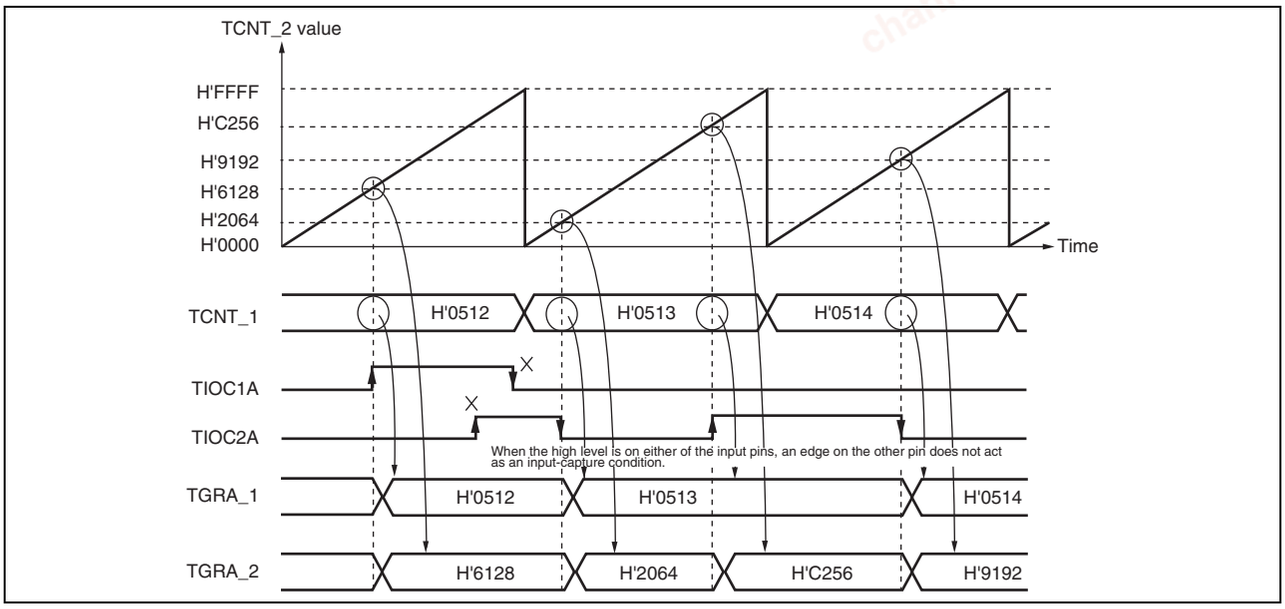


Figure 10.23 Cascaded Operation Example (c)

chanhvinh.com

(5) Cascaded Operation Example (d)

Figure 10.24 illustrates the operation when TCNT_1 and TCNT_2 have been cascaded and the I2AE bit in TICCR has been set to 1 to include the TIOC2A pin in the TGRA_1 input capture conditions. In this example, the IOA0 to IOA3 bits in TIOR_1 have selected TGRA_0 compare match or input capture occurrence for the input capture timing while the IOA0 to IOA3 bits in TIOR_2 have selected the TIOC2A rising edge for the input capture timing.

Under these conditions, as TIOR_1 has selected TGRA_0 compare match or input capture occurrence for the input capture timing, the TIOC2A edge is not used for TGRA_1 input capture condition although the I2AE bit in TICCR has been set to 1.

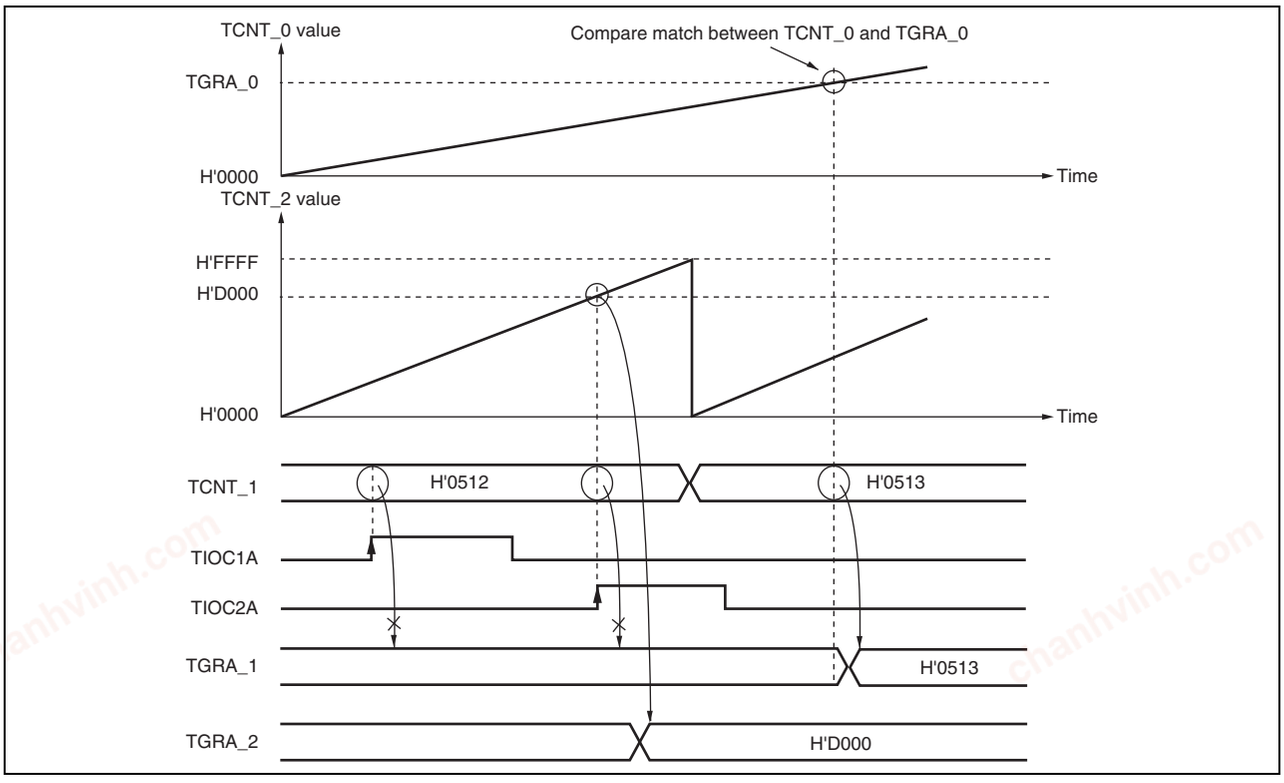


Figure 10.24 Cascaded Operation Example (d)

chanhvinh.com

10.4.5 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. The output level can be selected as 0, 1, or toggle output in response to a compare match of each TGR.

TGR registers settings can be used to output a PWM waveform in the range of 0% to 100% duty.

Designating TGR compare match as the counter clearing source enables the period to be set in that register. All channels can be designated for PWM mode independently. Synchronous operation is also possible.

There are two PWM modes, as described below.

- PWM mode 1
  PWM output is generated from the TIOCA and TIOCC pins by pairing TGRA with TGRB and TGRC with TGRD. The output specified by bits IOA0 to IOA3 and IOC0 to IOC3 in TIOR is output from the TIOCA and TIOCC pins at compare matches A and C, and the output specified by bits IOB0 to IOB3 and IOD0 to IOD3 in TIOR is output at compare matches B and D. The initial output value is the value set in TGRA or TGRC. If the set values of paired TGRs are identical, the output value does not change when a compare match occurs.
  In PWM mode 1, a maximum 8-phase PWM output is possible.
- PWM mode 2
  PWM output is generated using one TGR as the cycle register and the others as duty registers. The output specified in TIOR is performed by means of compare matches. Upon counter clearing by a cycle register compare match, the output value of each pin is the initial value set in TIOR. If the set values of the cycle and duty registers are identical, the output value does not change when a compare match occurs.
  In PWM mode 2, a maximum 8-phase PWM output is possible in combination use with synchronous operation.

The correspondence between PWM output pins and registers is shown in Table 10.44.

Table 10.44 PWM Output Registers and Output Pins

		Output Pins	
Channel	**Registers**	**PWM Mode 1**	**PWM Mode 2**
0	TGRA_0	TIOC0A	TIOC0A
	TGRB_0		TIOC0B
	TGRC_0	TIOC0C	TIOC0C
	TGRD_0		TIOC0D
1	TGRA_1	TIOC1A	TIOC1A
	TGRB_1		TIOC1B
2	TGRA_2	TIOC2A	TIOC2A
	TGRB_2		TIOC2B
3	TGRA_3	TIOC3A	Cannot be set
	TGRB_3		Cannot be set
	TGRC_3	TIOC3C	Cannot be set
	TGRD_3		Cannot be set
4	TGRA_4	TIOC4A	Cannot be set
	TGRB_4		Cannot be set
	TGRC_4	TIOC4C	Cannot be set
	TGRD_4		Cannot be set

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the period is set.

chanhvinh.com

(1) Example of PWM Mode Setting Procedure

Figure 10.25 shows an example of the PWM mode setting procedure.

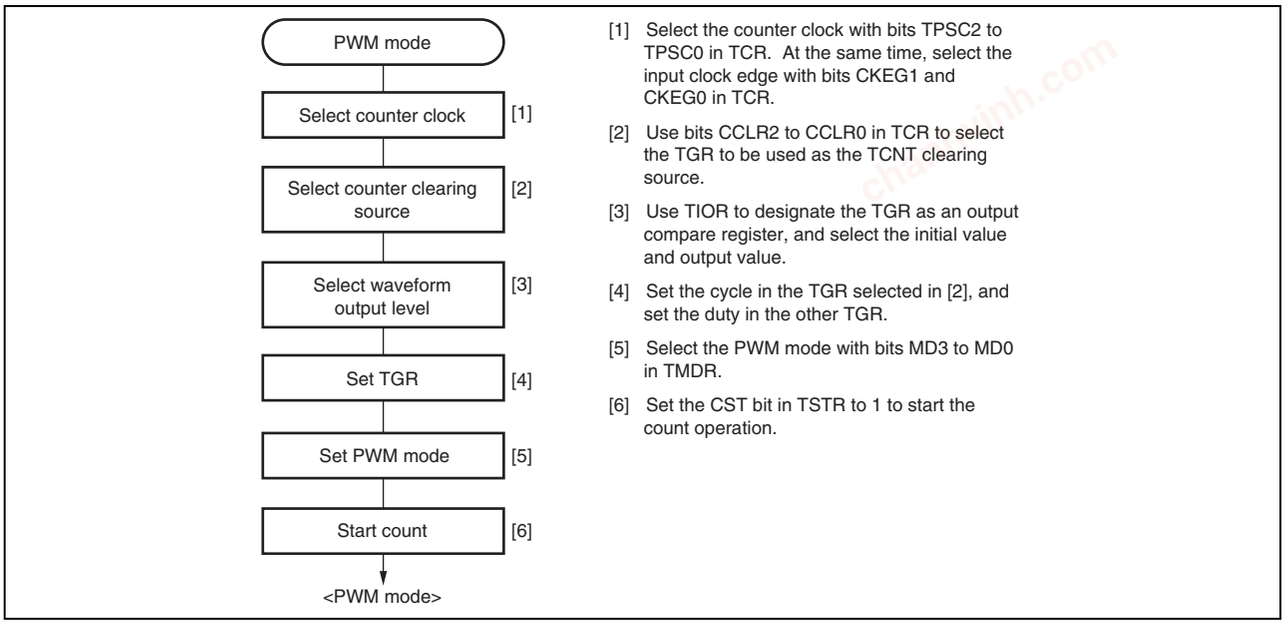


Figure 10.25 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 10.26 shows an example of PWM mode 1 operation.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set for the TGRA initial output value and output value, and 1 is set as the TGRB output value.

In this case, the value set in TGRA is used as the period, and the values set in the TGRB registers are used as the duty levels.

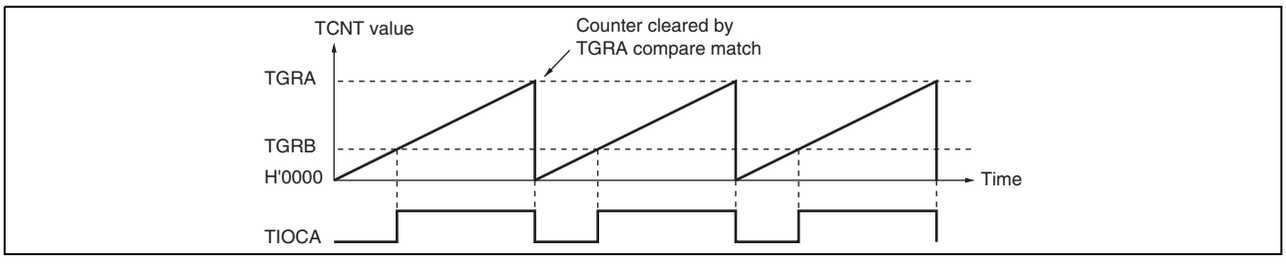


Figure 10.26 Example of PWM Mode Operation (1)

chanhvinh.com

Figure 10.27 shows an example of PWM mode 2 operation.

In this example, synchronous operation is designated for channels 0 and 1, TGRB_1 compare match is set as the TCNT clearing source, and 0 is set for the initial output value and 1 for the output value of the other TGR registers (TGRA_0 to TGRD_0, TGRA_1), outputting a 5-phase PWM waveform.

In this case, the value set in TGRB_1 is used as the cycle, and the values set in the other TGRs are used as the duty levels.

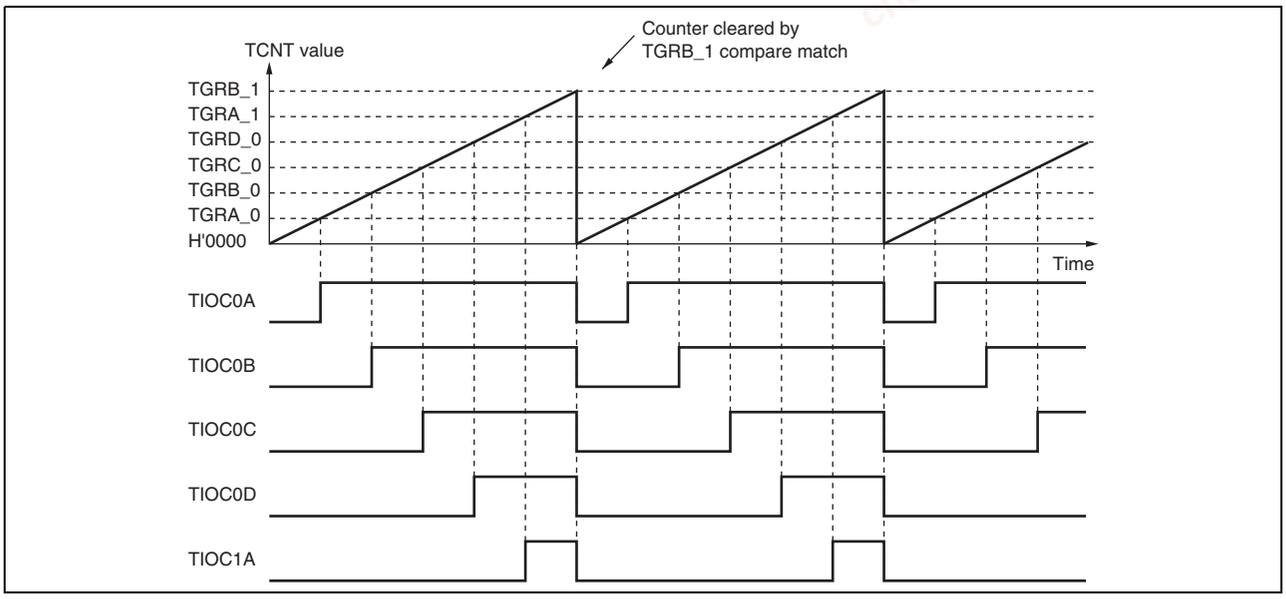


Figure 10.27 Example of PWM Mode Operation (2)

chanhvinh.com

Figure 10.28 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

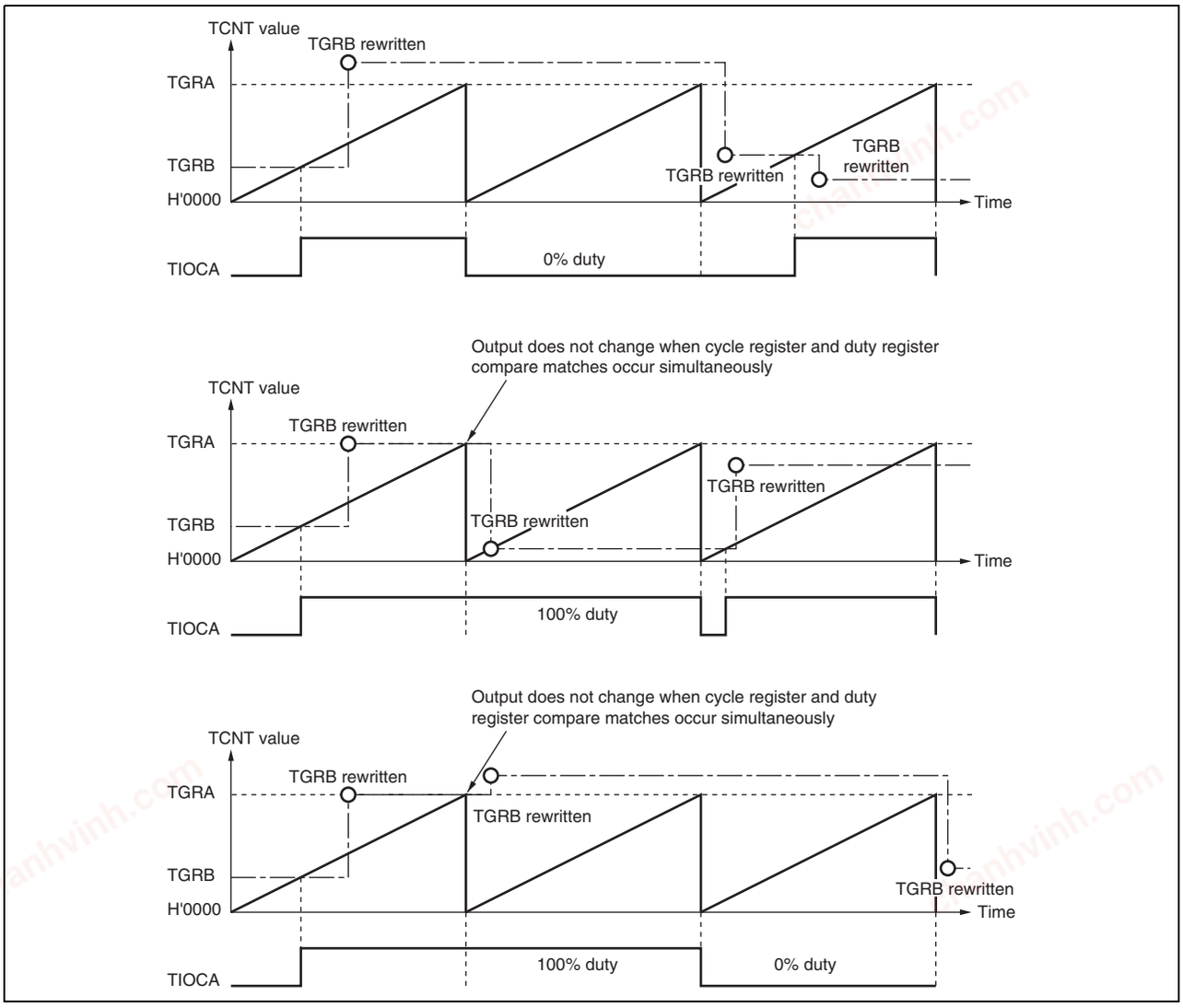


Figure 10.28 Example of PWM Mode Operation (3)

chanhvinh.com

10.4.6 Phase Counting Mode

In phase counting mode, the phase difference between two external clock inputs is detected and TCNT is incremented/decremented accordingly. This mode can be set for channels 1 and 2.

When phase counting mode is set, an external clock is selected as the counter input clock and TCNT operates as an up/down-counter regardless of the setting of bits TPSC0 to TPSC2 and bits CKEG0 and CKEG1 in TCR. However, the functions of bits CCLR0 and CCLR1 in TCR, and of TIOR, TIER, and TGR, are valid, and input capture/compare match and interrupt functions can be used.

This can be used for two-phase encoder pulse input.

If overflow occurs when TCNT is counting up, the TCFV flag in TSR is set; if underflow occurs when TCNT is counting down, the TCFU flag is set.

The TCFD bit in TSR is the count direction flag. Reading the TCFD flag reveals whether TCNT is counting up or down.

Table 10.45 shows the correspondence between external clock pins and channels.

Table 10.45 Phase Counting Mode Clock Input Pins

	External Clock Pins	
Channels	**A-Phase**	**B-Phase**
When channel 1 is set to phase counting mode	TCLKA	TCLKB
When channel 2 is set to phase counting mode	TCLKC	TCLKD

(1) Example of Phase Counting Mode Setting Procedure

Figure 10.29 shows an example of the phase counting mode setting procedure.

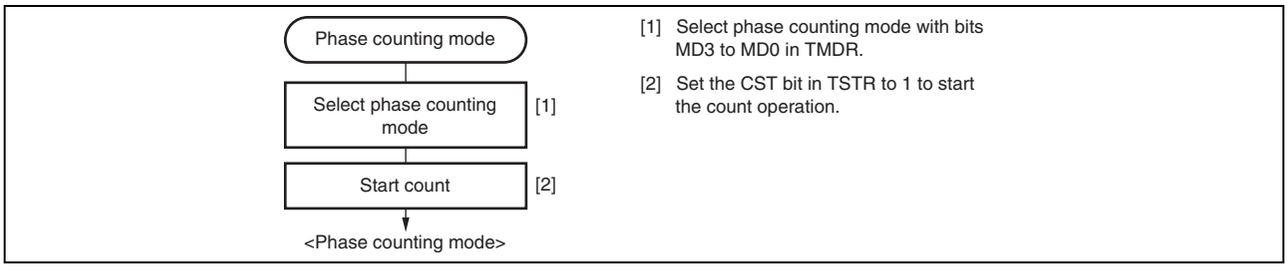


Figure 10.29 Example of Phase Counting Mode Setting Procedure

chanhvinh.com

(2) Examples of Phase Counting Mode Operation

In phase counting mode, TCNT counts up or down according to the phase difference between two external clocks. There are four modes, according to the count conditions.

(a) Phase counting mode 1

Figure 10.30 shows an example of phase counting mode 1 operation, and Table 10.46 summarizes the TCNT up/down-count conditions.

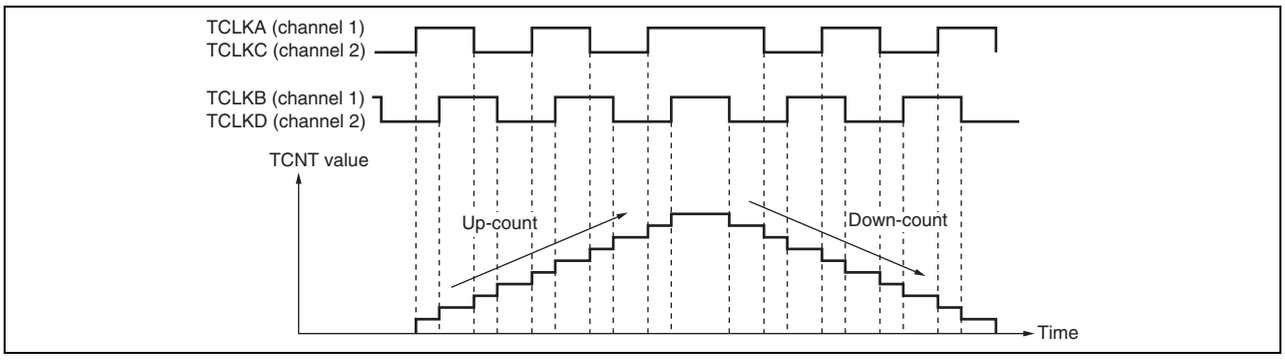


Figure 10.30 Example of Phase Counting Mode 1 Operation

Table 10.46 Up/Down-Count Conditions in Phase Counting Mode 1

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	
↓	Low level	

[Legend]
↑ :Rising edge
↓ :Falling edge

chanhvinh.com

(b) Phase counting mode 2

Figure 10.31 shows an example of phase counting mode 2 operation, and Table 10.47 summarizes the TCNT up/down-count conditions.

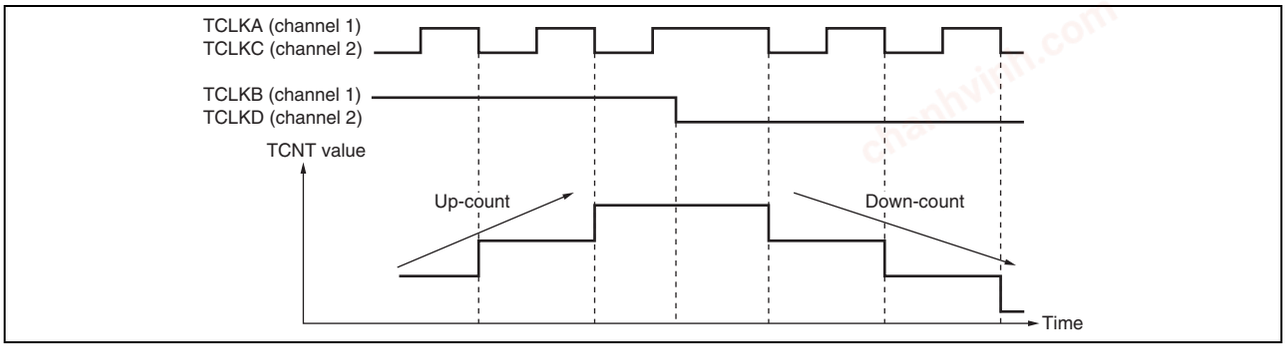


Figure 10.31 Example of Phase Counting Mode 2 Operation

Table 10.47 Up/Down-Count Conditions in Phase Counting Mode 2

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	↑	Don't care
Low level	↓	Don't care
↑	Low level	Don't care
↓	High level	Up-count
High level	↓	Don't care
Low level	↑	Don't care
↑	High level	Don't care
↓	Low level	Down-count

[Legend]
↑ :Rising edge
↓ :Falling edge

chanhvinh.com

(c) Phase counting mode 3

Figure 10.32 shows an example of phase counting mode 3 operation, and Table 10.48 summarizes the TCNT up/down-count conditions.

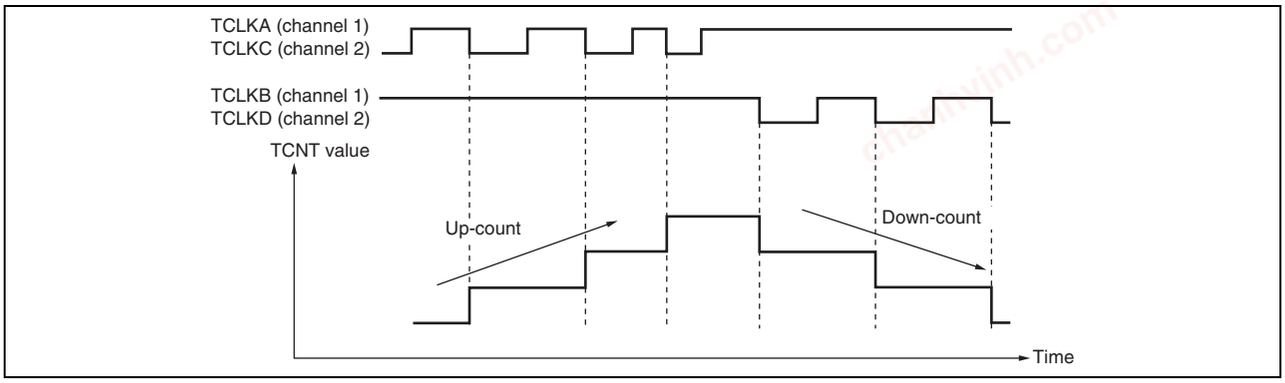


Figure 10.32 Example of Phase Counting Mode 3 Operation

Table 10.48 Up/Down-Count Conditions in Phase Counting Mode 3

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	↑	Don't care
Low level	↓	Don't care
↑	Low level	Don't care
↓	High level	Up-count
High level	↓	Down-count
Low level	↑	Don't care
↑	High level	Don't care
↓	Low level	Don't care

[Legend]
↑ :Rising edge
↓ :Falling edge

chanhvinh.com

(d) Phase counting mode 4

Figure 10.33 shows an example of phase counting mode 4 operation, and Table 10.49 summarizes the TCNT up/down-count conditions.

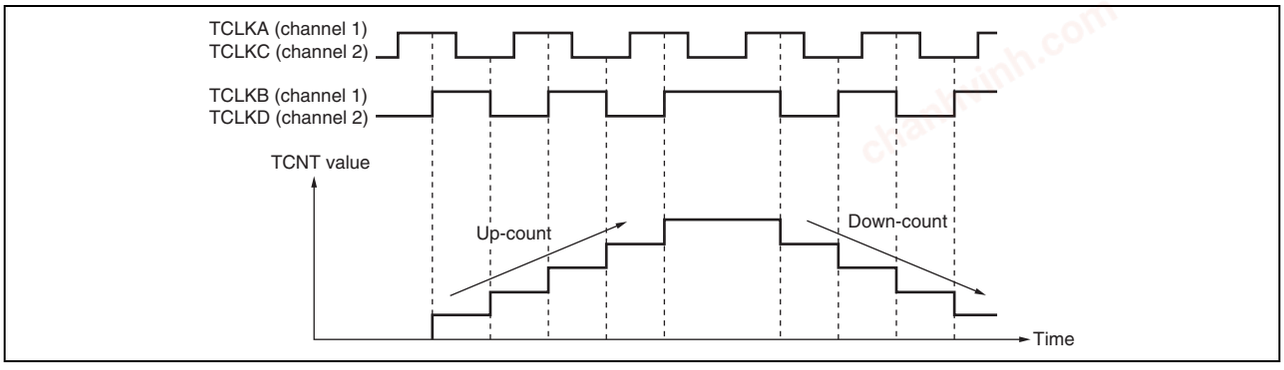


Figure 10.33 Example of Phase Counting Mode 4 Operation

Table 10.49 Up/Down-Count Conditions in Phase Counting Mode 4

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	⌐↑	Up-count
Low level	⌐↓	
⌐↑	Low level	Don't care
⌐↓	High level	
High level	⌐↓	Down-count
Low level	⌐↑	
⌐↑	High level	Don't care
⌐↓	Low level	

[Legend]
⌐↑ :Rising edge
⌐↓ :Falling edge

chanhvinh.com

(3) Phase Counting Mode Application Example

Figure 10.34 shows an example in which channel 1 is in phase counting mode, and channel 1 is coupled with channel 0 to input servo motor 2-phase encoder pulses in order to detect position or speed.

Channel 1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to TCLKA and TCLKB.

Channel 0 operates with TCNT counter clearing by TGRC_0 compare match; TGRA_0 and TGRC_0 are used for the compare match function and are set with the speed control period and position control period. TGRB_0 is used for input capture, with TGRB_0 and TGRD_0 operating in buffer mode. The channel 1 counter input clock is designated as the TGRB_0 input capture source, and the pulse widths of 2-phase encoder 4-multiplication pulses are detected.

TGRA_1 and TGRB_1 for channel 1 are designated for input capture, and channel 0 TGRA_0 and TGRC_0 compare matches are selected as the input capture source and store the up/down-counter values for the control periods.

This procedure enables the accurate detection of position and speed.

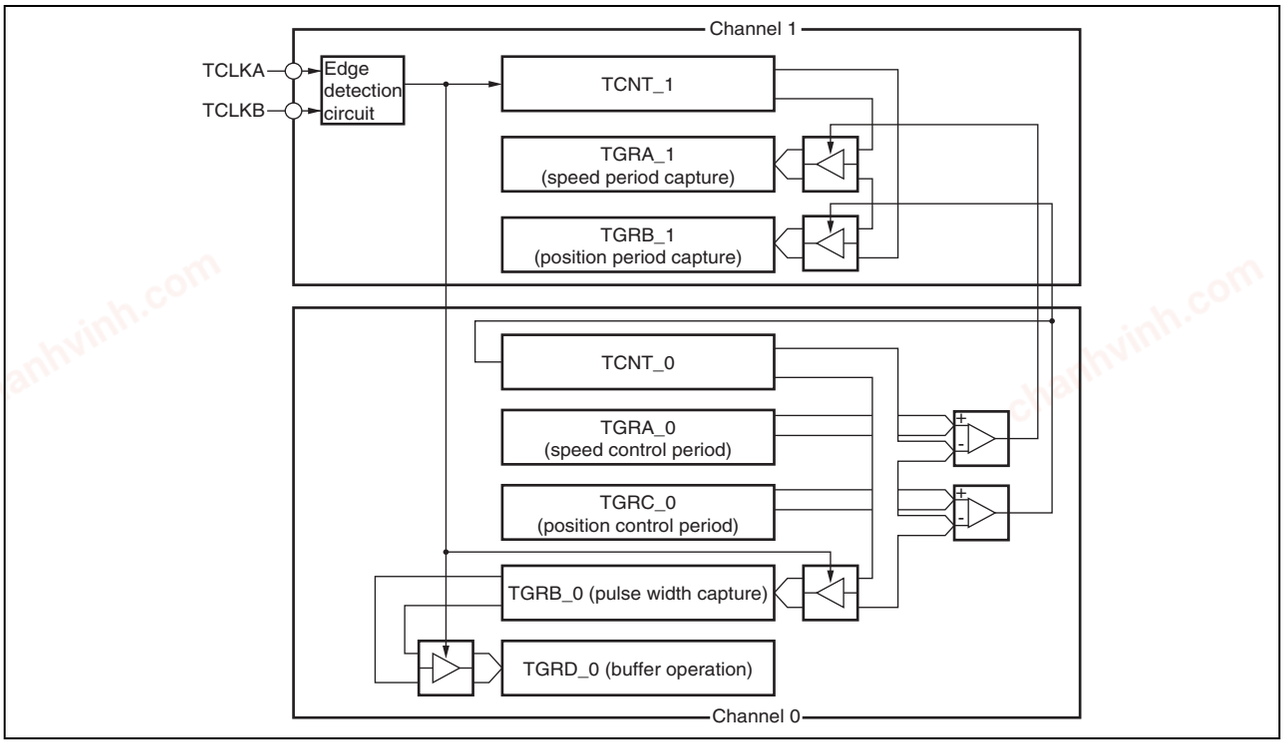


Figure 10.34 Phase Counting Mode Application Example

chanhvinh.com

10.4.7 Reset-Synchronized PWM Mode

In the reset-synchronized PWM mode, three-phase output of positive and negative PWM waveforms that share a common wave transition point can be obtained by combining channels 3 and 4.

When set for reset-synchronized PWM mode, the TIOC3B, TIOC3D, TIOC4A, TIOC4C, TIOC4B, and TIOC4D pins function as PWM output pins and TCNT_3 functions as an upcounter.

Table 10.50 shows the PWM output pins used. Table 10.51 shows the settings of the registers.

Table 10.50 Output Pins for Reset-Synchronized PWM Mode

Channel	Output Pin	Description
3	TIOC3B	PWM output pin 1
	TIOC3D	PWM output pin 1' (negative-phase waveform of PWM output 1)
4	TIOC4A	PWM output pin 2
	TIOC4C	PWM output pin 2' (negative-phase waveform of PWM output 2)
	TIOC4B	PWM output pin 3
	TIOC4D	PWM output pin 3' (negative-phase waveform of PWM output 3)

Table 10.51 Register Settings for Reset-Synchronized PWM Mode

Register	Description of Setting
TCNT_3	Initial setting of H'0000
TCNT_4	Initial setting of H'0000
TGRA_3	Set count cycle for TCNT_3
TGRB_3	Sets the turning point for PWM waveform output by the TIOC3B and TIOC3D pins
TGRA_4	Sets the turning point for PWM waveform output by the TIOC4A and TIOC4C pins
TGRB_4	Sets the turning point for PWM waveform output by the TIOC4B and TIOC4D pins

chanhvinh.com

(1) Procedure for Selecting the Reset-Synchronized PWM Mode

Figure 10.35 shows an example of procedure for selecting the reset synchronized PWM mode.

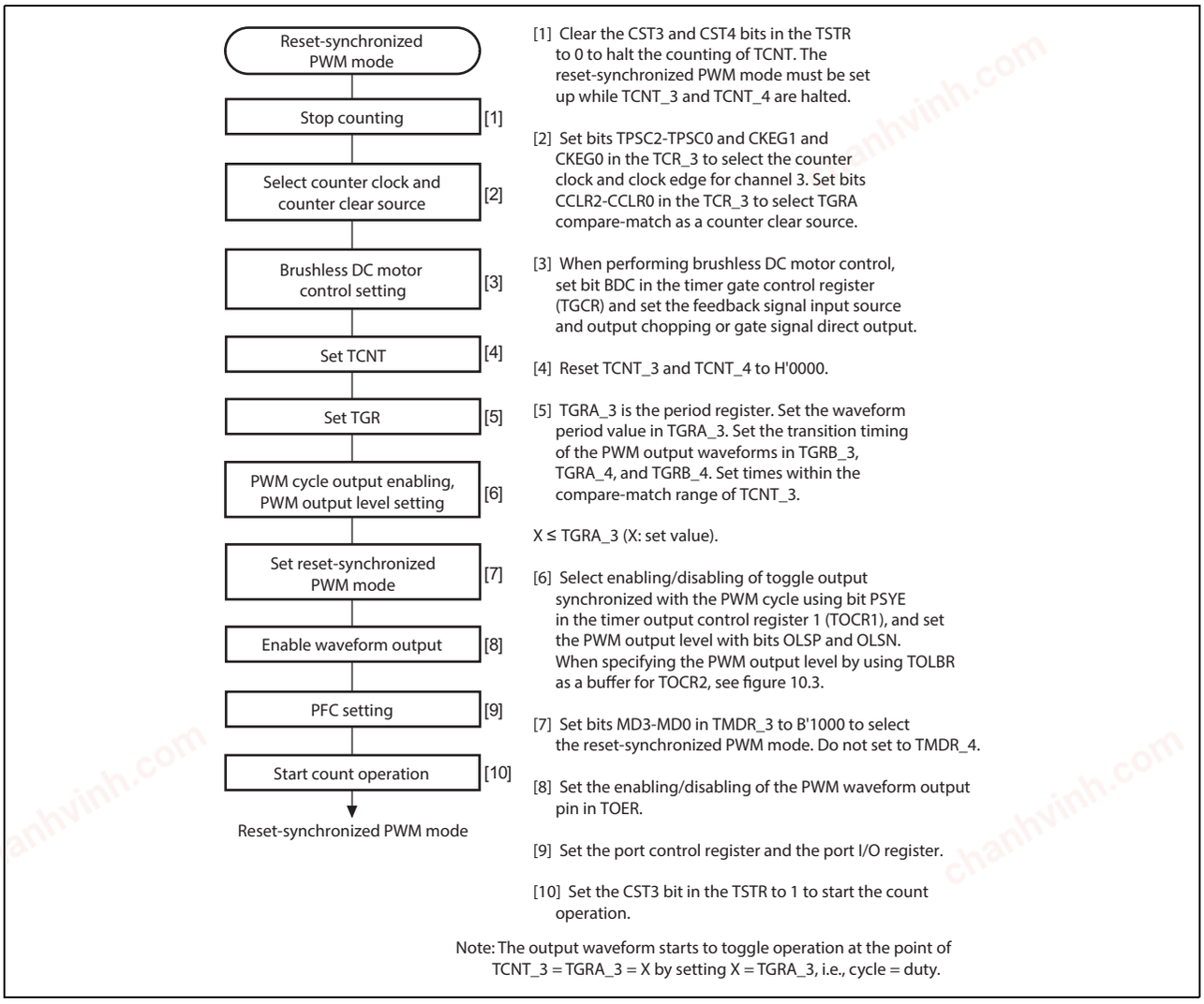


Figure 10.35 Procedure for Selecting Reset-Synchronized PWM Mode

chanhvinh.com

(2) Reset-Synchronized PWM Mode Operation

Figure 10.36 shows an example of operation in the reset-synchronized PWM mode. TCNT_3 and TCNT_4 operate as upcounters. The counter is cleared when a TCNT_3 and TGRA_3 compare-match occurs, and then begins incrementing from H'0000. The PWM output pin output toggles with each occurrence of a TGRB_3, TGRA_4, TGRB_4 compare-match, and upon counter clears.

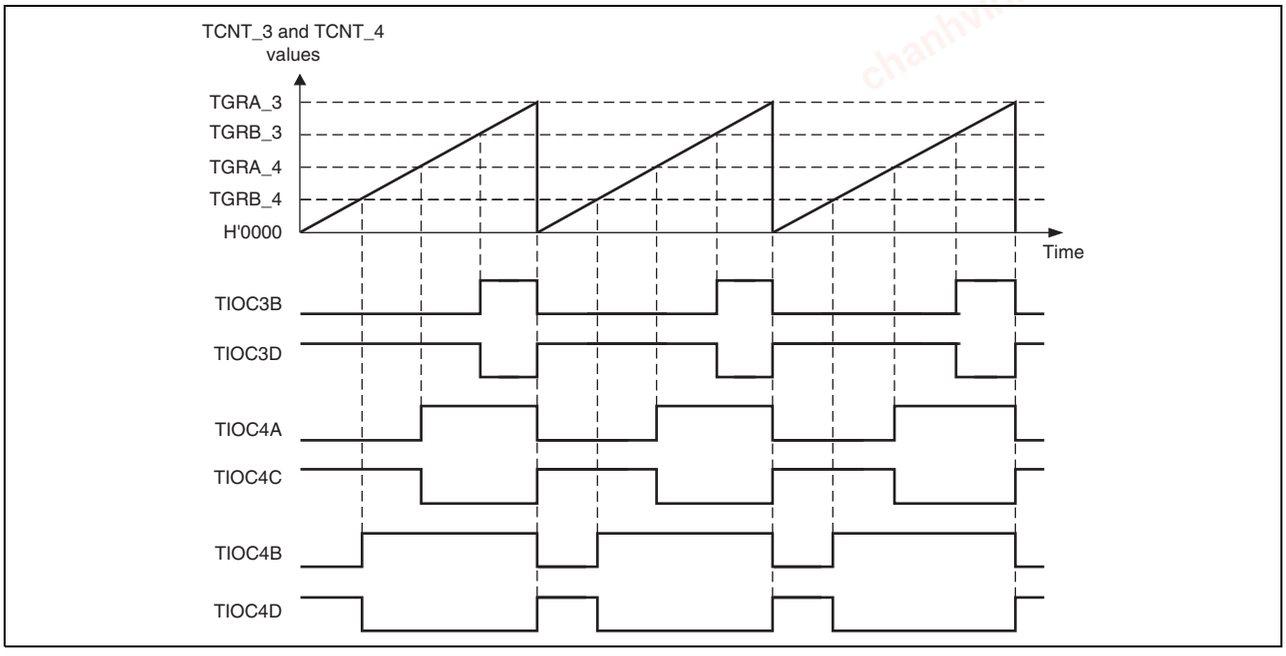


Figure 10.36 Reset-Synchronized PWM Mode Operation Example (When TOCR's OLSN = 1 and OLSP = 1)

chanhvinh.com

10.4.8 Complementary PWM Mode

In the complementary PWM mode, three-phase output of non-overlapping positive and negative PWM waveforms can be obtained by combining channels 3 and 4. PWM waveforms without non-overlapping interval are also available.

In complementary PWM mode, TIOC3B, TIOC3D, TIOC4A, TIOC4B, TIOC4C, and TIOC4D pins function as PWM output pins, the TIOC3A pin can be set for toggle output synchronized with the PWM period. TCNT_3 and TCNT_4 function as up/down counters.

Table 10.52 shows the PWM output pins used. Table 10.53 shows the settings of the registers used.

Table 10.52 Output Pins for Complementary PWM Mode

Channel	Output Pin	Description
3	TIOC3A	Toggle output synchronized with PWM period (or I/O port)
	TIOC3B	PWM output pin 1
	TIOC3C	I/O port*
	TIOC3D	PWM output pin 1' (non-overlapping negative-phase waveform of PWM output 1; PWM output without non-overlapping interval is also available)
4	TIOC4A	PWM output pin 2
	TIOC4B	PWM output pin 3
	TIOC4C	PWM output pin 2' (non-overlapping negative-phase waveform of PWM output 2; PWM output without non-overlapping interval is also available)
	TIOC4D	PWM output pin 3' (non-overlapping negative-phase waveform of PWM output 3; PWM output without non-overlapping interval is also available)

Note: * Avoid setting the TIOC3C pin as a timer I/O pin in the complementary PWM mode.

chanhvinh.com

Table 10.53 Register Settings for Complementary PWM Mode

Channel	Counter/Register	Description	Read/Write from CPU
3	TCNT_3	Start of up-count from value set in dead time register	Maskable by TRWER setting*
	TGRA_3	Set TCNT_3 upper limit value (1/2 carrier cycle + dead time)	Maskable by TRWER setting*
	TGRB_3	PWM output 1 compare register	Maskable by TRWER setting*
	TGRC_3	TGRA_3 buffer register	Always readable/writable
	TGRD_3	PWM output 1/TGRB_3 buffer register	Always readable/writable
4	TCNT_4	Up-count start, initialized to H'0000	Maskable by TRWER setting*
	TGRA_4	PWM output 2 compare register	Maskable by TRWER setting*
	TGRB_4	PWM output 3 compare register	Maskable by TRWER setting*
	TGRC_4	PWM output 2/TGRA_4 buffer register	Always readable/writable
	TGRD_4	PWM output 3/TGRB_4 buffer register	Always readable/writable
Timer dead time data register (TDDR)		Set TCNT_4 and TCNT_3 offset value (dead time value)	Maskable by TRWER setting*
Timer cycle data register (TCDR)		Set TCNT_4 upper limit value (1/2 carrier cycle)	Maskable by TRWER setting*
Timer cycle buffer register (TCBR)		TCDR buffer register	Always readable/writable
Subcounter (TCNTS)		Subcounter for dead time generation	Read-only
Temporary register 1 (TEMP1)		PWM output 1/TGRB_3 temporary register	Not readable/writable
Temporary register 2 (TEMP2)		PWM output 2/TGRA_4 temporary register	Not readable/writable
Temporary register 3 (TEMP3)		PWM output 3/TGRB_4 temporary register	Not readable/writable

Note: * Access can be enabled or disabled according to the setting of bit 0 (RWE) in TRWER (timer read/write enable register).

chanhvinh.com

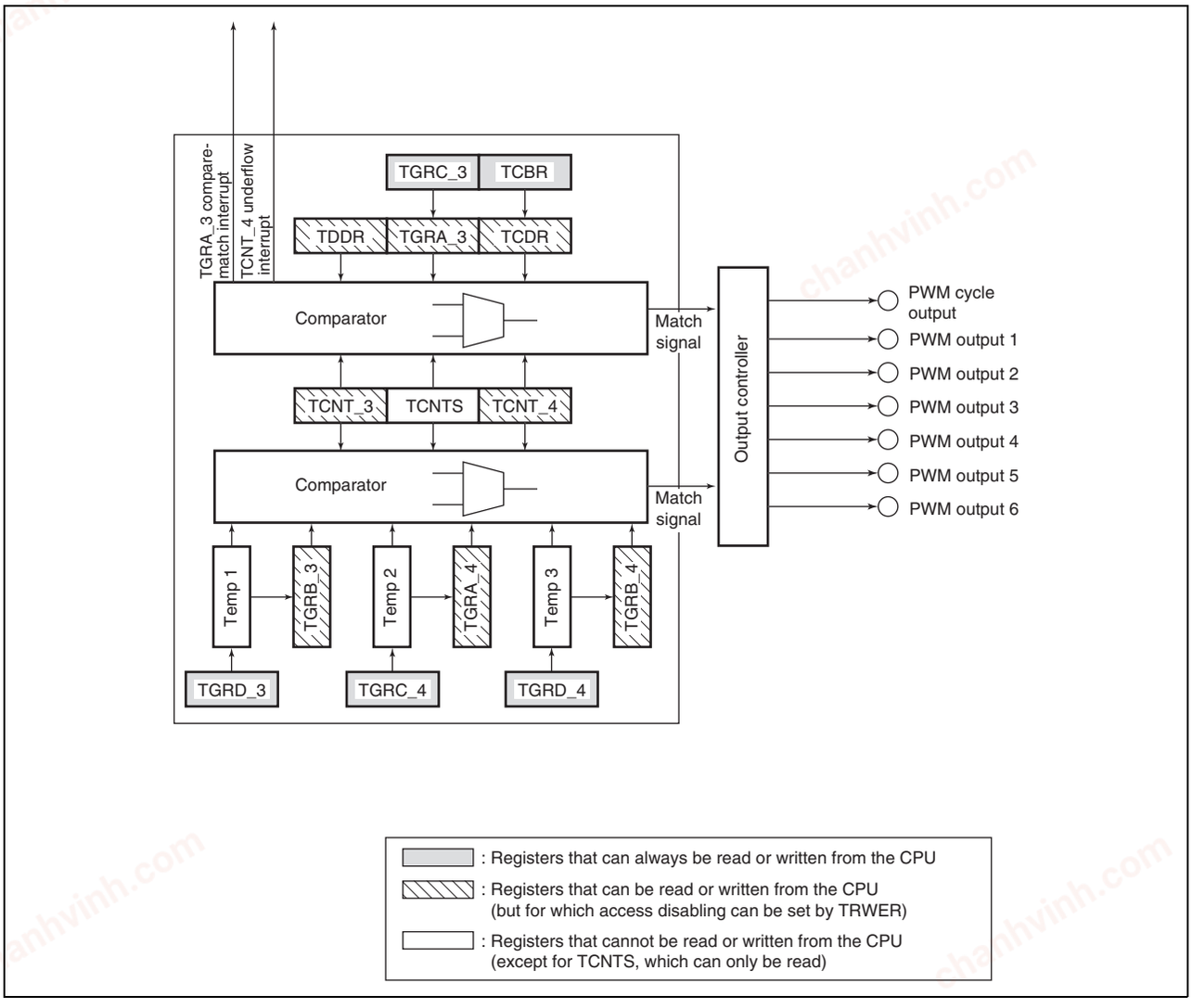


Figure 10.37 Block Diagram of Channels 3 and 4 in Complementary PWM Mode

chanhvinh.com

(1) Example of Complementary PWM Mode Setting Procedure

An example of the complementary PWM mode setting procedure is shown in Figure 10.38.

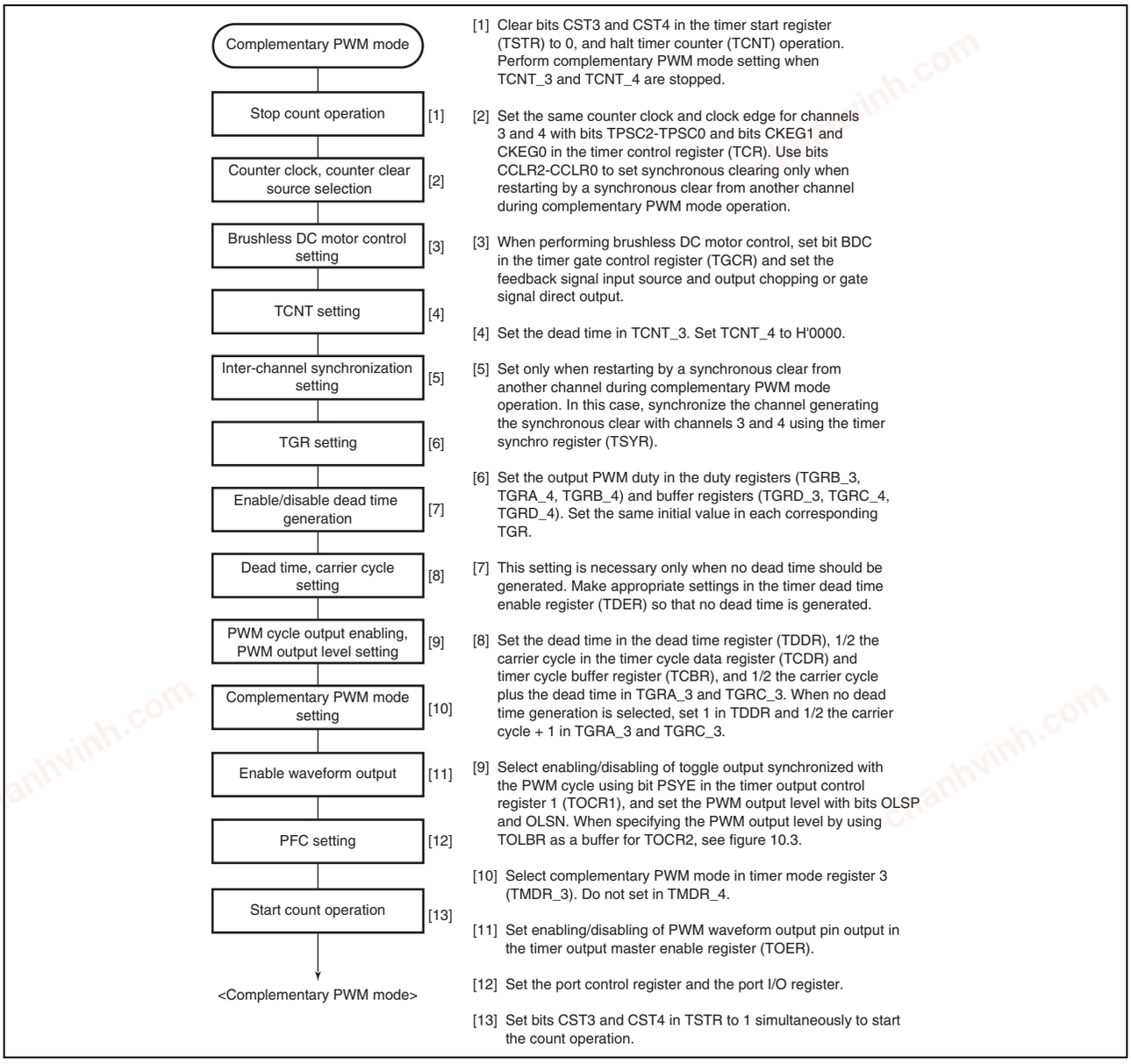


Figure 10.38 Example of Complementary PWM Mode Setting Procedure

chanhvinh.com

(2) Outline of Complementary PWM Mode Operation

In complementary PWM mode, 6-phase PWM output is possible. Figure 10.39 illustrates counter operation in complementary PWM mode, and Figure 10.40 shows an example of complementary PWM mode operation.

(a) Counter Operation

In complementary PWM mode, three counters—TCNT_3, TCNT_4, and TCNTS—perform up/down-count operations.

TCNT_3 is automatically initialized to the value set in TDDR when complementary PWM mode is selected and the CST bit in TSTR is 0.

When the CST bit is set to 1, TCNT_3 counts up to the value set in TGRA_3, then switches to down-counting when it matches TGRA_3. When the TCNT_3 value matches TDDR, the counter switches to up-counting, and the operation is repeated in this way.

TCNT_4 is initialized to H'0000.

When the CST bit is set to 1, TCNT_4 counts up in synchronization with TCNT_3, and switches to down-counting when it matches TCDR. On reaching H'0000, TCNT_4 switches to up-counting, and the operation is repeated in this way.

TCNTS is a read-only counter. It need not be initialized.

When TCNT_3 matches TCDR during TCNT_3 and TCNT_4 up/down-counting, down-counting is started, and when TCNTS matches TCDR, the operation switches to up-counting. When TCNTS matches TGRA_3, it is cleared to H'0000. When TCNT_4 matches TDDR during TCNT_3 and TCNT_4 down-counting, up-counting is started, and when TCNTS matches TDDR, the operation switches to down-counting. When TCNTS reaches H'0000, it is set with the value in TGRA_3.

TCNTS is compared with the compare register and temporary register in which the PWM duty is set during the count operation only.

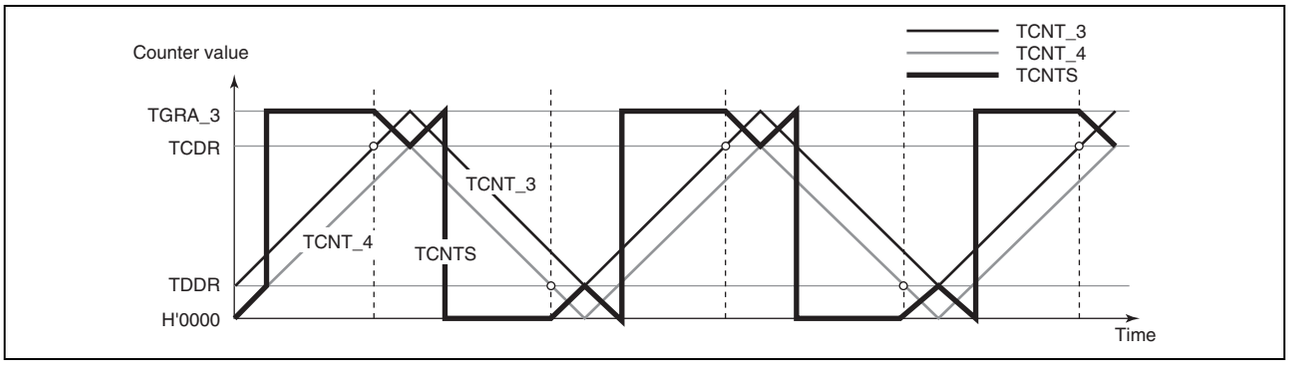


Figure 10.39 Complementary PWM Mode Counter Operation

chanhvinh.com

(b) Register Operation

In complementary PWM mode, nine registers are used, comprising compare registers, buffer registers, and temporary registers. Figure 10.40 shows an example of complementary PWM mode operation.

The registers which are constantly compared with the counters to perform PWM output are TGRB_3, TGRA_4, and TGRB_4. When these registers match the counter, the value set in bits OLSN and OLSP in the timer output control register (TOCR) is output.

The buffer registers for these compare registers are TGRD_3, TGRC_4, and TGRD_4.

Between a buffer register and compare register there is a temporary register. The temporary registers cannot be accessed by the CPU.

Data in a compare register is changed by writing the new data to the corresponding buffer register. The buffer registers can be read or written at any time.

The data written to a buffer register is constantly transferred to the temporary register in the Ta interval. Data is not transferred to the temporary register in the Tb interval. Data written to a buffer register in this interval is transferred to the temporary register at the end of the Tb interval.

The value transferred to a temporary register is transferred to the compare register when TCNTS for which the Tb interval ends matches TGRA_3 when counting up, or H'0000 when counting down. The timing for transfer from the temporary register to the compare register can be selected with bits MD3 to MD0 in the timer mode register (TMDR). Figure 10.40 shows an example in which the mode is selected in which the change is made in the trough.

In the Tb interval (Tb1 in Figure 10.40) in which data transfer to the temporary register is not performed, the temporary register has the same function as the compare register, and is compared with the counter. In this interval, therefore, there are two compare match registers for one-phase output, with the compare register containing the pre-change data, and the temporary register containing the new data. In this interval, the three counters—TCNT_3, TCNT_4, and TCNTS—and two registers—compare register and temporary register—are compared, and PWM output controlled accordingly.

chanhvinh.com

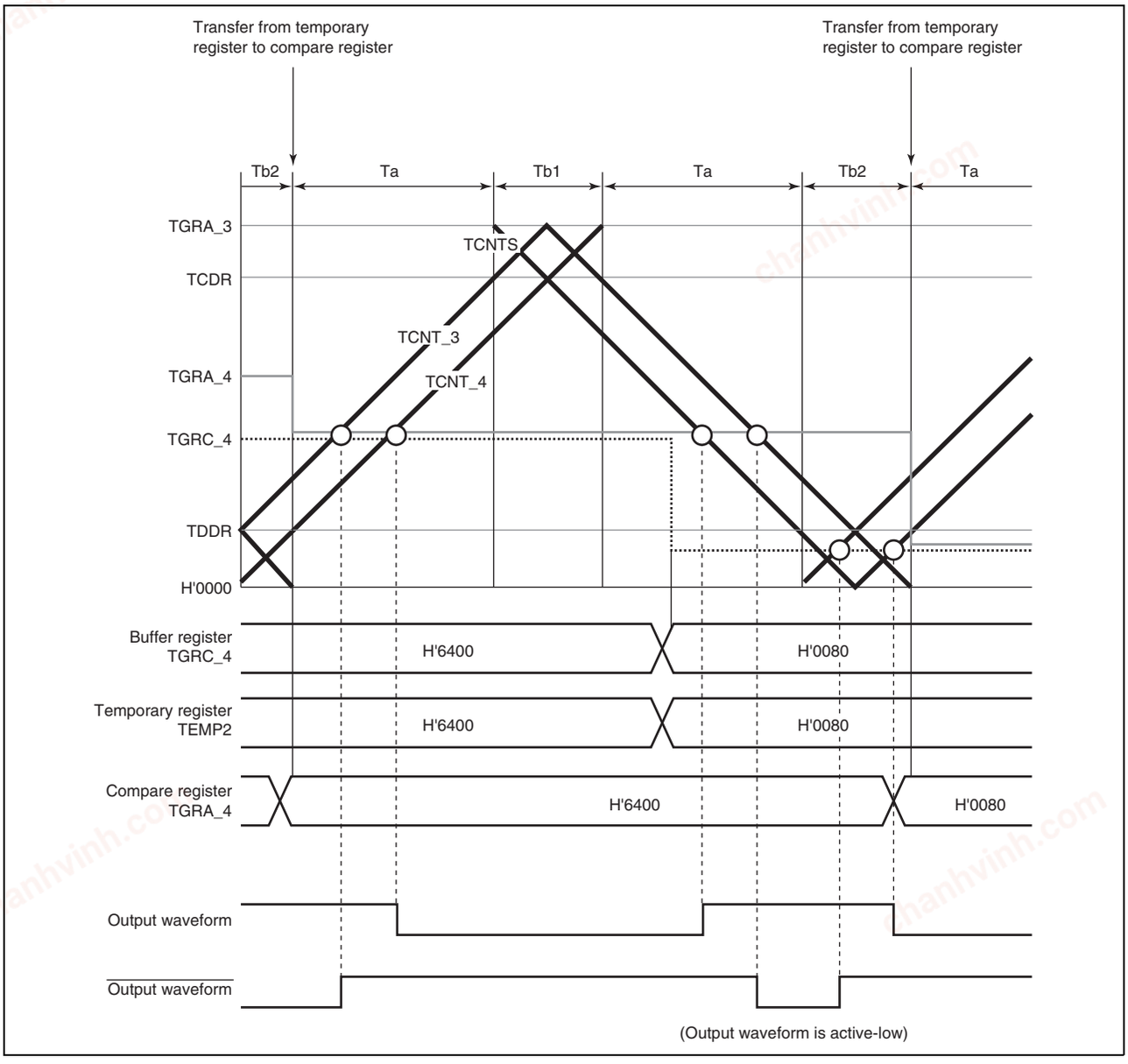


Figure 10.40 Example of Complementary PWM Mode Operation

chanhvinh.com

(c) Initialization

In complementary PWM mode, there are six registers that must be initialized. In addition, there is a register that specifies whether to generate dead time (it should be used only when dead time generation should be disabled).

Before setting complementary PWM mode with bits MD3 to MD0 in the timer mode register (TMDR), the following initial register values must be set.

TGRC_3 operates as the buffer register for TGRA_3, and should be set with 1/2 the PWM carrier cycle + dead time Td. The timer cycle buffer register (TCBR) operates as the buffer register for the timer cycle data register (TCDR), and should be set with 1/2 the PWM carrier cycle. Set dead time Td in the timer dead time data register (TDDR).

When dead time is not needed, the TDER bit in the timer dead time enable register (TDER) should be cleared to 0, TGRC_3 and TGRA_3 should be set to 1/2 the PWM carrier cycle + 1, and TDDR should be set to 1.

Set the respective initial PWM duty values in buffer registers TGRD_3, TGRC_4, and TGRD_4.

The values set in the five buffer registers excluding TDDR are transferred simultaneously to the corresponding compare registers when complementary PWM mode is set.

Set TCNT_4 to H'0000 before setting complementary PWM mode.

Table 10.54 Registers and Counters Requiring Initialization

Register/Counter	Set Value
TGRC_3	1/2 PWM carrier cycle + dead time Td (1/2 PWM carrier cycle + 1 when dead time generation is disabled by TDER)
TDDR	Dead time Td (1 when dead time generation is disabled by TDER)
TCBR	1/2 PWM carrier cycle
TGRD_3, TGRC_4, TGRD_4	Initial PWM duty value for each phase
TCNT_4	H'0000

Note: The TGRC_3 set value must be the sum of 1/2 the PWM carrier cycle set in TCBR and dead time Td set in TDDR. When dead time generation is disabled by TDER, TGRC_3 must be set to 1/2 the PWM carrier cycle + 1.

chanhvinh.com

(d) PWM Output Level Setting

In complementary PWM mode, the PWM pulse output level is set with bits OLSN and OLSP in timer output control register 1 (TOCR1) or bits OLS1P to OLS3P and OLS1N to OLS3N in timer output control register 2 (TOCR2).

The output level can be set for each of the three positive phases and three negative phases of 6-phase output.

Complementary PWM mode should be cleared before setting or changing output levels.

(e) Dead Time Setting

In complementary PWM mode, PWM pulses are output with a non-overlapping relationship between the positive and negative phases. This non-overlap time is called the dead time.

The non-overlap time is set in the timer dead time data register (TDDR). The value set in TDDR is used as the TCNT_3 counter start value, and creates non-overlap between TCNT_3 and TCNT_4. Complementary PWM mode should be cleared before changing the contents of TDDR.

(f) Dead Time Suppressing

Dead time generation is suppressed by clearing the TDER bit in the timer dead time enable register (TDER) to 0. TDER can be cleared to 0 only when 0 is written to it after reading TDER = 1.

TGRA_3 and TGRC_3 should be set to 1/2 PWM carrier cycle + 1 and the timer dead time data register (TDDR) should be set to 1.

By the above settings, PWM waveforms without dead time can be obtained. Figure 10.41 shows an example of operation without dead time.

chanhvinh.com

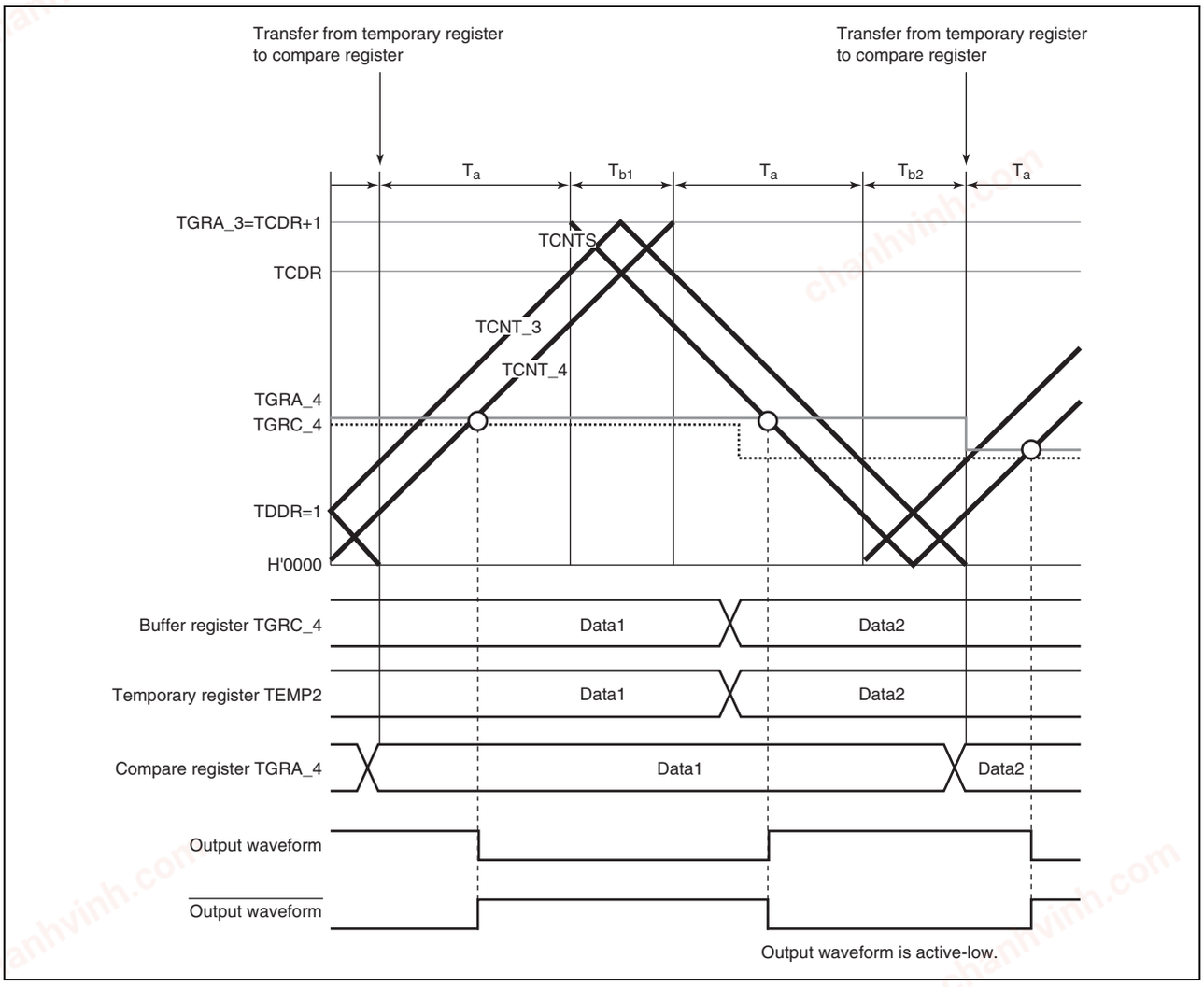


Figure 10.41 Example of Operation without Dead Time

chanhvinh.com

(g) PWM Cycle Setting

In complementary PWM mode, the PWM pulse cycle is set in two registers—TGRA_3, in which the TCNT_3 upper limit value is set, and TCDR, in which the TCNT_4 upper limit value is set. The settings should be made so as to achieve the following relationship between these two registers:

With dead time: TGRA_3 setting = TCDR setting + TDDR setting
TCDR setting > (twice the TDDR setting) + 2
Without dead time: TGRA_3 setting = TCDR setting + 1
TCDR setting > 4

The TGRA_3 and TCDR settings are made by setting the values in buffer registers TGRC_3 and TCBR. The values set in TGRC_3 and TCBR are transferred simultaneously to TGRA_3 and TCDR in accordance with the transfer timing selected with bits MD3 to MD0 in the timer mode register (TMDR).

The updated PWM cycle is reflected from the next cycle when the data update is performed at the crest, and from the current cycle when performed in the trough. Figure 10.42 illustrates the operation when the PWM cycle is updated at the crest.

See (h) Register Data Updating, for the method of updating the data in each buffer register.

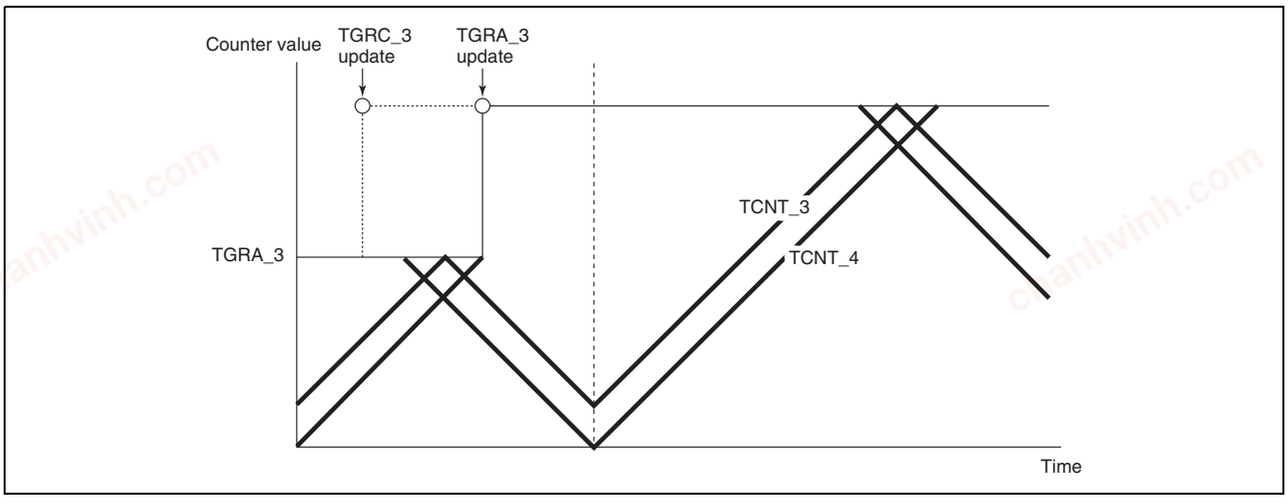


Figure 10.42 Example of PWM Cycle Updating

chanhvinh.com

(h) Register Data Updating

In complementary PWM mode, the buffer register is used to update the data in a compare register. The update data can be written to the buffer register at any time. There are five PWM duty and carrier cycle registers that have buffer registers and can be updated during operation.

There is a temporary register between each of these registers and its buffer register. When subcounter TCNTS is not counting, if buffer register data is updated, the temporary register value is also rewritten. Transfer is not performed from buffer registers to temporary registers when TCNTS is counting; in this case, the value written to a buffer register is transferred after TCNTS halts.

The temporary register value is transferred to the compare register at the data update timing set with bits MD3 to MD0 in the timer mode register (TMDR). Figure 10.43 shows an example of data updating in complementary PWM mode. This example shows the mode in which data updating is performed at both the counter crest and trough.

When rewriting buffer register data, a write to TGRD_4 must be performed at the end of the update. Data transfer from the buffer registers to the temporary registers is performed simultaneously for all five registers after the write to TGRD_4.

A write to TGRD_4 must be performed after writing data to the registers to be updated, even when not updating all five registers, or when updating the TGRD_4 data. In this case, the data written to TGRD_4 should be the same as the data prior to the write operation.

chanhvinh.com

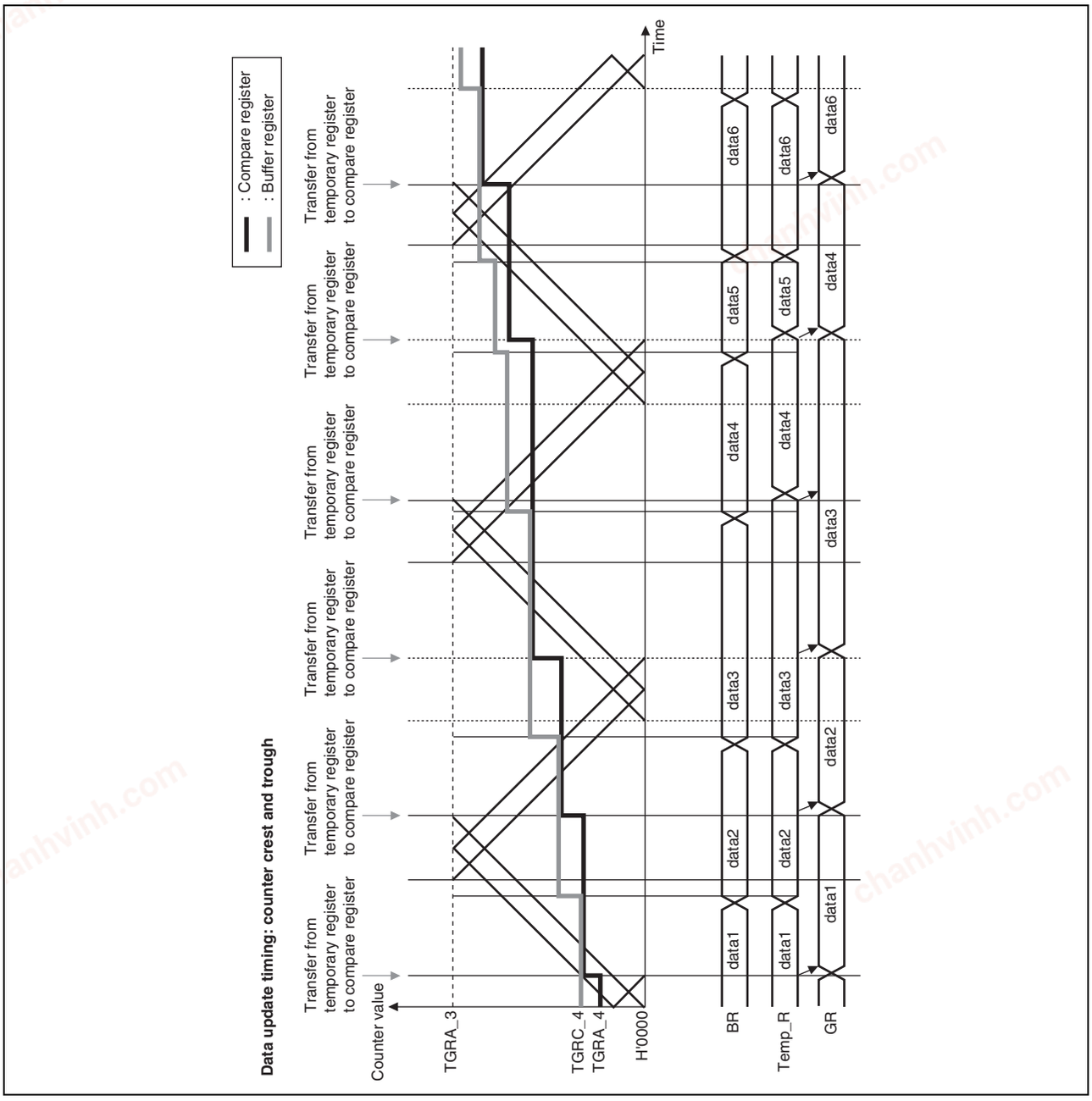


Figure 10.43 Example of Data Update in Complementary PWM Mode

chanhvinh.com

(i) Initial Output in Complementary PWM Mode

In complementary PWM mode, the initial output is determined by the setting of bits OLSN and OLSP in timer output control register 1 (TOCR1) or bits OLS1N to OLS3N and OLS1P to OLS3P in timer output control register 2 (TOCR2).

This initial output is the PWM pulse non-active level, and is output from when complementary PWM mode is set with the timer mode register (TMDR) until TCNT_4 exceeds the value set in the dead time register (TDDR). Figure 10.44 shows an example of the initial output in complementary PWM mode.

An example of the waveform when the initial PWM duty value is smaller than the TDDR value is shown in Figure 10.45.

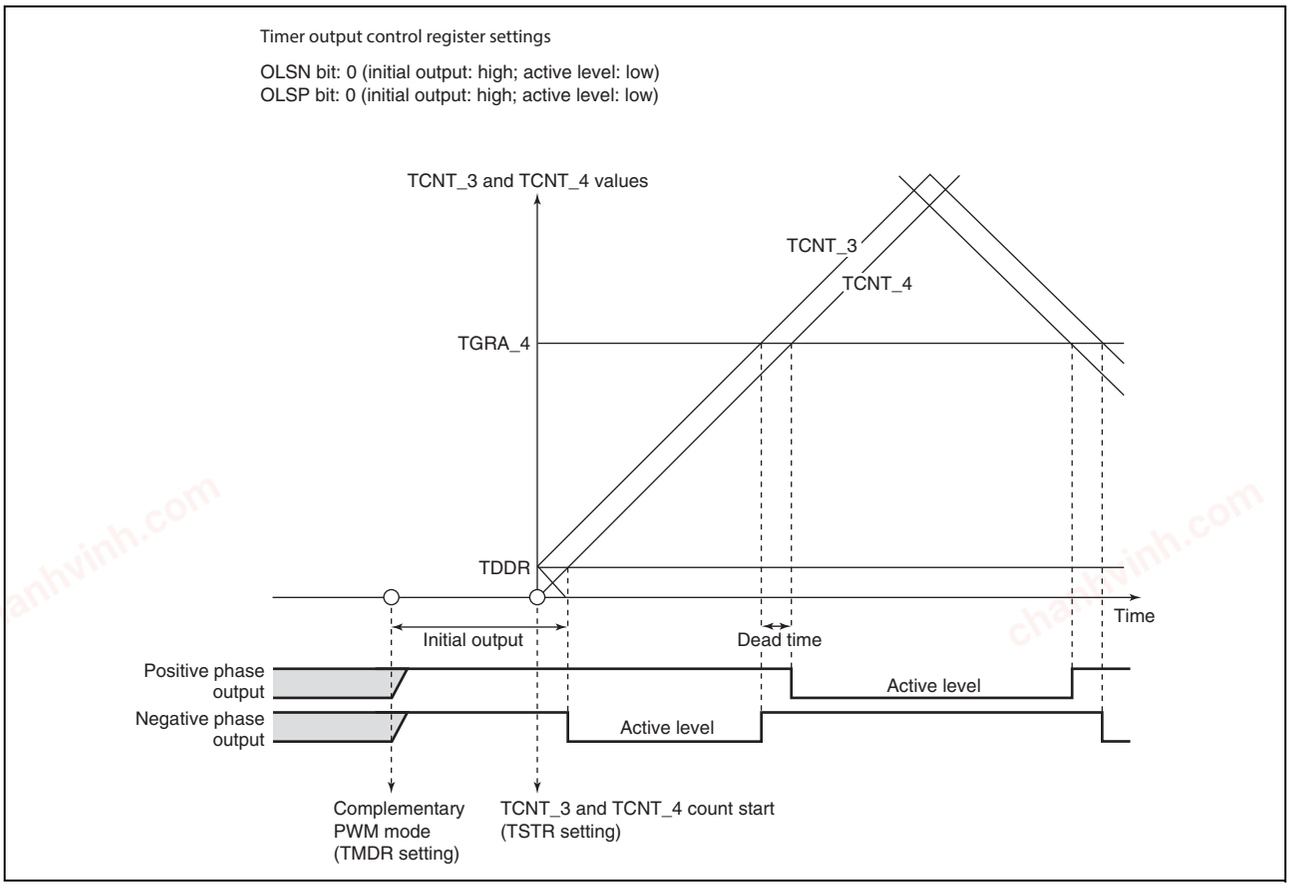


Figure 10.44 Example of Initial Output in Complementary PWM Mode (1)

chanhvinh.com

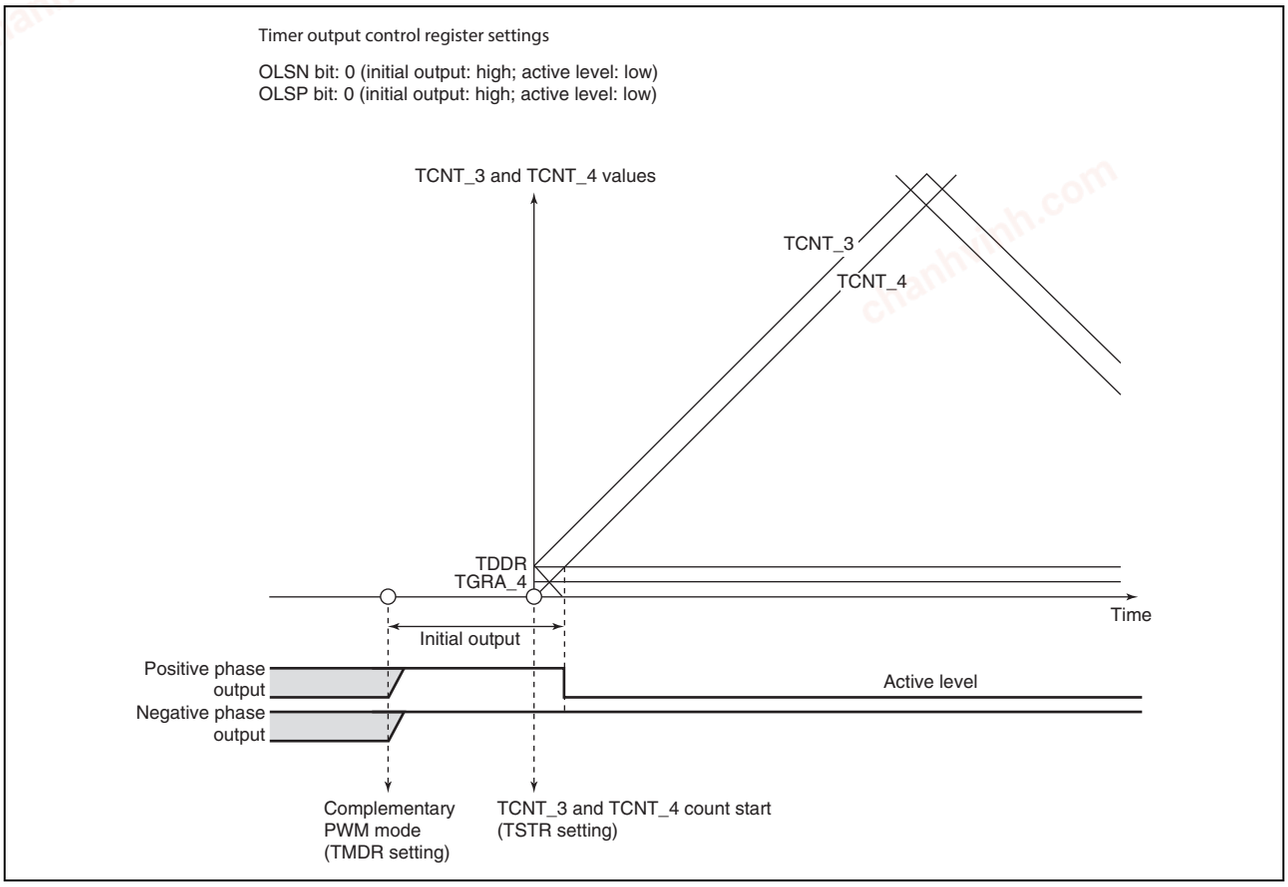


Figure 10.45 Example of Initial Output in Complementary PWM Mode (2)

chanhvinh.com

(j) Complementary PWM Mode PWM Output Generation Method

In complementary PWM mode, 3-phase output is performed of PWM waveforms with a non-overlap time between the positive and negative phases. This non-overlap time is called the dead time.

A PWM waveform is generated by output of the output level selected in the timer output control register in the event of a compare-match between a counter and compare register. While TCNTS is counting, compare register and temporary register values are simultaneously compared to create consecutive PWM pulses from 0 to 100%. The relative timing of on and off compare-match occurrence may vary, but the compare-match that turns off each phase takes precedence to secure the dead time and ensure that the positive phase and negative phase on times do not overlap. Figure 10.46 to Figure 10.48 show examples of waveform generation in complementary PWM mode.

The positive phase/negative phase off timing is generated by a compare-match with the solid-line counter, and the on timing by a compare-match with the dotted-line counter operating with a delay of the dead time behind the solid-line counter. In the T1 period, compare-match **a** that turns off the negative phase has the highest priority, and compare-matches occurring prior to **a** are ignored. In the T2 period, compare-match **c** that turns off the positive phase has the highest priority, and compare-matches occurring prior to **c** are ignored.

In normal cases, compare-matches occur in the order **a** → **b** → **c** → **d** (or **c** → **d** → **a'** → **b'**), as shown in figure 10.46.

If compare-matches deviate from the **a** → **b** → **c** → **d** order, since the time for which the negative phase is off is less than twice the dead time, the figure shows the positive phase is not being turned on. If compare-matches deviate from the **c** → **d** → **a'** → **b'** order, since the time for which the positive phase is off is less than twice the dead time, the figure shows the negative phase is not being turned on.

If compare-match **c** occurs first following compare-match **a**, as shown in Figure 10.47, compare-match **b** is ignored, and the negative phase is turned on by compare-match **d**. This is because turning off of the positive phase has priority due to the occurrence of compare-match **c** (positive phase off timing) before compare-match **b** (positive phase on timing) (consequently, the waveform does not change since the positive phase goes from off to off).

Similarly, in the example in Figure 10.48, compare-match **a'** with the new data in the temporary register occurs before compare-match **c**, but other compare-matches occurring up to **c**, which turns off the positive phase, are ignored. As a result, the negative phase is not turned on.

Thus, in complementary PWM mode, compare-matches at turn-off timings take precedence, and turn-on timing compare-matches that occur before a turn-off timing compare-match are ignored.

chanhvinh.com

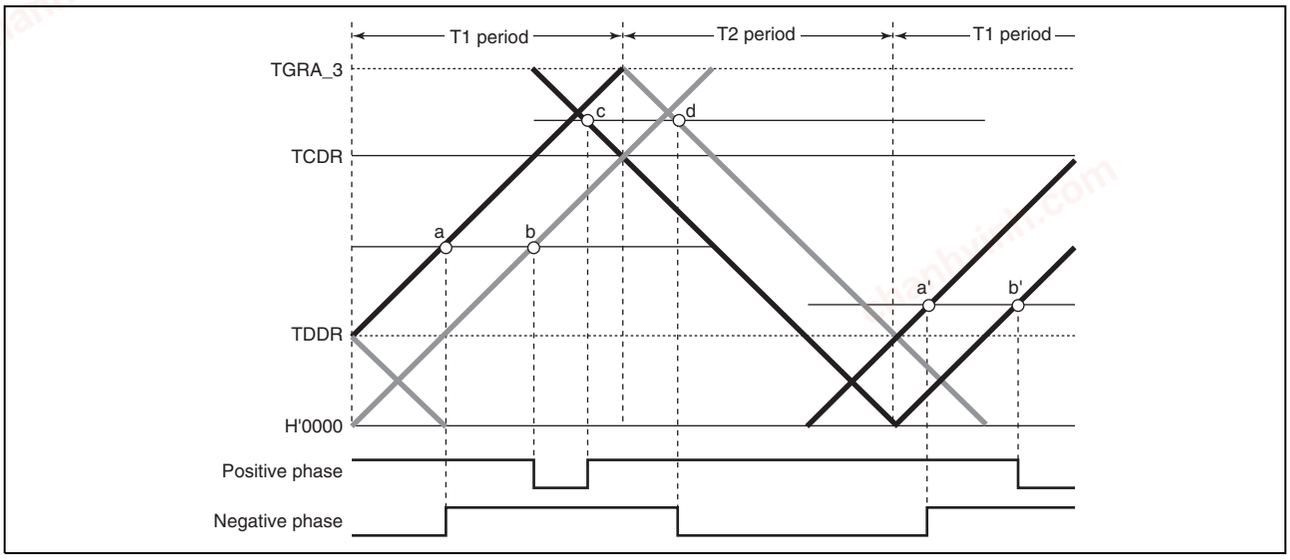


Figure 10.46 Example of Complementary PWM Mode Waveform Output (1)

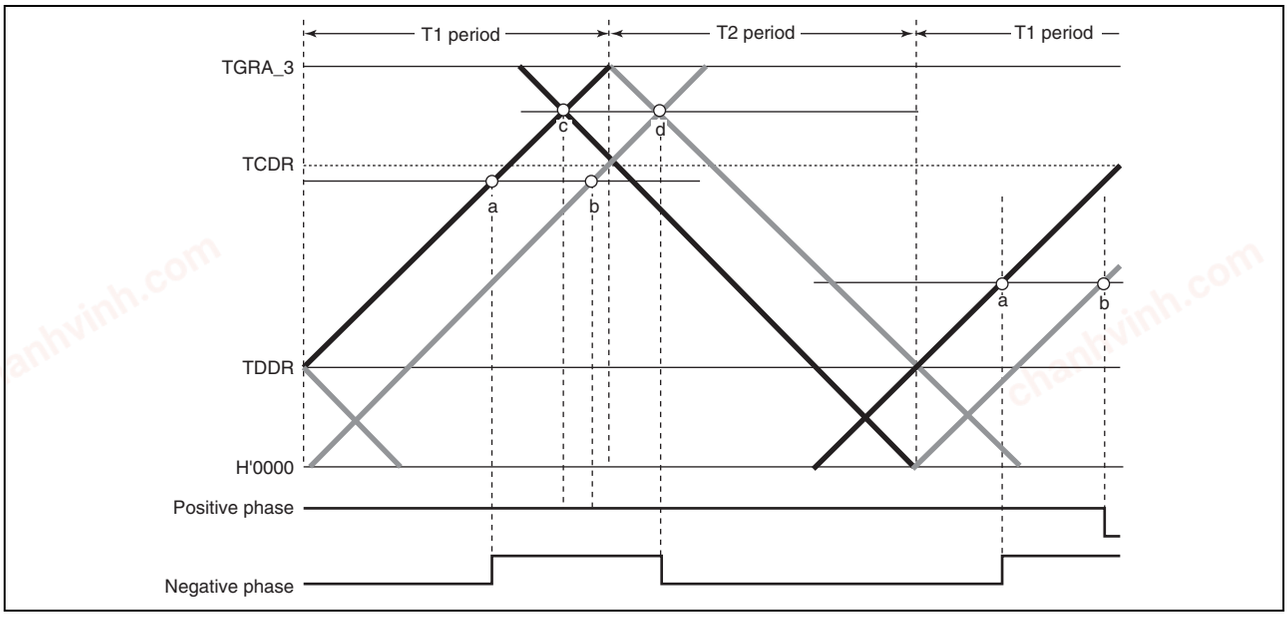


Figure 10.47 Example of Complementary PWM Mode Waveform Output (2)

chanhvinh.com

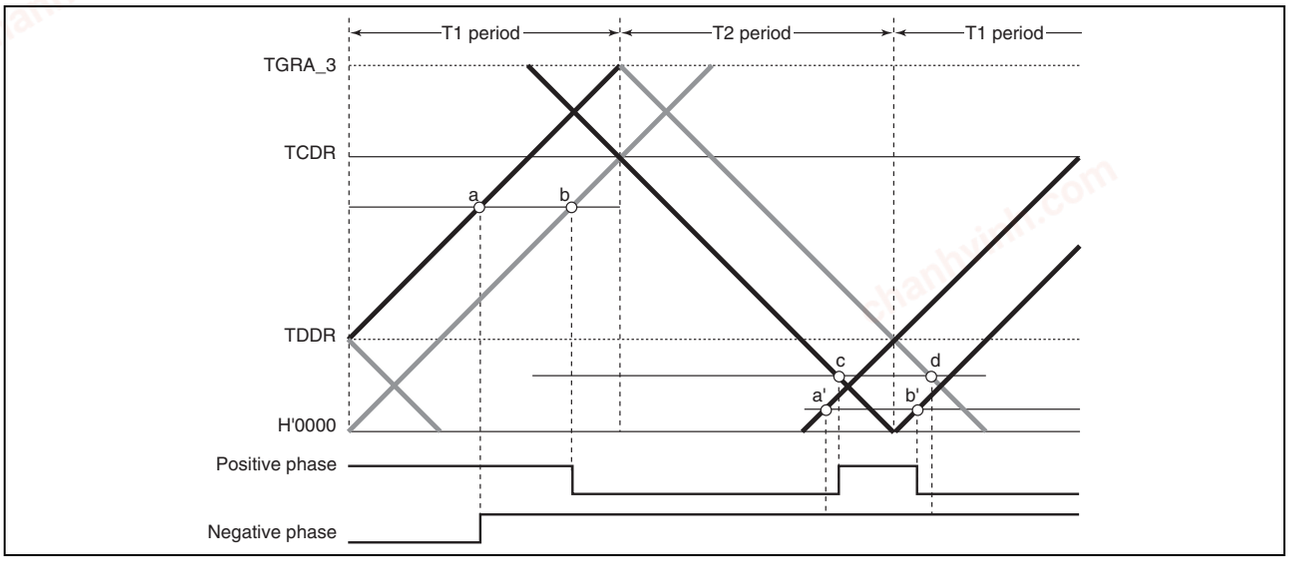


Figure 10.48 Example of Complementary PWM Mode Waveform Output (3)

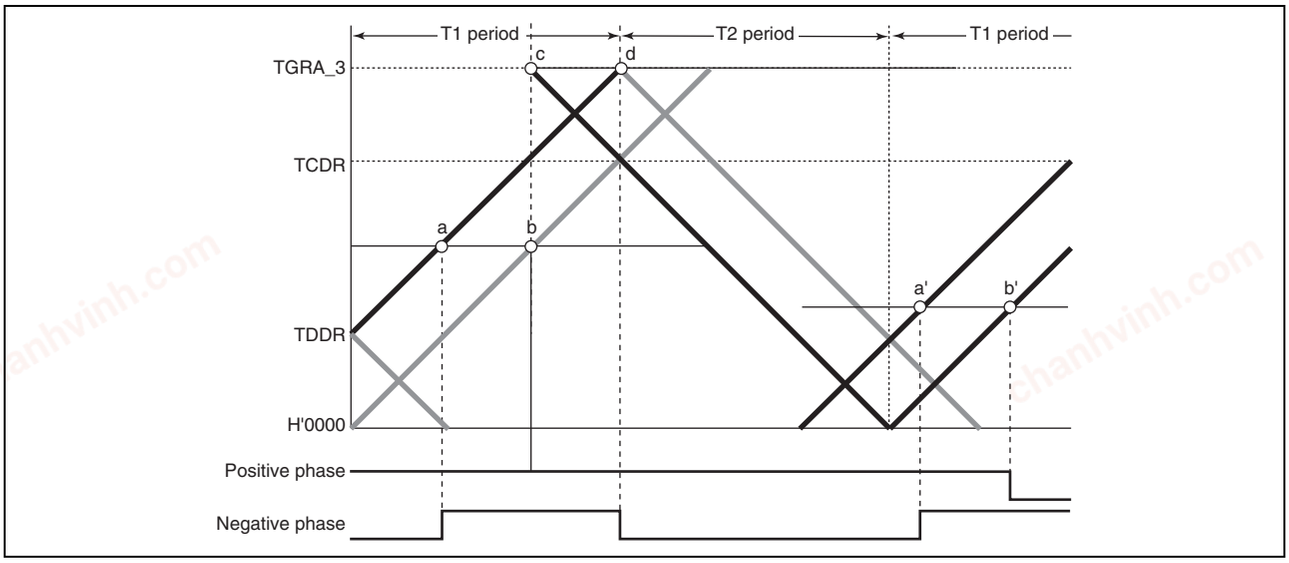


Figure 10.49 Example of Complementary PWM Mode 0% and 100% Waveform Output (1)

chanhvinh.com

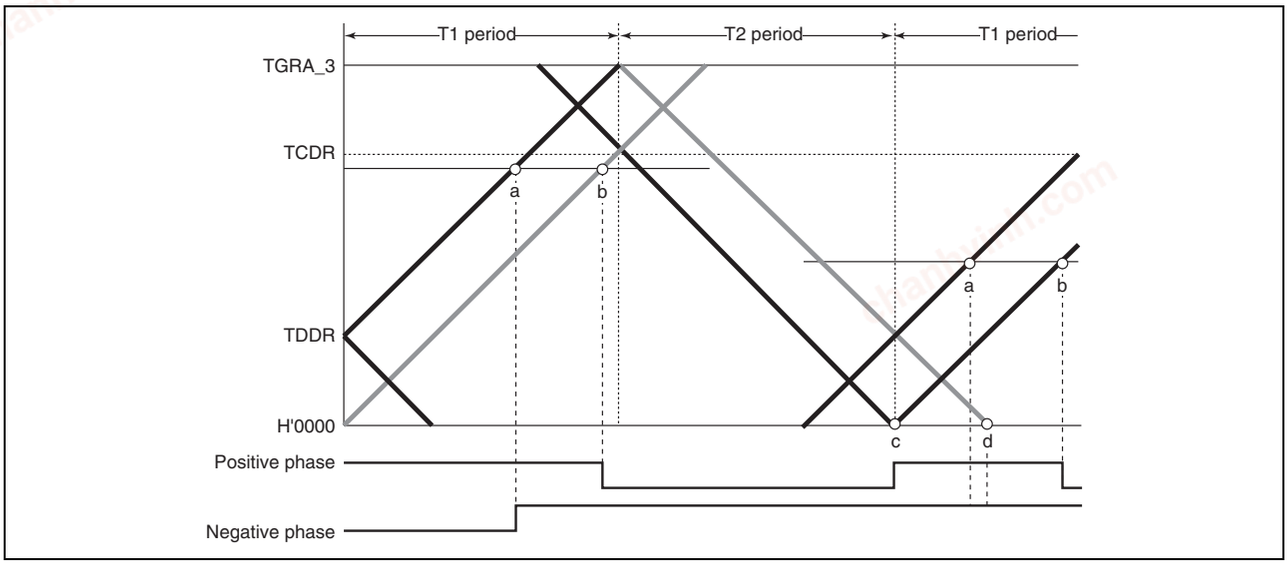


Figure 10.50 Example of Complementary PWM Mode 0% and 100% Waveform Output (2)

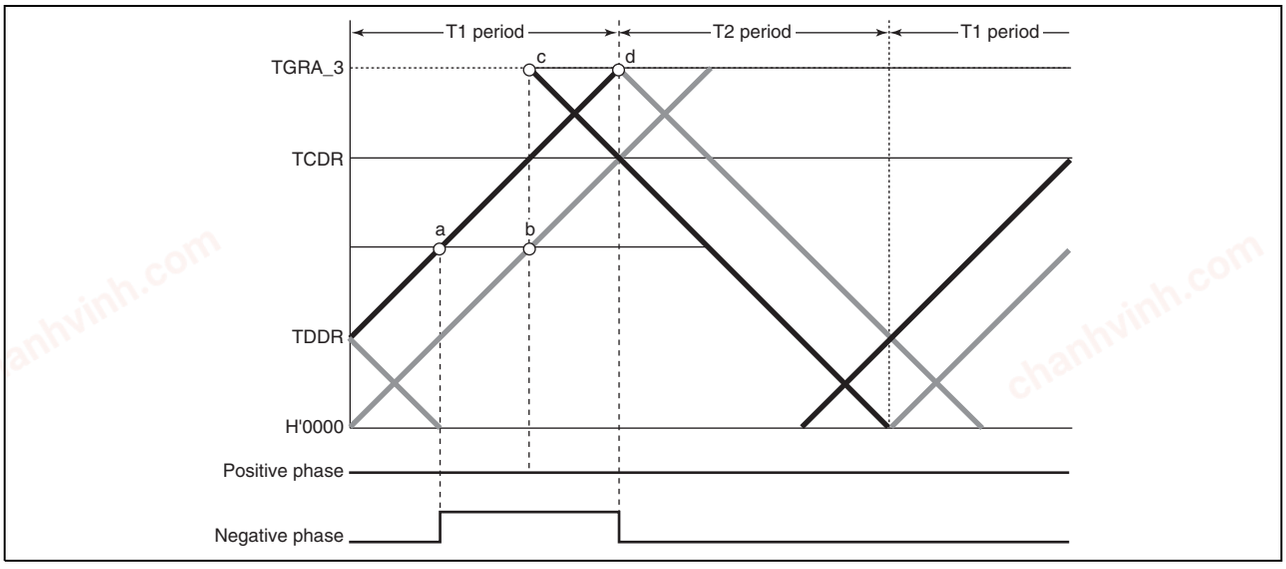


Figure 10.51 Example of Complementary PWM Mode 0% and 100% Waveform Output (3)

chanhvinh.com

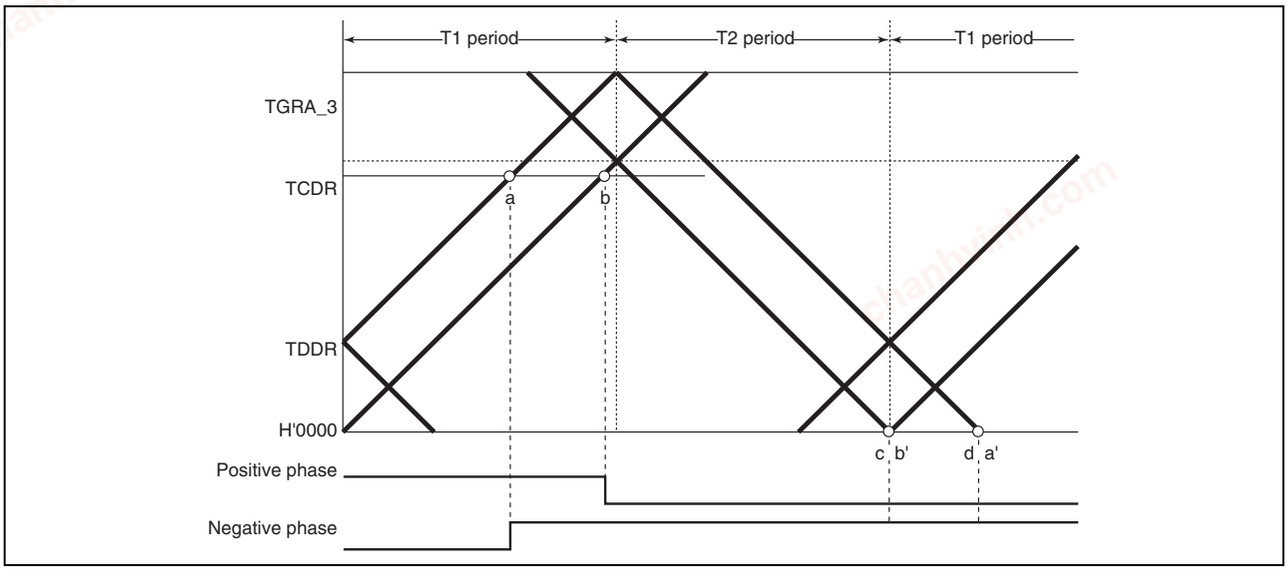


Figure 10.52 Example of Complementary PWM Mode 0% and 100% Waveform Output (4)

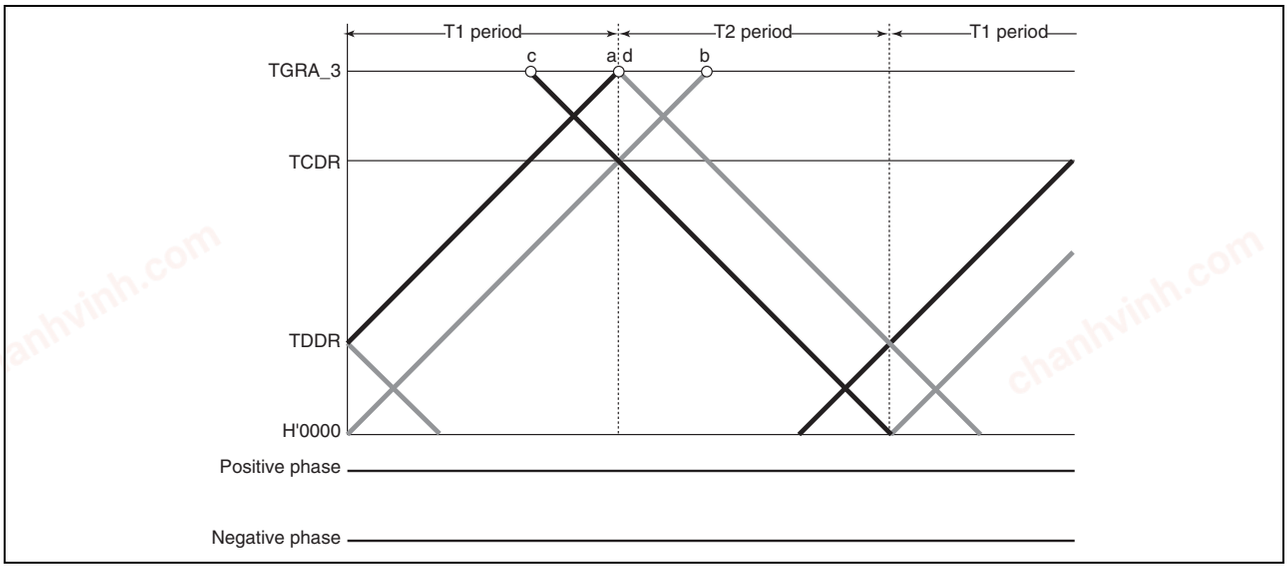


Figure 10.53 Example of Complementary PWM Mode 0% and 100% Waveform Output (5)

chanhvinh.com

(k) Complementary PWM Mode 0% and 100% Duty Output

In complementary PWM mode, 0% and 100% duty cycles can be output as required. Figure 10.49 to Figure 10.53 show output examples.

100% duty output is performed when the compare register value is set to H'0000. The waveform in this case has a positive phase with a 100% on-state. 0% duty output is performed when the compare register value is set to the same value as TGRA_3. The waveform in this case has a positive phase with a 100% off-state.

On and off compare-matches occur simultaneously, but if a turn-on compare-match and turn-off compare-match for the same phase occur simultaneously, both compare-matches are ignored and the waveform does not change.

(l) Toggle Output Synchronized with PWM Cycle

In complementary PWM mode, toggle output can be performed in synchronization with the PWM carrier cycle by setting the PSYE bit to 1 in the timer output control register (TOCR). An example of a toggle output waveform is shown in Figure 10.54.

This output is toggled by a compare-match between TCNT_3 and TGRA_3 and a compare-match between TCNT_4 and H'0000.

The output pin for this toggle output is the TIOC3A pin. The initial output is 1.

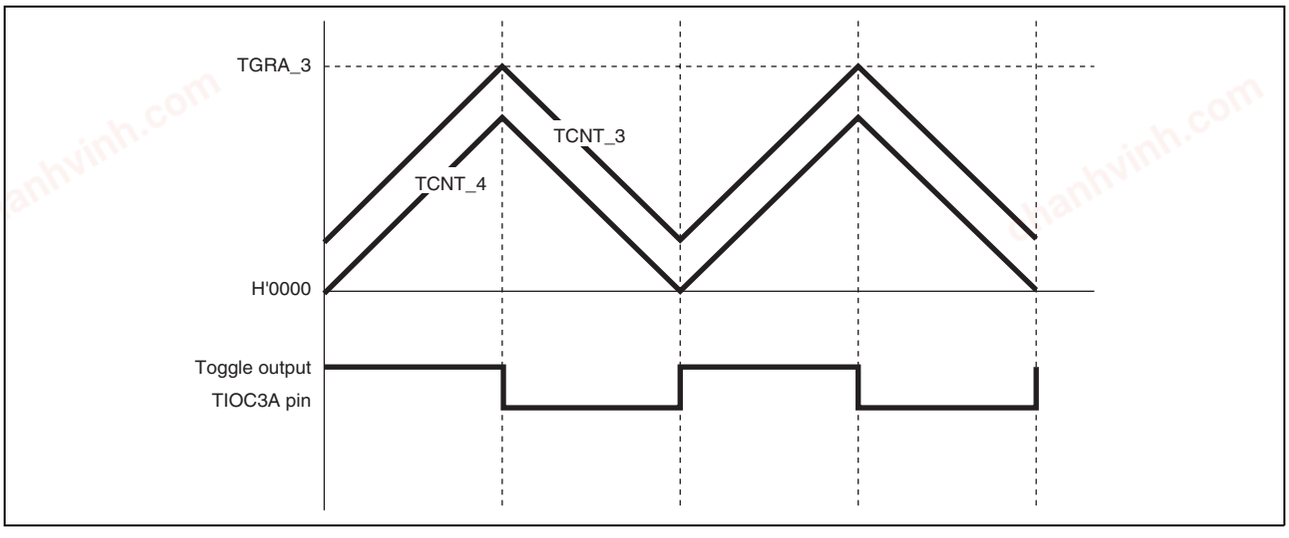


Figure 10.54 Example of Toggle Output Waveform Synchronized with PWM Output

chanhvinh.com

(m) Counter Clearing by Another Channel

In complementary PWM mode, by setting a mode for synchronization with another channel by means of the timer synchronous register (TSYR), and selecting synchronous clearing with bits CCLR2 to CCLR0 in the timer control register (TCR), it is possible to have TCNT_3, TCNT_4, and TCNTS cleared by another channel.

Figure 10.55 illustrates the operation.

Use of this function enables counter clearing and restarting to be performed by means of an external signal.

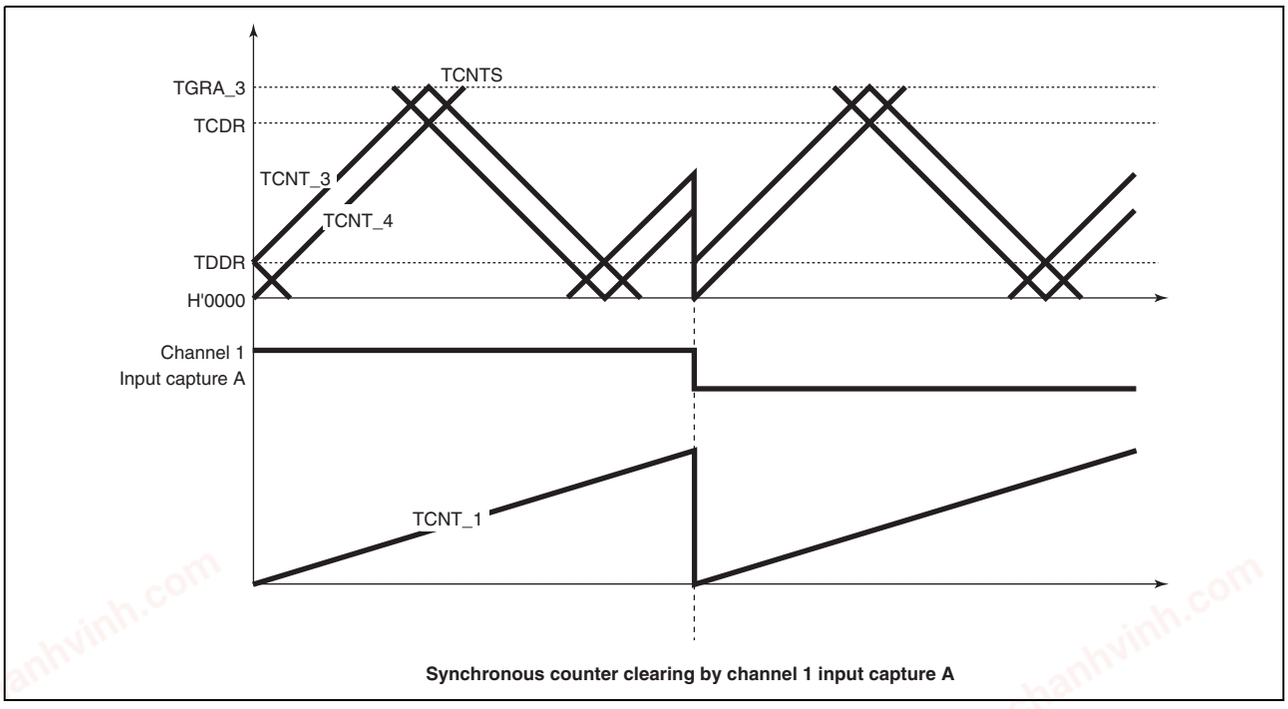


Figure 10.55 Counter Clearing Synchronized with Another Channel

chanhvinh.com

(n) Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Setting the WRE bit in TWCR to 1 suppresses initial output when synchronous counter clearing occurs in the Tb interval at the trough in complementary PWM mode and controls abrupt change in duty cycle at synchronous counter clearing.

Initial output suppression is applicable only when synchronous clearing occurs in the Tb interval at the trough as indicated by (10) or (11) in Figure 10.56. When synchronous clearing occurs outside that interval, the initial value specified by the OLS bits in TOCR is output. Even in the Tb interval at the trough, if synchronous clearing occurs in the initial value output period (indicated by (1) in Figure 10.56) immediately after the counters start operation, initial value output is not suppressed.

When using the initial output suppression function, make sure to set compare registers TGRB_3, TGRA_4, and TGRB_4 to a value twice or more the setting of dead time data register TDDR. If synchronous clearing occurs with the compare registers set to a value less than twice the setting of TDDR, the PWM output dead time may be too short (or nonexistent) or illegal active-level PWM negative-phase output may occur during the initial output suppression interval. For details, see section 10.7.23, Notes on Output Waveform Control During Synchronous Counter Clearing in Complementary PWM Mode.

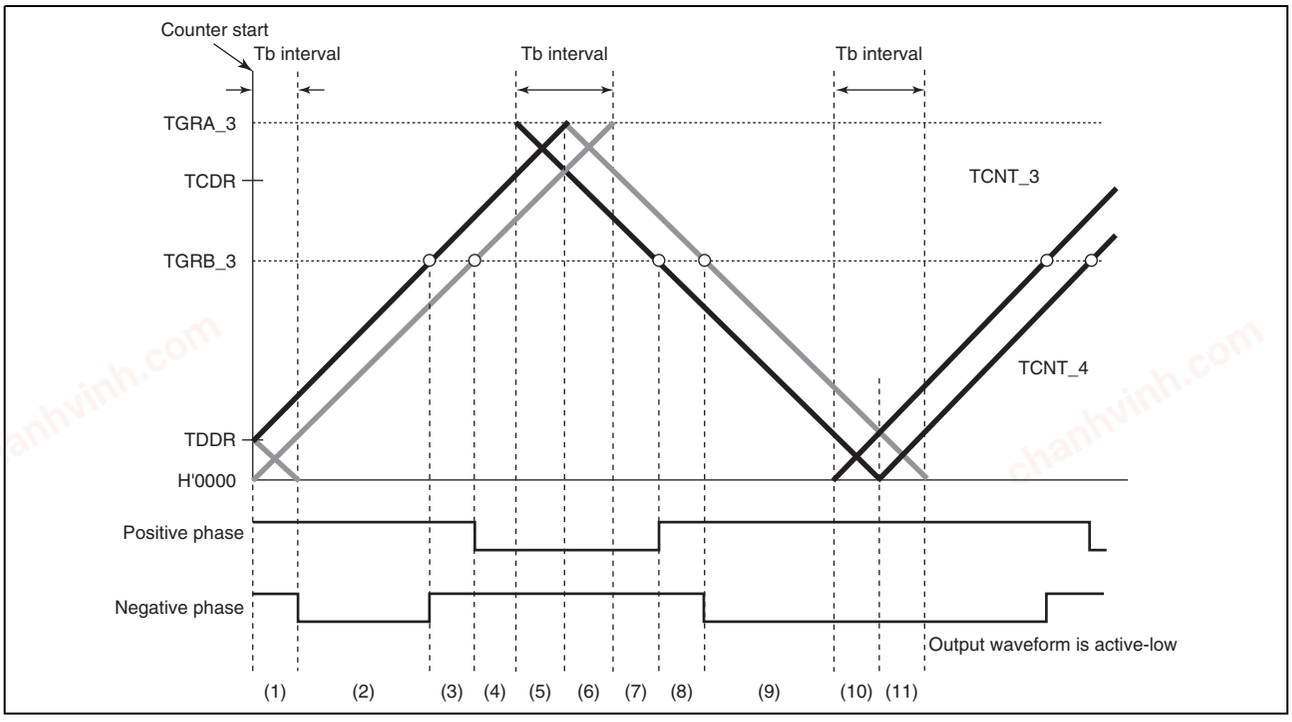


Figure 10.56 Timing for Synchronous Counter Clearing

chanhvinh.com

- Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode
  An example of the procedure for setting output waveform control at synchronous counter clearing in complementary PWM mode is shown in Figure 10.57.

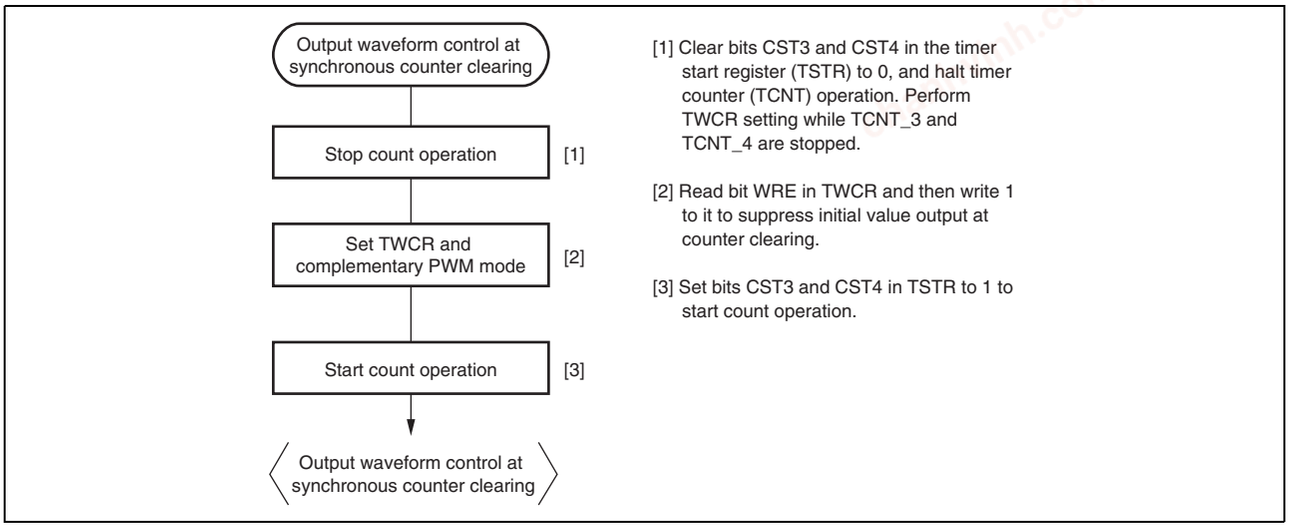


Figure 10.57 Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

- Examples of Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode
  Figure 10.58 to Figure 10.61 show examples of output waveform control in which this module operates in complementary PWM mode and synchronous counter clearing is generated while the WRE bit in TWCR is set to 1. In the examples shown in Figure 10.58 to Figure 10.61, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in Figure 10.56, respectively.

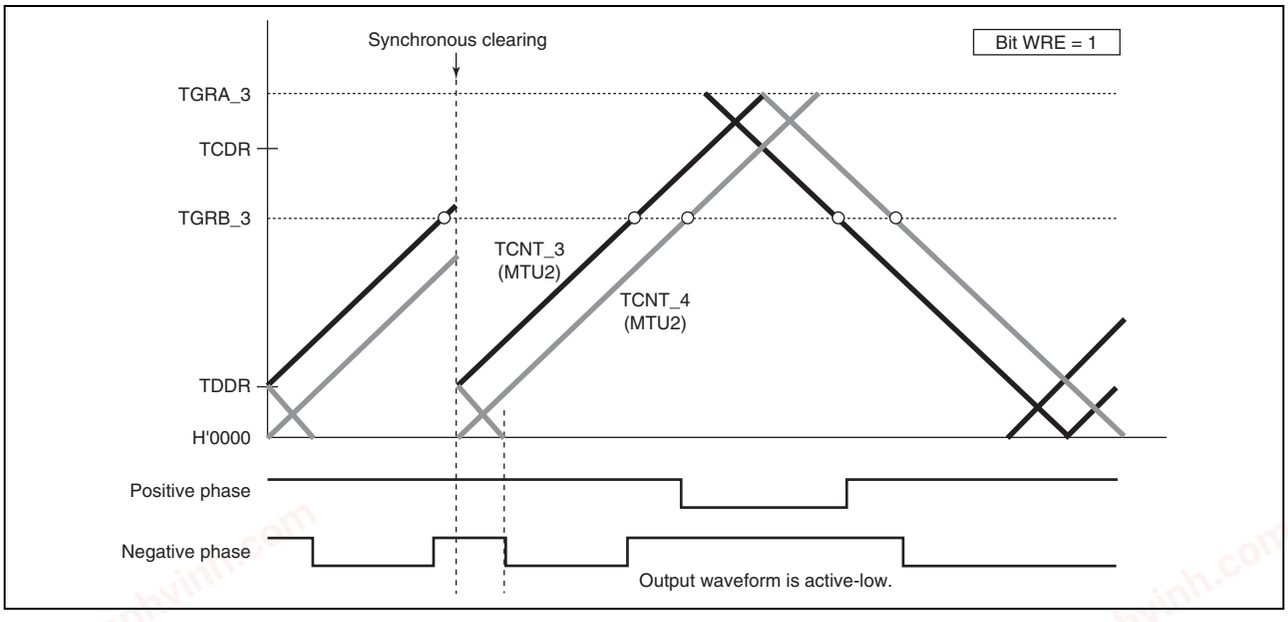


Figure 10.58 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 10.56; Bit WRE of TWCR is 1)

chanhvinh.com

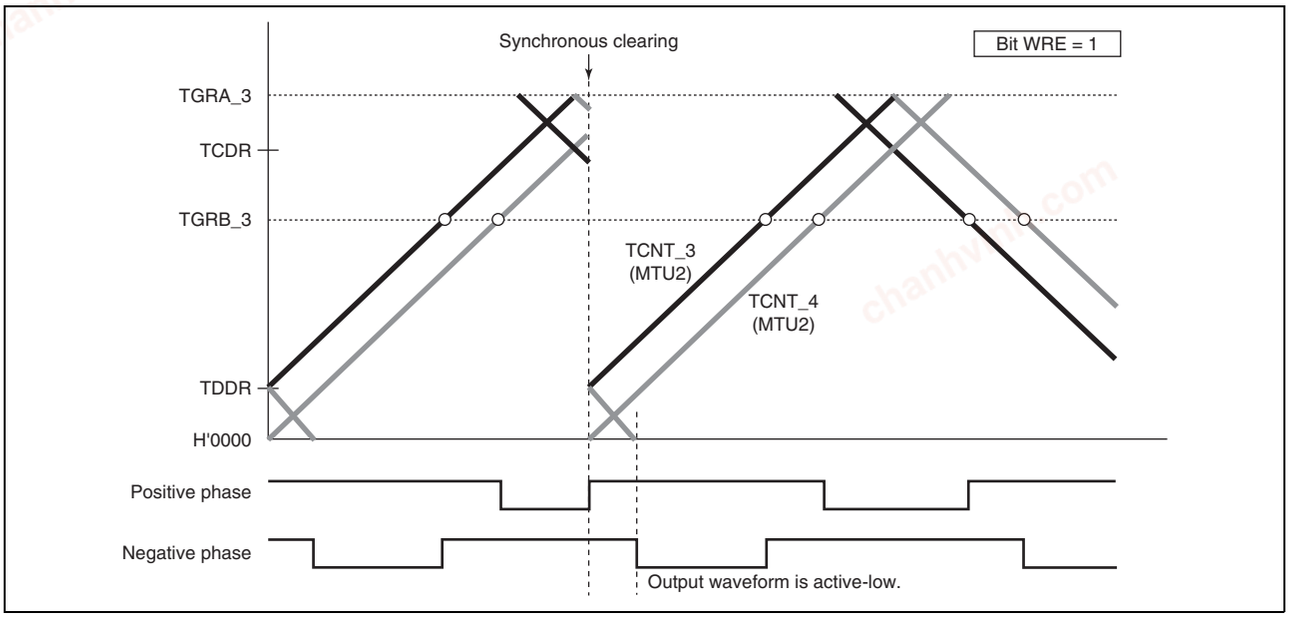


Figure 10.59 Example of Synchronous Clearing in Interval Tb at Crest (Timing (6) in Figure 10.56; Bit WRE of TWCR is 1)

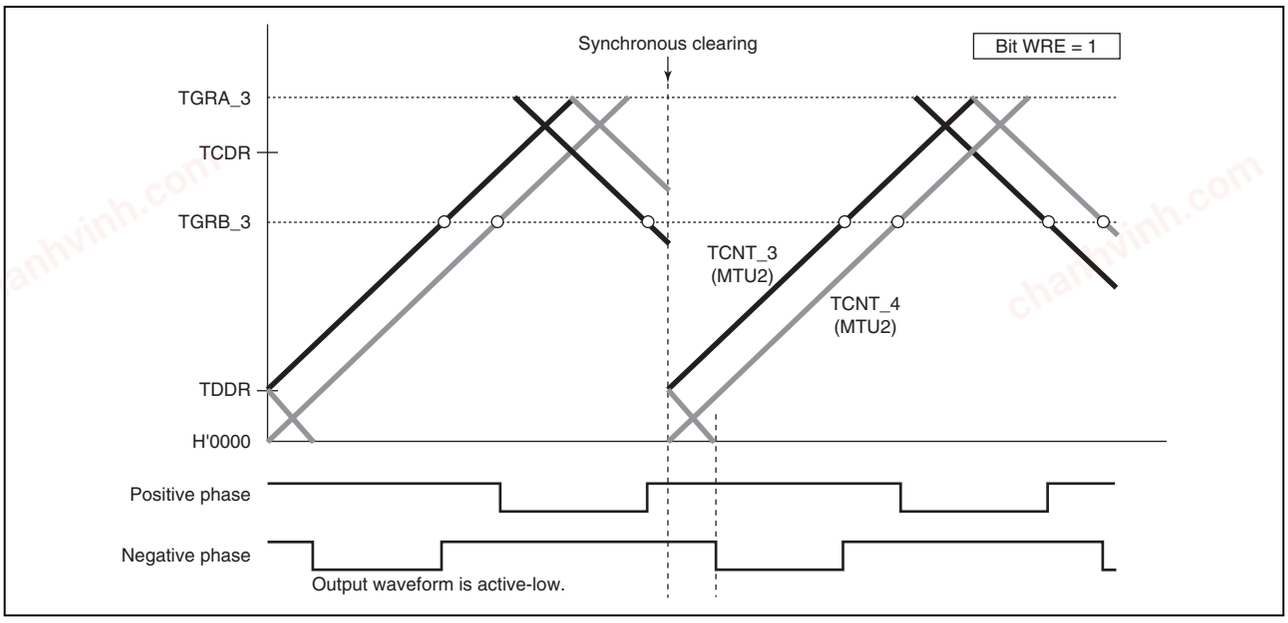


Figure 10.60 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 10.56; Bit WRE of TWCR is 1)

chanhvinh.com

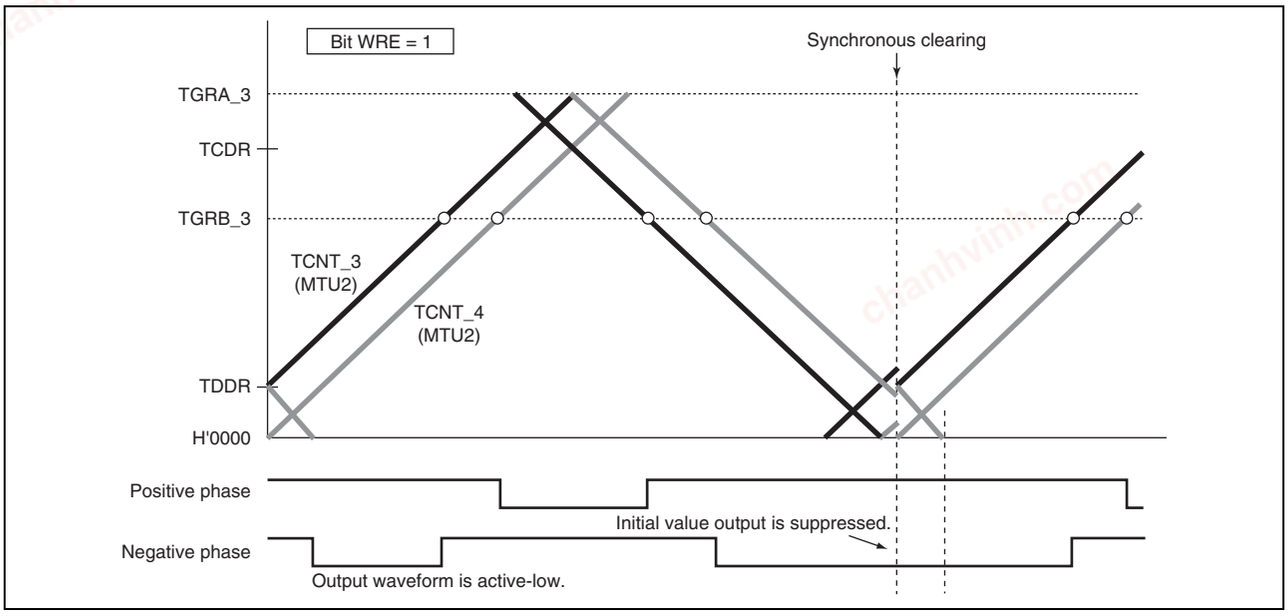


Figure 10.61 Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 10.56; Bit WRE of TWCR is 1)

chanhvinh.com

(o) Counter Clearing by TGRA_3 Compare Match

In complementary PWM mode, by setting the CCE bit in the timer waveform control register (TWCR), it is possible to have TCNT_3, TCNT_4, and TCNTS cleared by TGRA_3 compare match.

Figure 10.62 illustrates an operation example.

Note 1. Use this function only in complementary PWM mode 1 (transfer at crest)
Note 2. Do not specify synchronous clearing by another channel (do not set the SYNC0 to SYNC4 bits in the timer synchronous register (TSYR) to 1).
Note 3. Do not set the PWM duty value to H'0000.
Note 4. Do not set the PSYE bit in timer output control register 1 (TOCR1) to 1.

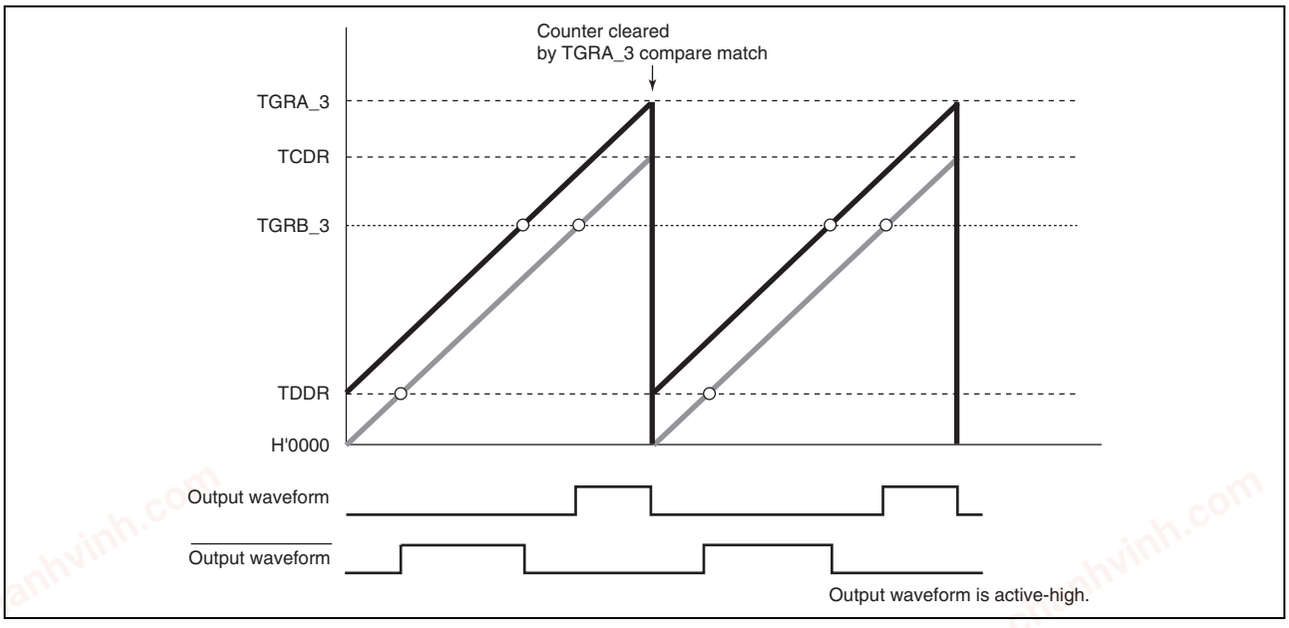


Figure 10.62 Example of Counter Clearing Operation by TGRA_3 Compare Match

chanhvinh.com

(p) Example of AC Synchronous Motor (Brushless DC Motor) Drive Waveform Output

In complementary PWM mode, a brushless DC motor can easily be controlled using the timer gate control register (TGCR). Figure 10.63 to Figure 10.66 show examples of brushless DC motor drive waveforms created using TGCR.

When output phase switching for a 3-phase brushless DC motor is performed by means of external signals detected with a Hall element, etc., clear the FB bit in TGCR to 0. In this case, the external signals indicating the polarity position are input to channel 0 timer input pins TIOC0A, TIOC0B, and TIOC0C (set with the general I/O ports). When an edge is detected at pin TIOC0A, TIOC0B, or TIOC0C, the output on/off state is switched automatically.

When the FB bit is 1, the output on/off state is switched when the UF, VF, or WF bit in TGCR is cleared to 0 or set to 1. The drive waveforms are output from the complementary PWM mode 6-phase output pins. With this 6-phase output, in the case of on output, it is possible to use complementary PWM mode output and perform chopping output by setting the N bit or P bit to 1. When the N bit or P bit is 0, level output is selected.

The 6-phase output active level (on output level) can be set with the OLSN and OLSP bits in the timer output control register (TOCR) regardless of the setting of the N and P bits.

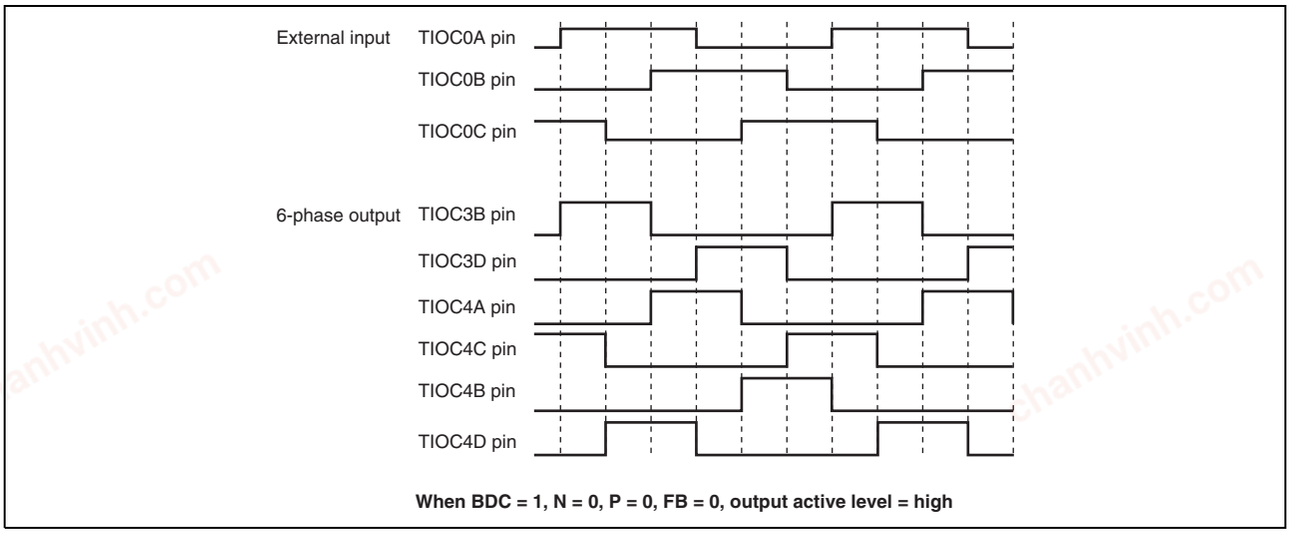


Figure 10.63 Example of Output Phase Switching by External Input (1)

chanhvinh.com

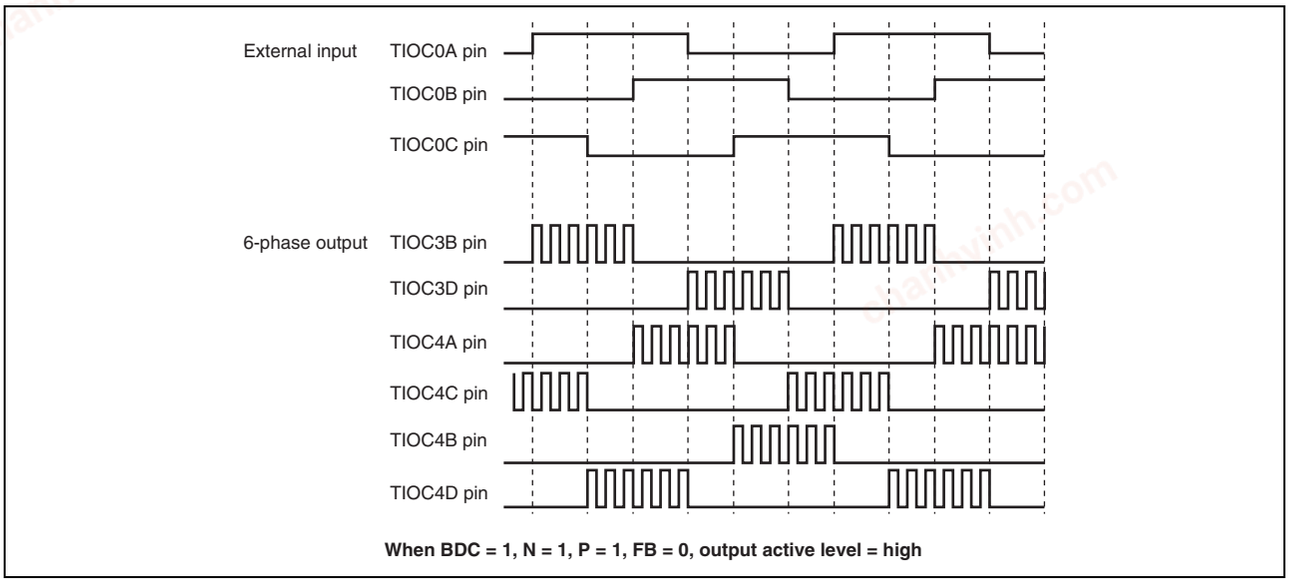


Figure 10.64 Example of Output Phase Switching by External Input (2)

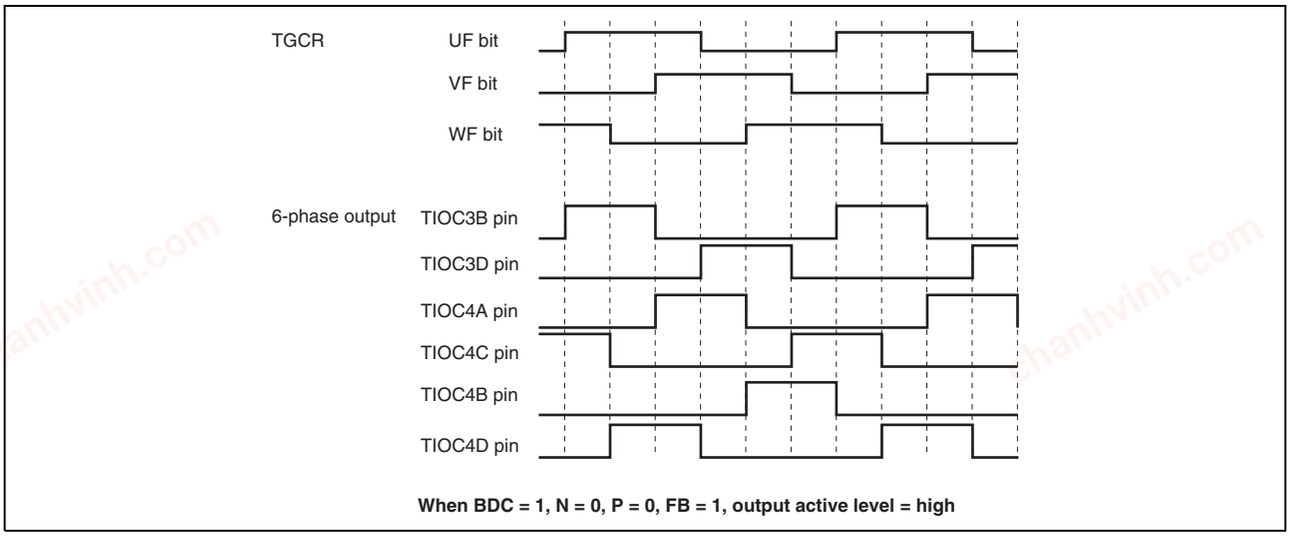


Figure 10.65 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (1)

chanhvinh.com

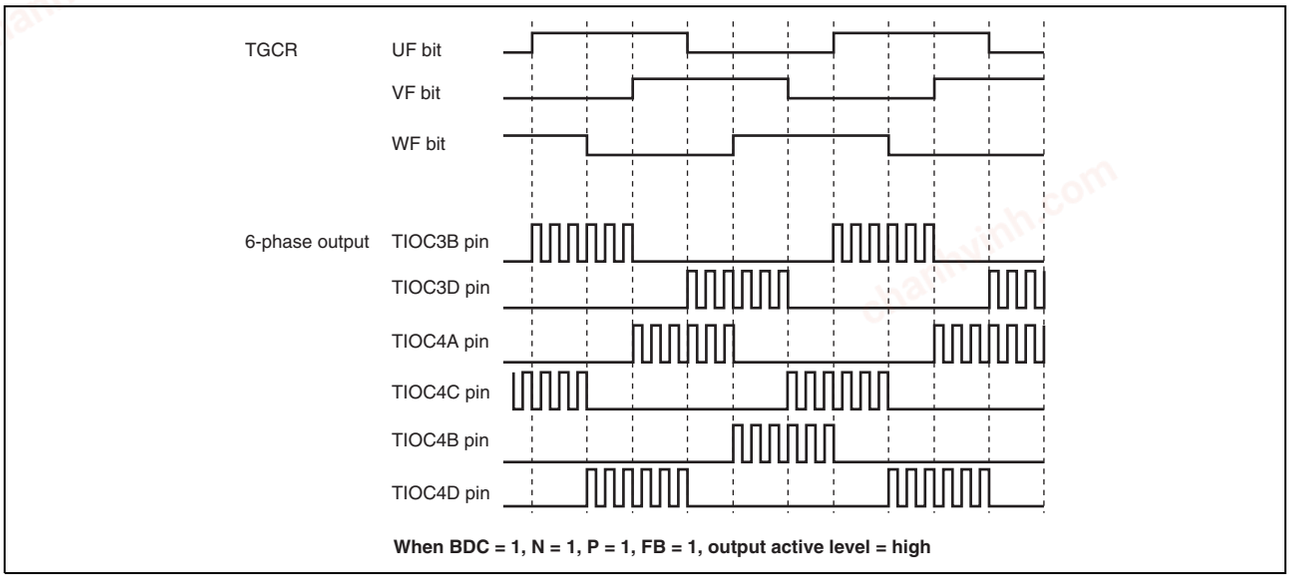


Figure 10.66 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (2)

(q) A/D Converter Start Request Setting

In complementary PWM mode, an A/D converter start request can be issued using a TGRA_3 compare-match, TCNT_4 underflow (trough), or compare-match on a channel other than channels 3 and 4.

When start requests using a TGRA_3 compare-match are specified, A/D conversion can be started at the crest of the TCNT_3 count.

A/D converter start requests can be set by setting the TTGE bit to 1 in the timer interrupt enable register (TIER). To issue an A/D converter start request at a TCNT_4 underflow (trough), set the TTGE2 bit in TIER_4 to 1.

chanhvinh.com

(3) Interrupt Skipping in Complementary PWM Mode

Interrupts TGIA_3 (at the crest) and TCIV_4 (at the trough) in channels 3 and 4 can be skipped up to seven times by making settings in the timer interrupt skipping set register (TITCR).

Transfers from a buffer register to a temporary register or a compare register can be skipped in coordination with interrupt skipping by making settings in the timer buffer transfer register (TBTER). For the linkage with buffer registers, refer to description (c), Buffer Transfer Control Linked with Interrupt Skipping, below.

A/D converter start requests generated by the A/D converter start request delaying function can also be skipped in coordination with interrupt skipping by making settings in the timer A/D converter request control register (TADCR). For the linkage with the A/D converter start request delaying function, refer to section 10.4.9, A/D Converter Start Request Delaying Function.

The setting of the timer interrupt skipping setting register (TITCR) must be done while the TGIA_3 and TCIV_4 interrupt requests are disabled by the settings of TIER_3 and TIER_4 along with under the conditions in which TGFA_3 and TCFV_4 flag settings by compare match never occur. Before changing the skipping count, be sure to clear the T3AEN and T4VEN bits to 0 to clear the skipping counter.

(a) Example of Interrupt Skipping Operation Setting Procedure

Figure 10.67 shows an example of the interrupt skipping operation setting procedure. Figure 10.68 shows the periods during which interrupt skipping count can be changed.

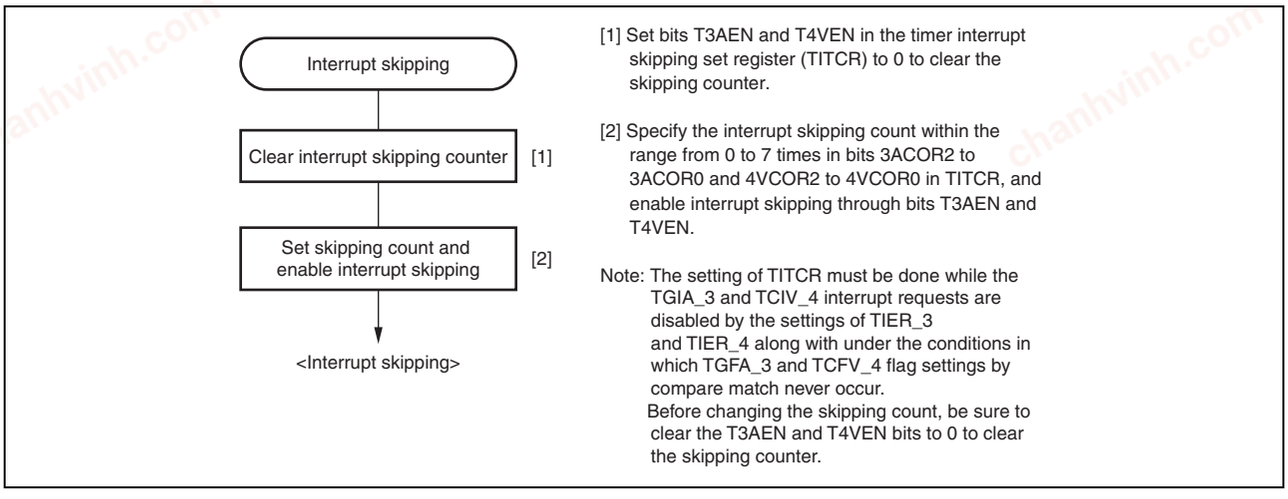


Figure 10.67 Example of Interrupt Skipping Operation Setting Procedure

chanhvinh.com

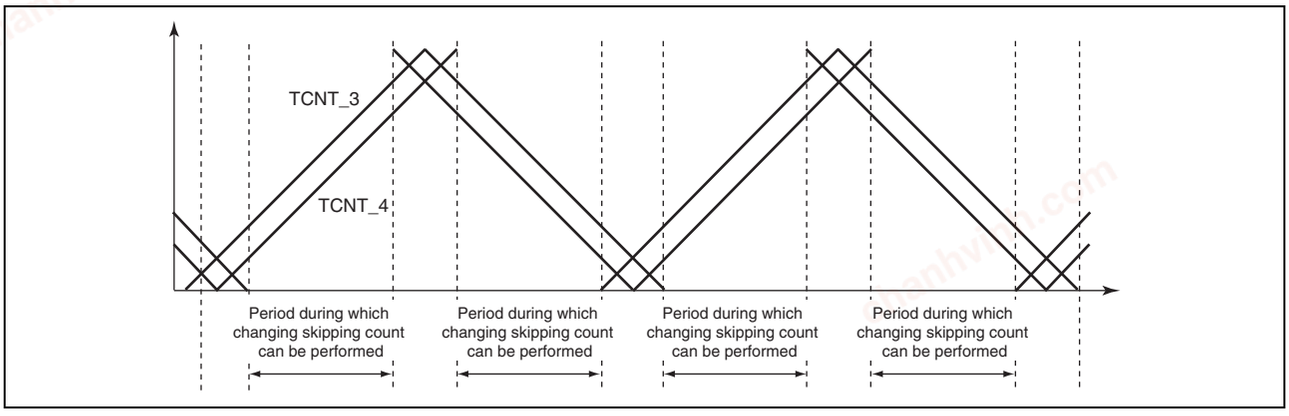


Figure 10.68 Periods during which Interrupt Skipping Count can be Changed

(b) Example of Interrupt Skipping Operation

Figure 10.69 shows an example of TGIA_3 interrupt skipping in which the interrupt skipping count is set to three by the 3ACOR bit and the T3AEN bit is set to 1 in the timer interrupt skipping set register (TITCR).

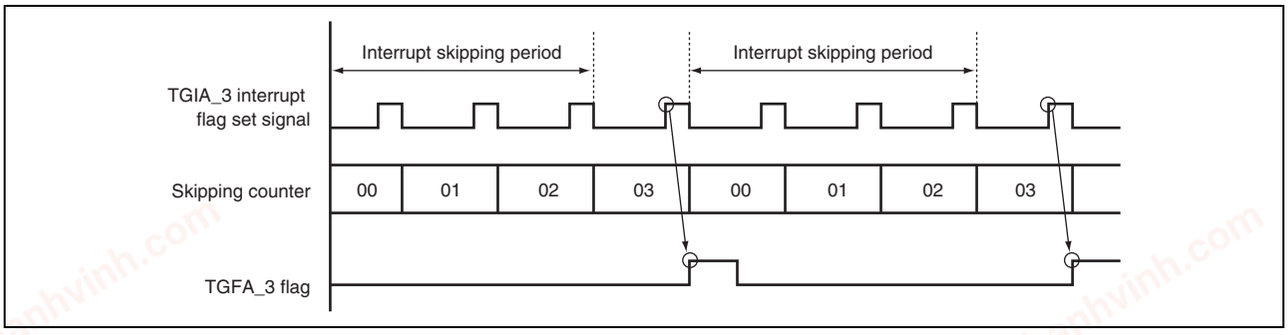


Figure 10.69 Example of Interrupt Skipping Operation

chanhvinh.com

(c) Buffer Transfer Control Linked with Interrupt Skipping

In complementary PWM mode, whether to transfer data from a buffer register to a temporary register and whether to link the transfer with interrupt skipping can be specified with the BTE1 and BTE0 bits in the timer buffer transfer set register (TBTER).

Figure 10.70 shows an example of operation when buffer transfer is suppressed (BTE1 = 0 and BTE0 = 1). While this setting is valid, data is not transferred from the buffer register to the temporary register.

Figure 10.71 shows an example of operation when buffer transfer is linked with interrupt skipping (BTE1 = 1 and BET0 = 0). While this setting is valid, data is not transferred from the buffer register to the temporary register outside the buffer transfer-enabled period. Depending on the rewrite timing from the interrupt generation to the buffer register, there are two types of the transfer timing such as from the buffer register to the temporary register and from the temporary register to the general register.

Note that the buffer transfer-enabled period depends on the T3AEN and T4VEN bit settings in the timer interrupt skipping set register (TITCR). Figure 10.72 shows the relationship between the T3AEN and T4VEN bit settings in TITCR and buffer transfer-enabled period.

Note: This function must always be used in combination with interrupt skipping.
When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) are cleared to 0 or the skipping count set bits (3ACOR and 4VCOR) in TITCR are cleared to 0), make sure that buffer transfer is not linked with interrupt skipping (clear the BTE1 bit in the timer buffer transfer set register (TBTER) to 0). If buffer transfer is linked with interrupt skipping while interrupt skipping is disabled, buffer transfer is never performed.

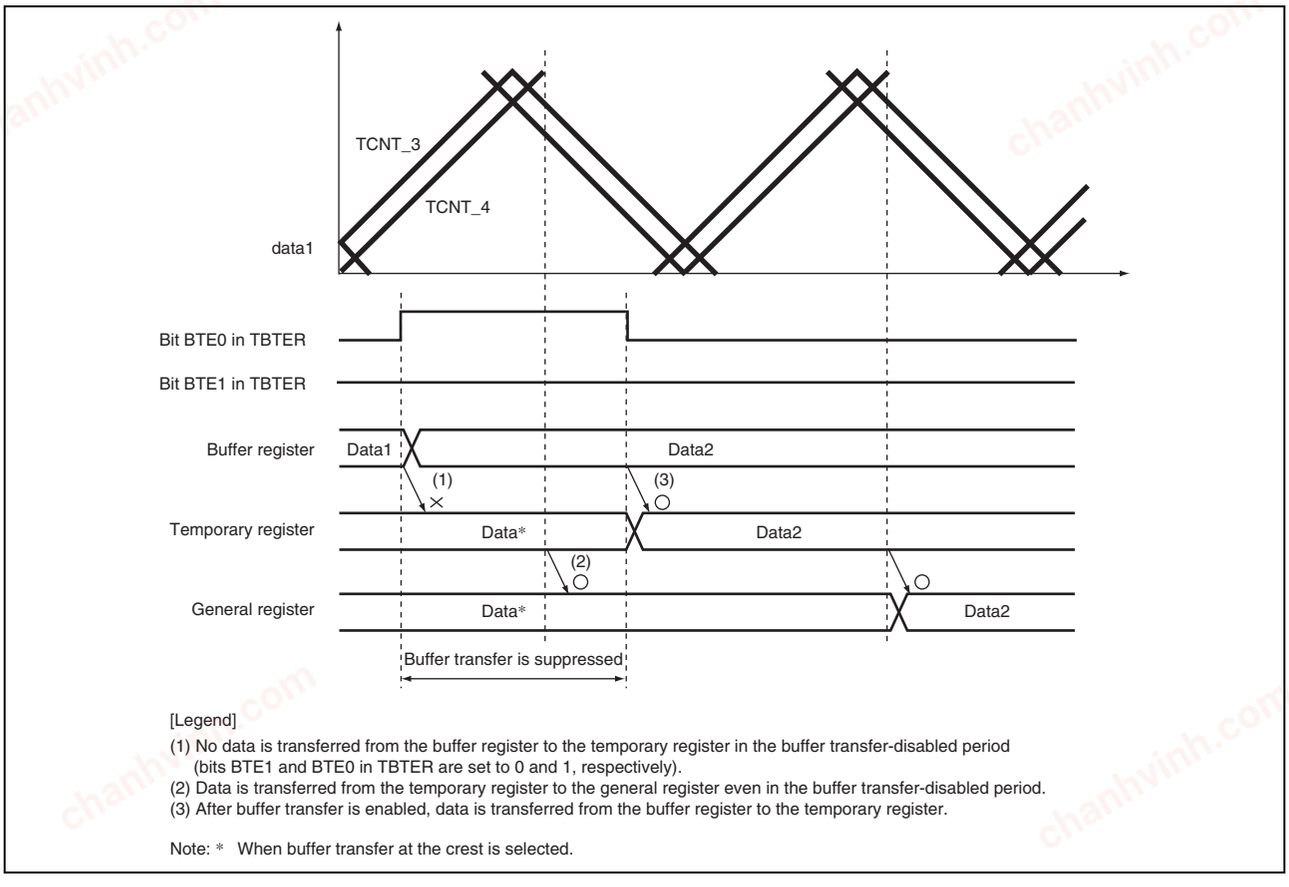
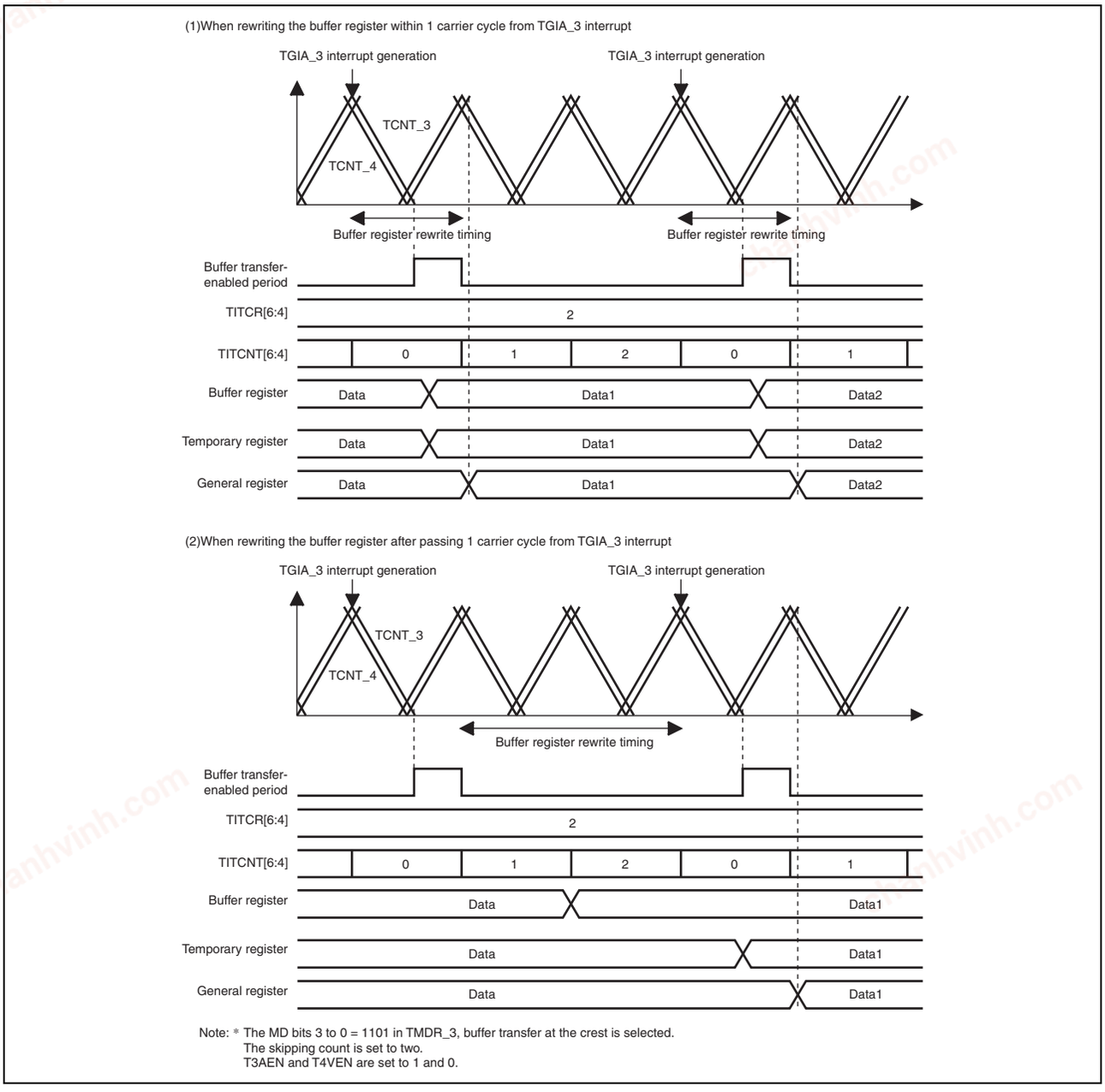


Figure 10.70 Example of Operation when Buffer Transfer is Suppressed (BTE1 = 0 and BTE0 = 1)

chanhvinh.com



Note: * The MD bits 3 to 0 = 1101 in TMDR_3, buffer transfer at the crest is selected.
The skipping count is set to two.
T3AEN and T4VEN are set to 1 and 0.

Figure 10.71 Example of Operation when Buffer Transfer is Linked with Interrupt Skipping (BTE1 = 1 and BTE0 = 0)

chanhvinh.com

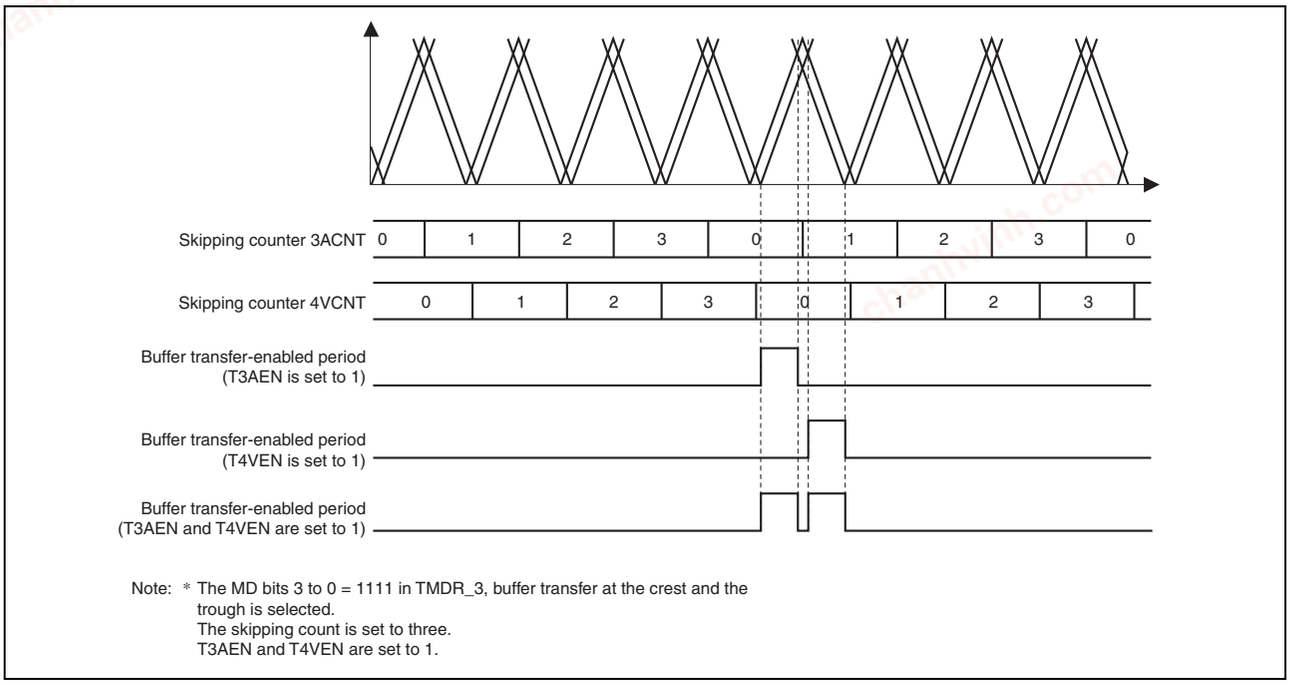


Figure 10.72 Relationship between Bits T3AEN and T4VEN in TITCR and Buffer Transfer-Enabled Period

(4) Complementary PWM Mode Output Protection Function

Complementary PWM mode output has the following protection function.

(a) Register and counter miswrite prevention function

With the exception of the buffer registers, which can be rewritten at any time, access by the CPU can be enabled or disabled for the mode registers, control registers, compare registers, and counters used in complementary PWM mode by means of the RWE bit in the timer read/write enable register (TRWER). The applicable registers are some (21 in total) of the registers in channels 3 and 4 shown in the following:

- TCR_3 and TCR_4, TMDR_3 and TMDR_4, TIORH_3 and TIORH_4, TIORL_3 and TIORL_4, TIER_3 and TIER_4, TCNT_3 and TCNT_4, TGRA_3 and TGRA_4, TGRB_3 and TGRB_4, TOER, TOCR, TGCR, TCDR, and TDDR.

This function enables miswriting due to CPU runaway to be prevented by disabling CPU access to the mode registers, control registers, and counters. When the applicable registers are read in the access-disabled state, undefined values are returned. Writing to these registers is ignored.

chanhvinh.com

10.4.9 A/D Converter Start Request Delaying Function

A/D converter start requests can be issued in channel 4 by making settings in the timer A/D converter start request control register (TADCR), timer A/D converter start request cycle set registers (TADCORA_4 and TADCORB_4), and timer A/D converter start request cycle set buffer registers (TADCOBRA_4 and TADCOBRB_4).

The A/D converter start request delaying function compares TCNT_4 with TADCORA_4 or TADCORB_4, and when their values match, the function issues a respective A/D converter start request (TRG4AN or TRG4BN).

A/D converter start requests (TRG4AN and TRG4BN) can be skipped in coordination with interrupt skipping by setting the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in TADCR.

- Example of Procedure for Specifying A/D Converter Start Request Delaying Function
  Figure 10.73 shows an example of procedure for specifying the A/D converter start request delaying function.

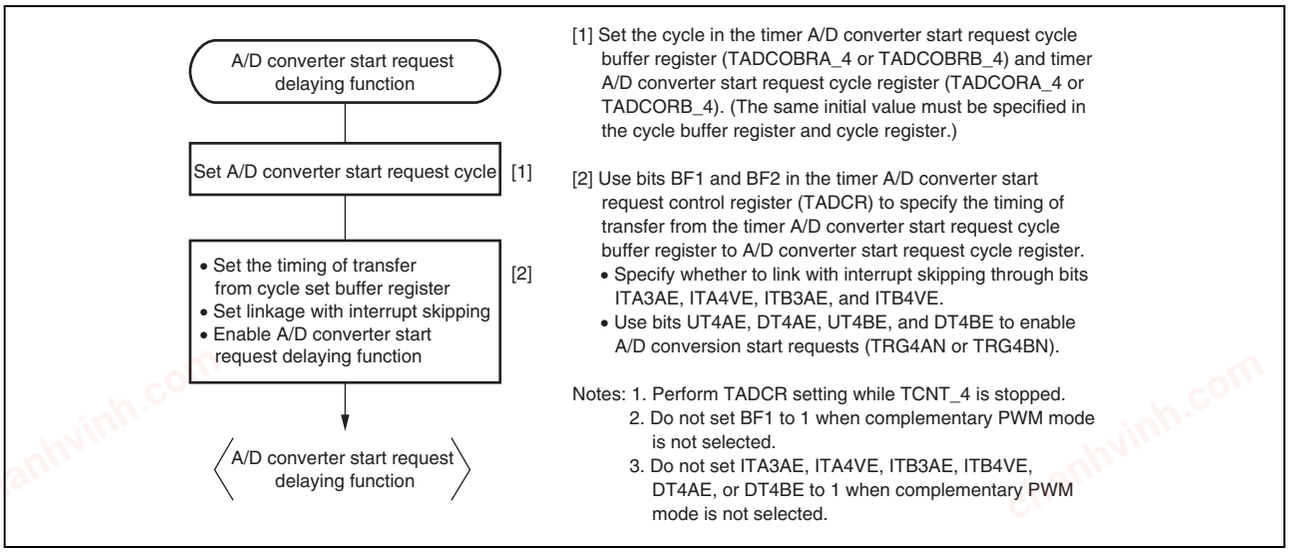


Figure 10.73 Example of Procedure for Specifying A/D Converter Start Request Delaying Function

chanhvinh.com

- Basic Operation Example of A/D Converter Start Request Delaying Function
  Figure 10.74 shows a basic example of A/D converter request signal (TRG4AN) operation when the trough of TCNT_4 is specified for the buffer transfer timing and an A/D converter start request signal is output during TCNT_4 down-counting.

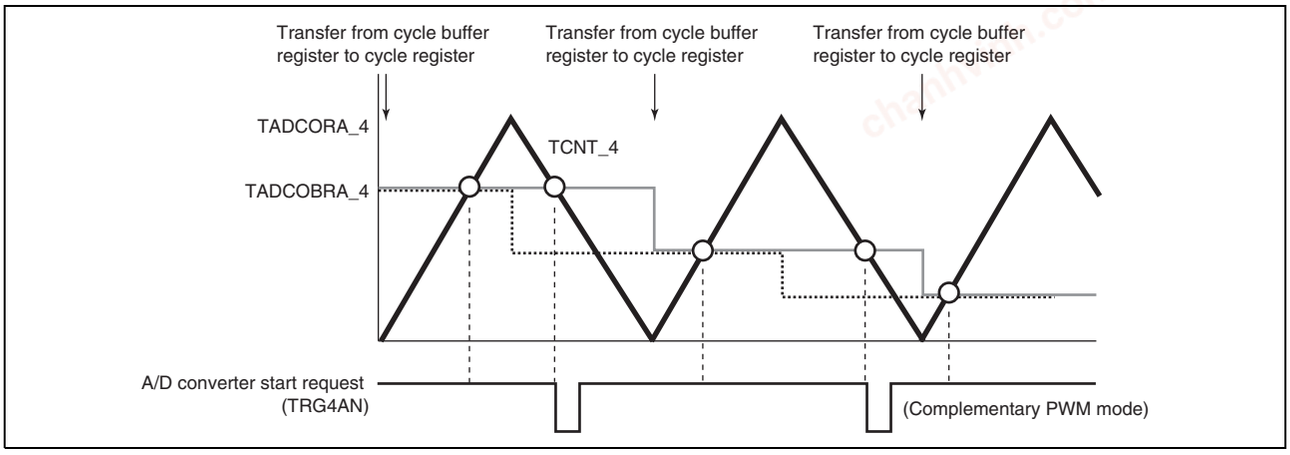


Figure 10.74 Basic Example of A/D Converter Start Request Signal (TRG4AN) Operation

- Buffer Transfer
  The data in the timer A/D converter start request cycle set registers (TADCORA_4 and TADCORB_4) is updated by writing data to the timer A/D converter start request cycle set buffer registers (TADCOBRA_4 and TADCOBRB_4). Data is transferred from the buffer registers to the respective cycle set registers at the timing selected with the BF1 and BF0 bits in the timer A/D converter start request control register (TADCR_4).

- A/D Converter Start Request Delaying Function Linked with Interrupt Skipping
  A/D converter start requests (TRG4AN and TRG4BN) can be issued in coordination with interrupt skipping by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR).
  Figure 10.75 shows an example of A/D converter start request signal (TRG4AN) operation when TRG4AN output is enabled during TCNT_4 up counting and down counting and A/D converter start requests are linked with interrupt skipping.
  Figure 10.76 shows another example of A/D converter start request signal (TRG4AN) operation when TRG4AN output is enabled during TCNT_4 up counting and A/D converter start requests are linked with interrupt skipping.

Note: This function must be used in combination with interrupt skipping.
When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) are cleared to 0 or the skipping count set bits (3ACOR and 4VCOR) in TITCR are cleared to 0), make sure that A/D converter start requests are not linked with interrupt skipping (clear the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR) to 0).

chanhvinh.com

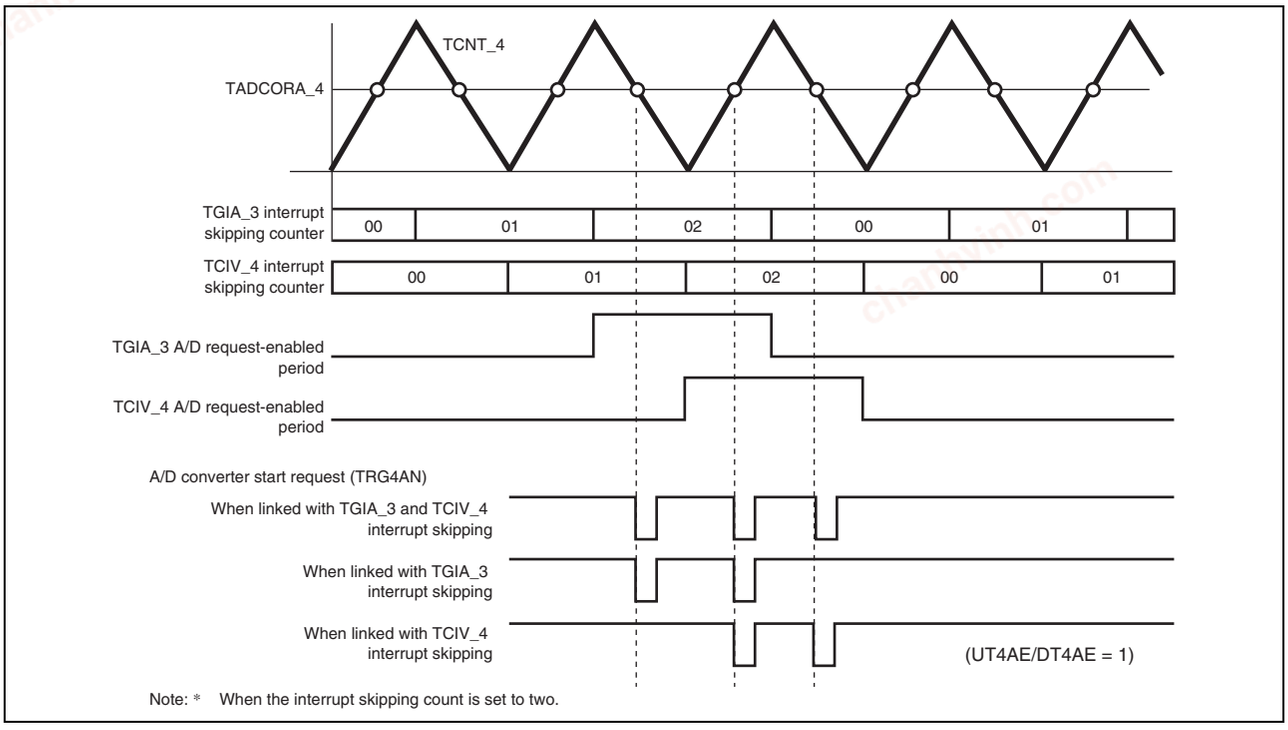


Figure 10.75 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping

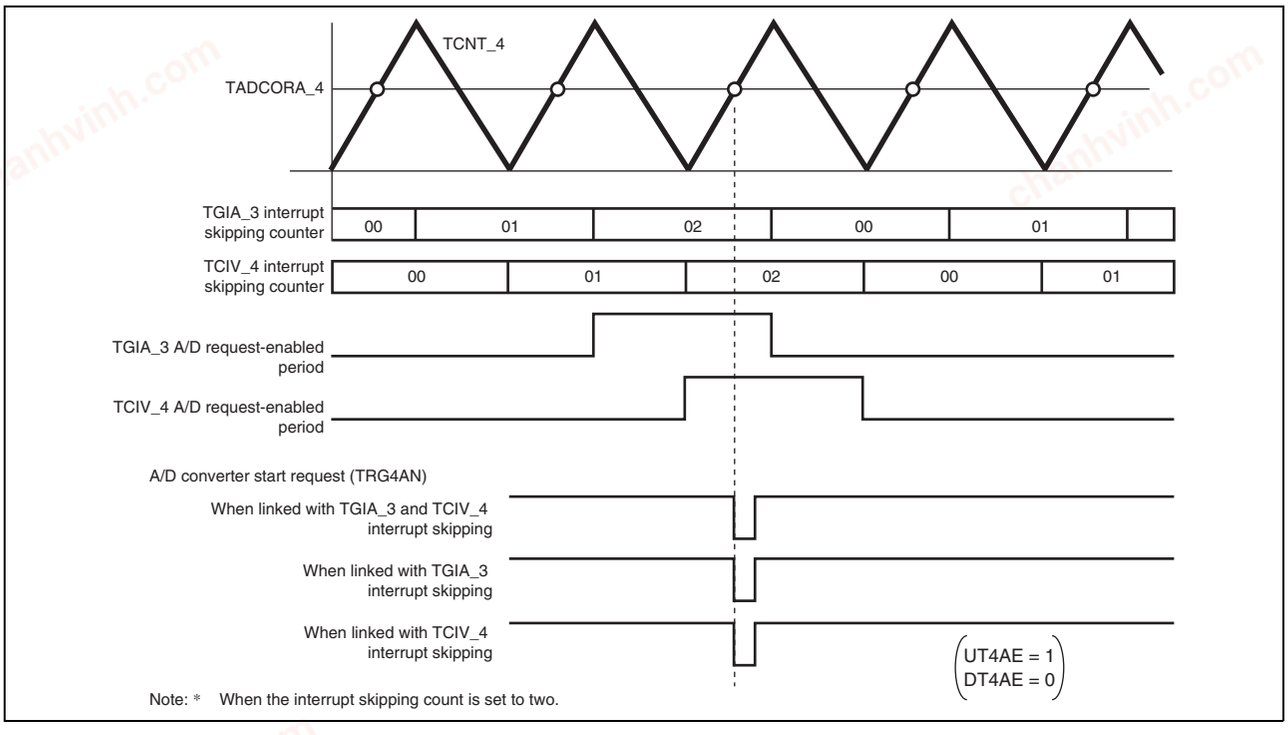


Figure 10.76 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping

chanhvinh.com

10.4.10 TCNT Capture at Crest and/or Trough in Complementary PWM Operation

The TCNT value is captured in TGR at either the crest or trough or at both the crest and trough during complementary PWM operation. The timing for capturing in TGR can be selected by TIOR.

Figure 10.77 shows an example in which TCNT is used as a free-running counter without being cleared, and the TCNT value is captured in TGR at the specified timing (either crest or trough, or both crest and trough).

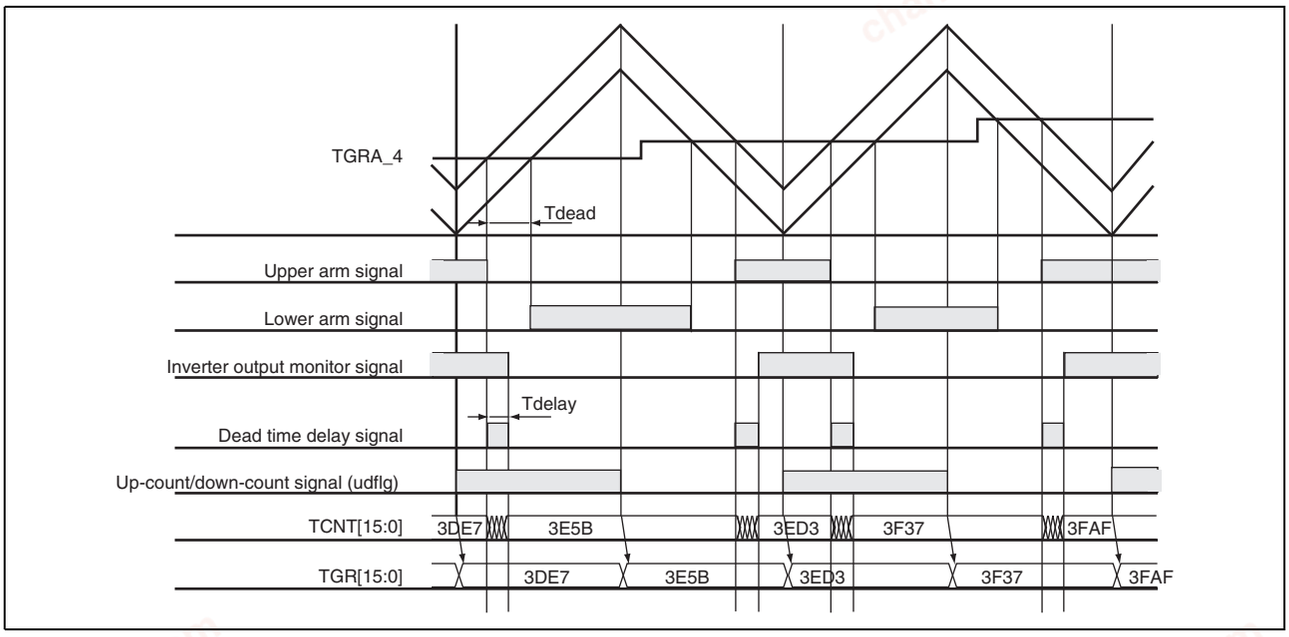


Figure 10.77 TCNT Capturing at Crest and/or Trough in Complementary PWM Operation

chanhvinh.com

10.5 Interrupt Sources

10.5.1 Interrupt Sources and Priorities

This module has three kinds of interrupt sources; TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own status flag and enable/disabled bit, allowing the generation of interrupt request signals to be enabled or disabled individually.

When an interrupt request is generated, the corresponding status flag in TSR is set to 1. If the corresponding enable/disable bit in TIER is set to 1 at this time, an interrupt is requested. The interrupt request is cleared by clearing the status flag to 0.

Relative channel priorities can be changed by the interrupt controller, however the priority order within a channel is fixed. For details, see section 7, Interrupt Controller.

Table 10.55 lists the interrupt sources of this module.

Table 10.55 Interrupts of Multi-Function Timer Pulse Unit 2

Channel	Name	Interrupt Source	Interrupt Flag	Activation of Direct Memory Access Controller	Priority
0	TGIA_0	TGRA_0 input capture/compare match	TGFA_0	Possible	High
	TGIB_0	TGRB_0 input capture/compare match	TGFB_0	Not possible	
	TGIC_0	TGRC_0 input capture/compare match	TGFC_0	Not possible	
	TGID_0	TGRD_0 input capture/compare match	TGFD_0	Not possible	
	TCIV_0	TCNT_0 overflow	TCFV_0	Not possible	
	TGIE_0	TGRE_0 compare match	TGFE_0	Not possible	
	TGIF_0	TGRF_0 compare match	TGFF_0	Not possible	
1	TGIA_1	TGRA_1 input capture/compare match	TGFA_1	Possible	
	TGIB_1	TGRB_1 input capture/compare match	TGFB_1	Not possible	
	TCIV_1	TCNT_1 overflow	TCFV_1	Not possible	
	TCIU_1	TCNT_1 underflow	TCFU_1	Not possible	
2	TGIA_2	TGRA_2 input capture/compare match	TGFA_2	Possible	
	TGIB_2	TGRB_2 input capture/compare match	TGFB_2	Not possible	
	TCIV_2	TCNT_2 overflow	TCFV_2	Not possible	
	TCIU_2	TCNT_2 underflow	TCFU_2	Not possible	
3	TGIA_3	TGRA_3 input capture/compare match	TGFA_3	Possible	
	TGIB_3	TGRB_3 input capture/compare match	TGFB_3	Not possible	
	TGIC_3	TGRC_3 input capture/compare match	TGFC_3	Not possible	
	TGID_3	TGRD_3 input capture/compare match	TGFD_3	Not possible	
	TCIV_3	TCNT_3 overflow	TCFV_3	Not possible	
4	TGIA_4	TGRA_4 input capture/compare match	TGFA_4	Possible	
	TGIB_4	TGRB_4 input capture/compare match	TGFB_4	Not possible	
	TGIC_4	TGRC_4 input capture/compare match	TGFC_4	Not possible	
	TGID_4	TGRD_4 input capture/compare match	TGFD_4	Not possible	
	TCIV_4	TCNT_4 overflow/underflow	TCFV_4	Not possible	Low

Note: This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

chanhvinh.com

(1) Input Capture/Compare Match Interrupt

An interrupt is requested if the TGIE bit in TIER is set to 1 when the TGF flag in TSR is set to 1 by the occurrence of a TGR input capture/compare match on a particular channel. The interrupt request is cleared by clearing the TGF flag to 0. This module has eighteen input capture/compare match interrupts, six for channel 0, four each for channels 3 and 4, and two each for channels 1 and 2. The TGFE_0 and TGFF_0 flags in channel 0 are not set by the occurrence of an input capture.

(2) Overflow Interrupt

An interrupt is requested if the TCIEV bit in TIER is set to 1 when the TCFV flag in TSR is set to 1 by the occurrence of TCNT overflow on a channel. The interrupt request is cleared by clearing the TCFV flag to 0. This module has five overflow interrupts, one for each channel.

(3) Underflow Interrupt

An interrupt is requested if the TCIEU bit in TIER is set to 1 when the TCFU flag in TSR is set to 1 by the occurrence of TCNT underflow on a channel. The interrupt request is cleared by clearing the TCFU flag to 0. This module has two underflow interrupts, one each for channels 1 and 2.

10.5.2 Activation of Direct Memory Access Controller

The direct memory access controller can be activated by the TGRA input capture/compare match interrupt in each channel. For details, see section 9, Direct Memory Access Controller.

In this module, a total of five TGRA input capture/compare match interrupts can be used as direct memory access controller activation sources, one each for channels 0 to 4.

10.5.3 A/D Converter Activation

The A/D converter can be activated by one of the following three methods in this module. Table 10.56 shows the relationship between interrupt sources and A/D converter start request signals.

(1) A/D Converter Activation by TGRA Input Capture/Compare Match or at TCNT_4 Trough in Complementary PWM Mode

The A/D converter can be activated by the occurrence of a TGRA input capture/compare match in each channel. In addition, if complementary PWM operation is performed while the TTGE2 bit in TIER_4 is set to 1, the A/D converter can be activated at the trough of TCNT_4 count (TCNT_4 = H'0000).

A/D converter start request signal TRGAN is issued to the A/D converter under either one of the following conditions.

- When the TGFA flag in TSR is set to 1 by the occurrence of a TGRA input capture/compare match on a particular channel while the TTGE bit in TIER is set to 1
- When the TCNT_4 count reaches the trough (TCNT_4 = H'0000) during complementary PWM operation while the TTGE2 bit in TIER_4 is set to 1

When either condition is satisfied, if A/D converter start signal TRGAN from this module is selected as the trigger in the A/D converter, A/D conversion will start.

chanhvinh.com

(2) A/D Converter Activation by Compare Match between TCNT_0 and TGRE_0

The A/D converter can be activated by generating A/D converter start request signal TRG0N when a compare match occurs between TCNT_0 and TGRE_0 in channel 0.

When the TGFE flag in TSR2_0 is set to 1 by the occurrence of a compare match between TCNT_0 and TGRE_0 in channel 0 while the TTGE2 bit in TIER2_0 is set to 1, A/D converter start request TGR0N is issued to the A/D converter. If A/D converter start signal TGR0N from this module is selected as the trigger in the A/D converter, A/D conversion will start.

(3) A/D Converter Activation by A/D Converter Start Request Delaying Function

The A/D converter can be activated by generating A/D converter start request signal TRG4AN or TRG4BN when the TCNT_4 count matches the TADCORA or TADCORB value if the UT4AE, DT4AE, UT4BE, or DT4BE bit in the A/D converter start request control register (TADCR) is set to 1. For details, refer to section 10.4.9, A/D Converter Start Request Delaying Function.

A/D conversion will start if A/D converter start signal TRG4AN from this module is selected as the trigger in the A/D converter when TRG4AN is generated or if TRG4BN from this module is selected as the trigger in the A/D converter when TRG4BN is generated.

Table 10.56 Interrupt Sources and A/D Converter Start Request Signals

Target Registers	Interrupt Source	A/D Converter Start Request Signal
TGRA_0 and TCNT_0	Input capture/compare match	TRGAN
TGRA_1 and TCNT_1		
TGRA_2 and TCNT_2		
TGRA_3 and TCNT_3		
TGRA_4 and TCNT_4		
TCNT_4	TCNT_4 Trough in complementary PWM mode	
TGRE_0 and TCNT_0	Compare match	TRG0N
TADCORA and TCNT_4		TRG4AN
TADCORB and TCNT_4		TRG4BN

chanhvinh.com

10.6 Operation Timing

10.6.1 Input/Output Timing

(1) TCNT Count Timing

Figure 10.78 shows TCNT count timing in internal clock operation, and Figure 10.79 shows TCNT count timing in external clock operation (normal mode), and Figure 10.80 shows TCNT count timing in external clock operation (phase counting mode).

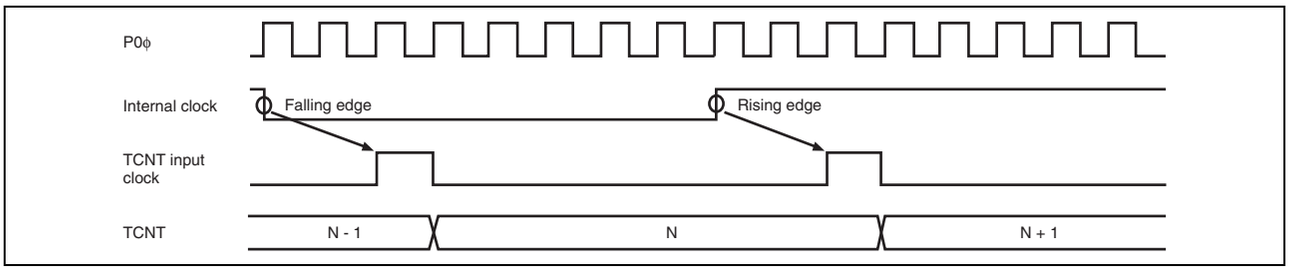


Figure 10.78 Count Timing in Internal Clock Operation

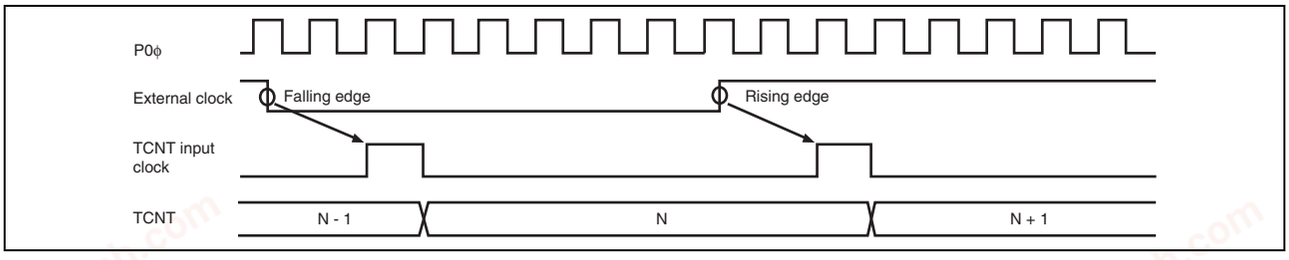


Figure 10.79 Count Timing in External Clock Operation

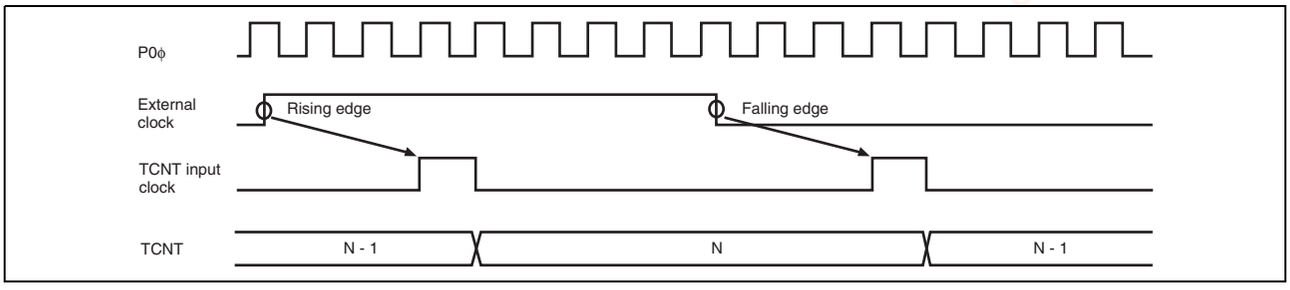


Figure 10.80 Count Timing in External Clock Operation (Phase Counting Mode)

chanhvinh.com

(2) Output Compare Output Timing

A compare match signal is generated in the final state in which TCNT and TGR match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the output value set in TIOR is output at the output compare output pin (TIOC pin). After a match between TCNT and TGR, the compare match signal is not generated until the TCNT input clock is generated.

Figure 10.81 shows output compare output timing (normal mode and PWM mode) and Figure 10.82 shows output compare output timing (complementary PWM mode and reset synchronous PWM mode).

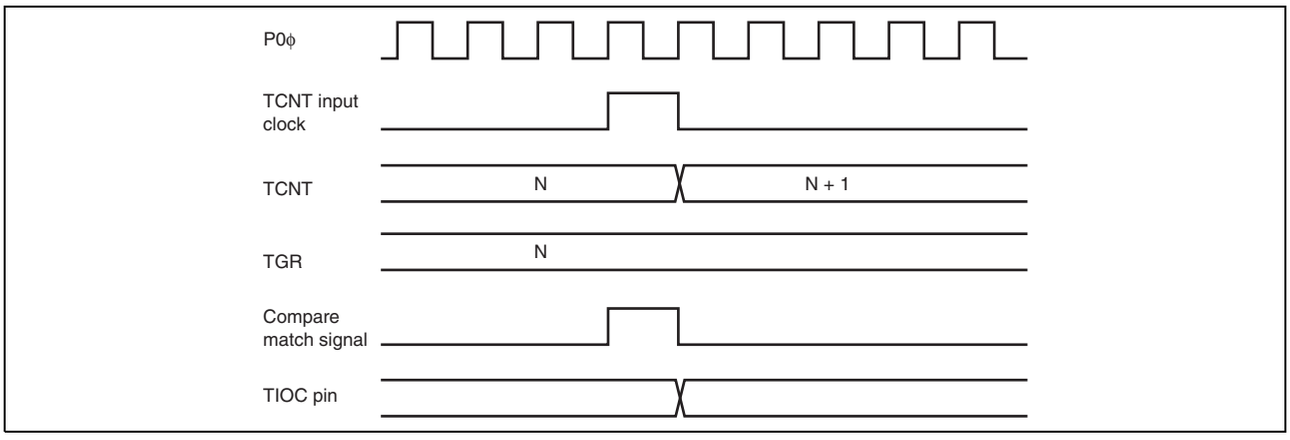


Figure 10.81 Output Compare Output Timing (Normal Mode/PWM Mode)

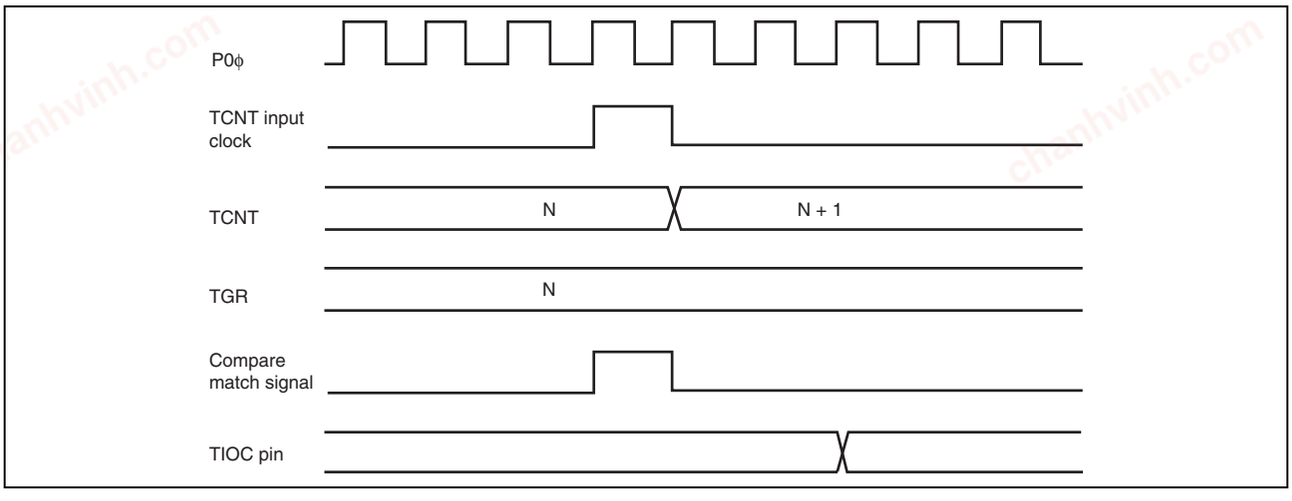


Figure 10.82 Output Compare Output Timing (Complementary PWM Mode/Reset Synchronous PWM Mode)

chanhvinh.com

(3) Input Capture Signal Timing

Figure 10.83 shows input capture signal timing.

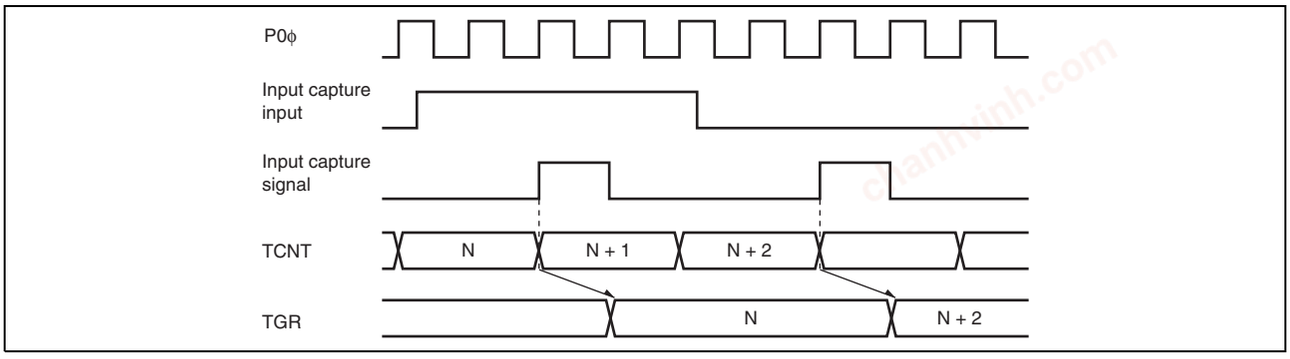


Figure 10.83 Input Capture Input Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

Figure 10.84 shows the timing when counter clearing on compare match is specified, and Figure 10.85 shows the timing when counter clearing on input capture is specified.

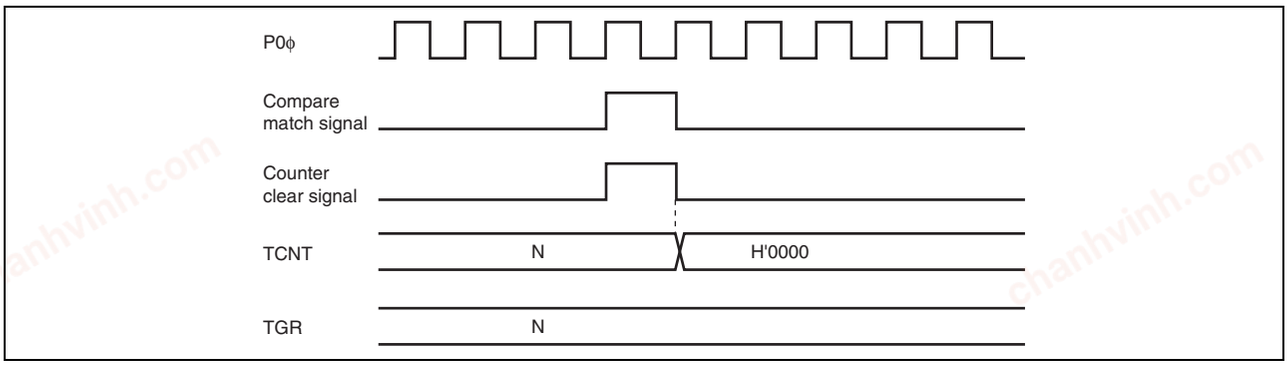


Figure 10.84 Counter Clear Timing (Compare Match)

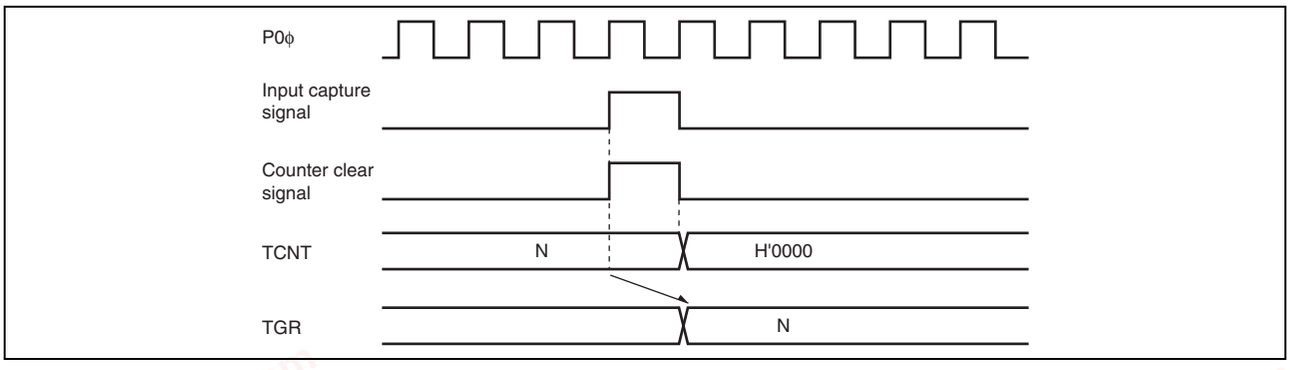


Figure 10.85 Counter Clear Timing (Input Capture)

chanhvinh.com

(5) Buffer Operation Timing

Figure 10.86 to Figure 10.88 show the timing in buffer operation.

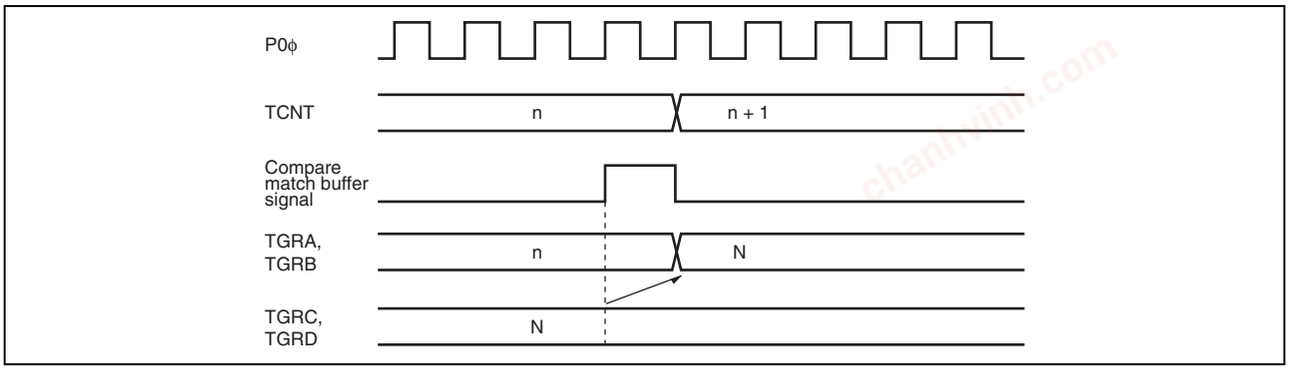


Figure 10.86 Buffer Operation Timing (Compare Match)

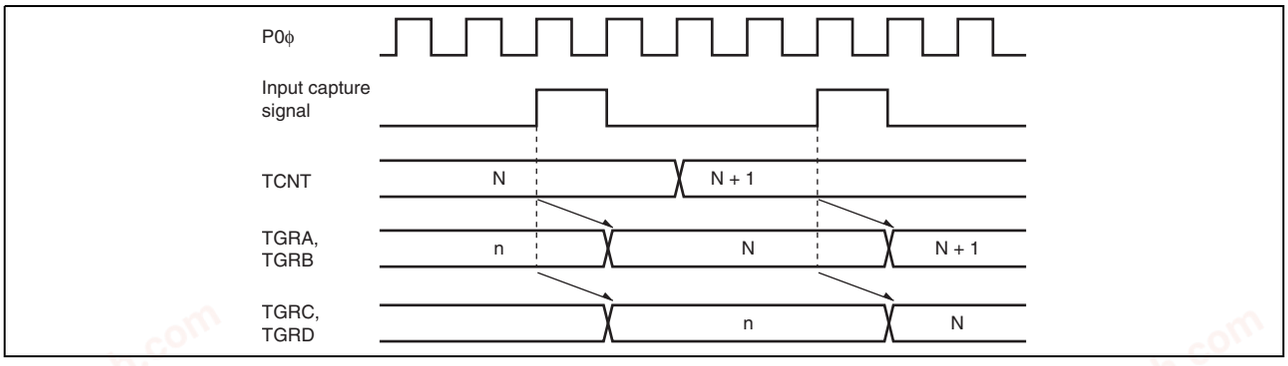


Figure 10.87 Buffer Operation Timing (Input Capture)

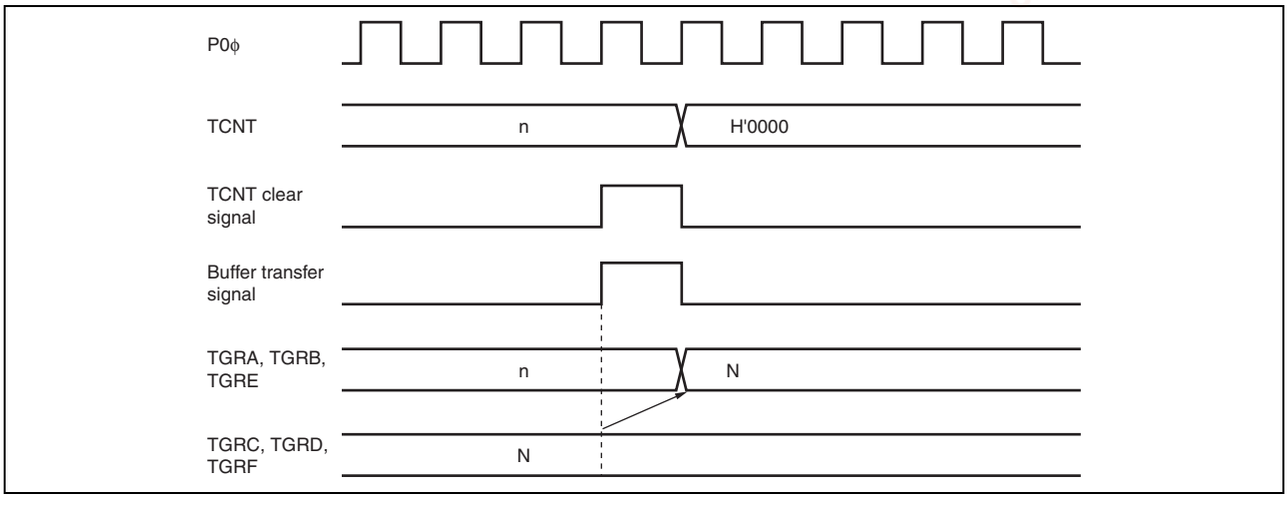


Figure 10.88 Buffer Transfer Timing (when TCNT Cleared)

chanhvinh.com

(6) Buffer Transfer Timing (Complementary PWM Mode)

Figure 10.89 to Figure 10.91 show the buffer transfer timing in complementary PWM mode.

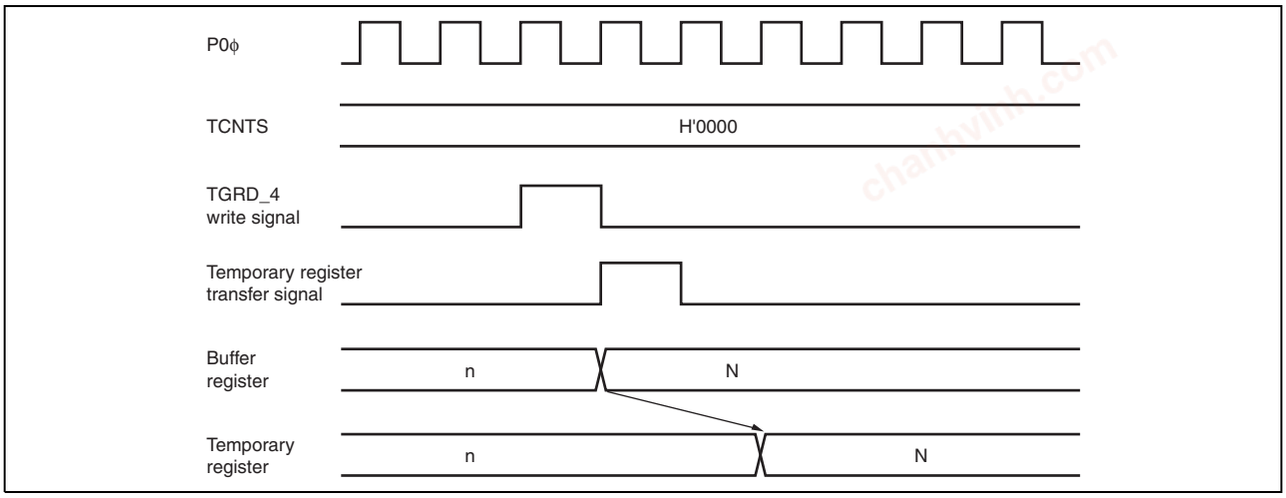


Figure 10.89 Transfer Timing from Buffer Register to Temporary Register (TCNTS Stop)

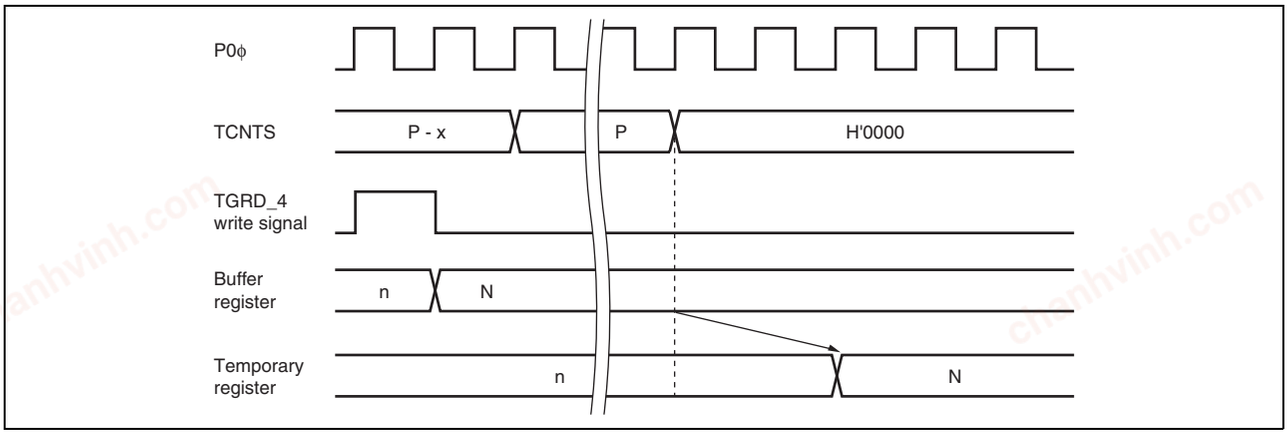


Figure 10.90 Transfer Timing from Buffer Register to Temporary Register (TCNTS Operating)

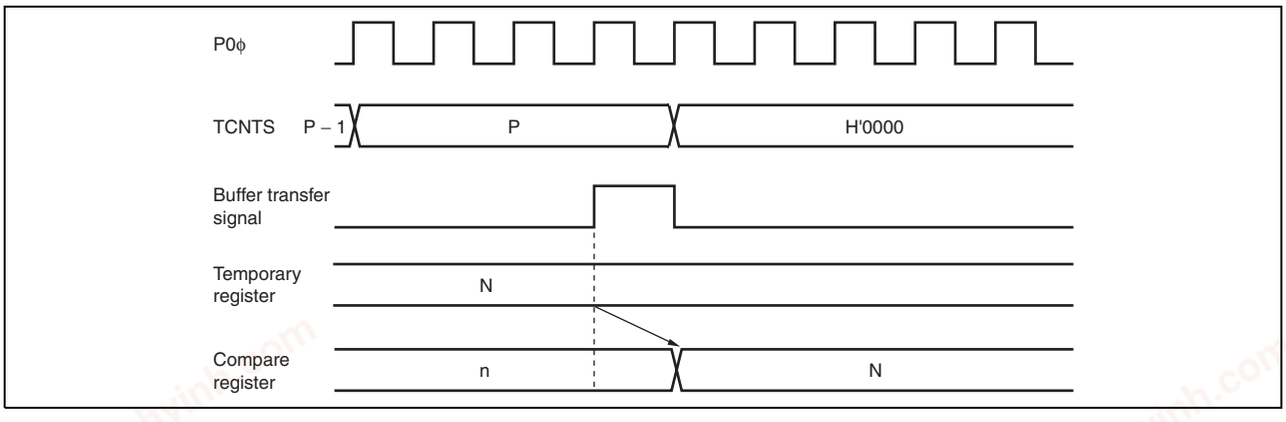


Figure 10.91 Transfer Timing from Temporary Register to Compare Register

chanhvinh.com

10.6.2 Interrupt Signal Timing

(1) TGF Flag Setting Timing in Case of Compare Match

Figure 10.92 shows the timing for setting of the TGF flag in TSR on compare match, and TGI interrupt request signal timing.

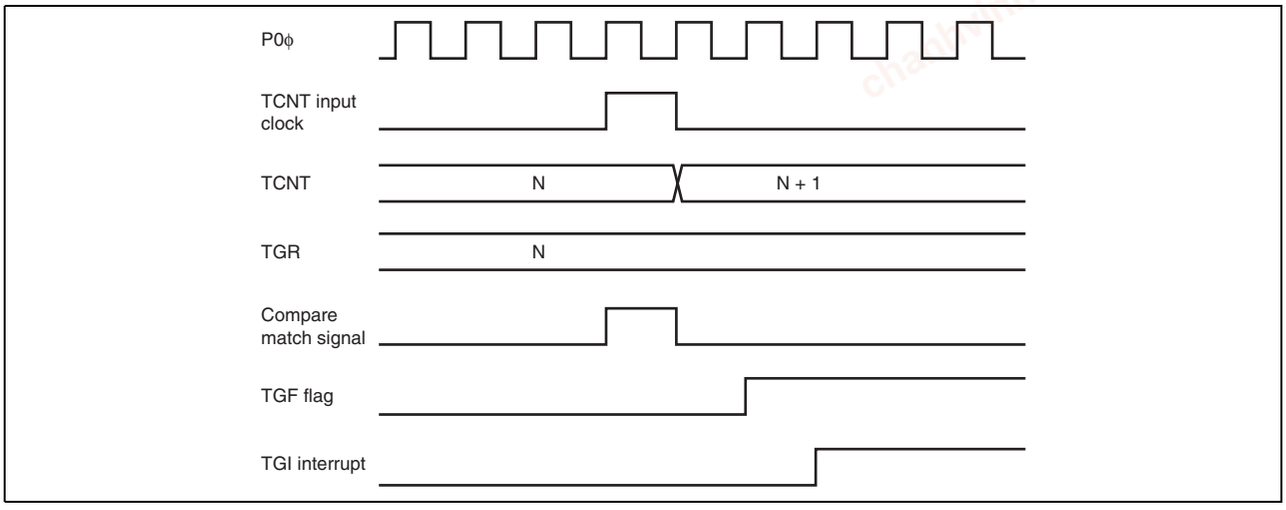


Figure 10.92 TGI Interrupt Timing (Compare Match)

(2) TGF Flag Setting Timing in Case of Input Capture

Figure 10.93 shows the timing for setting of the TGF flag in TSR on input capture, and TGI interrupt request signal timing.

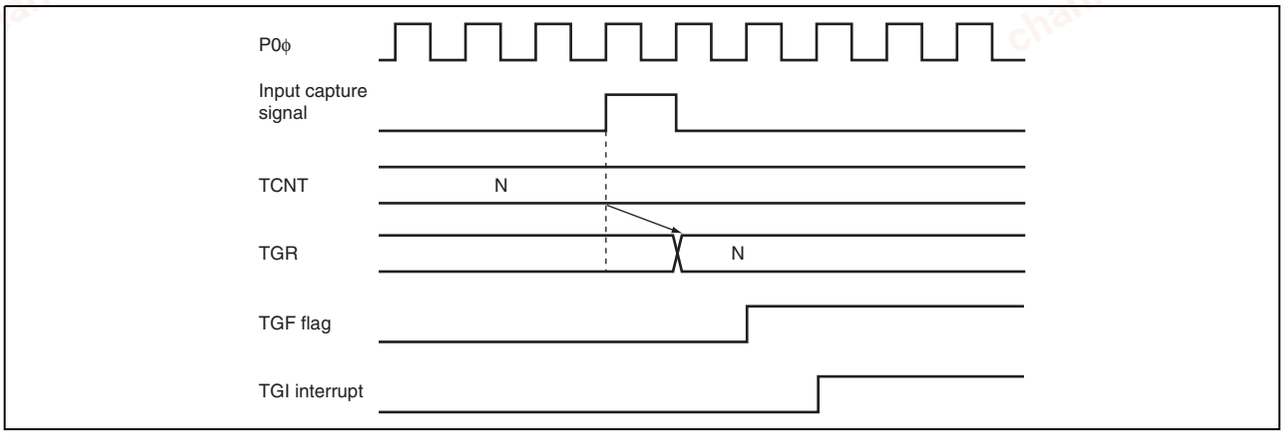


Figure 10.93 TGI Interrupt Timing (Input Capture)

chanhvinh.com

(3) TCFV Flag/TCFU Flag Setting Timing

Figure 10.94 shows the timing for setting of the TCFV flag in TSR on overflow, and TCIV interrupt request signal timing.

Figure 10.95 shows the timing for setting of the TCFU flag in TSR on underflow, and TCIU interrupt request signal timing.

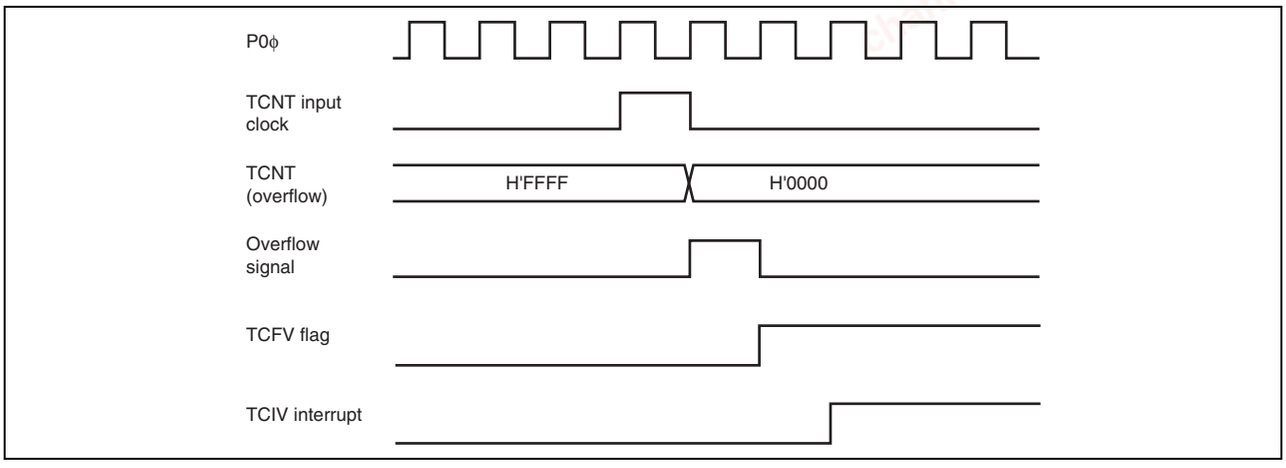


Figure 10.94 TCIV Interrupt Setting Timing

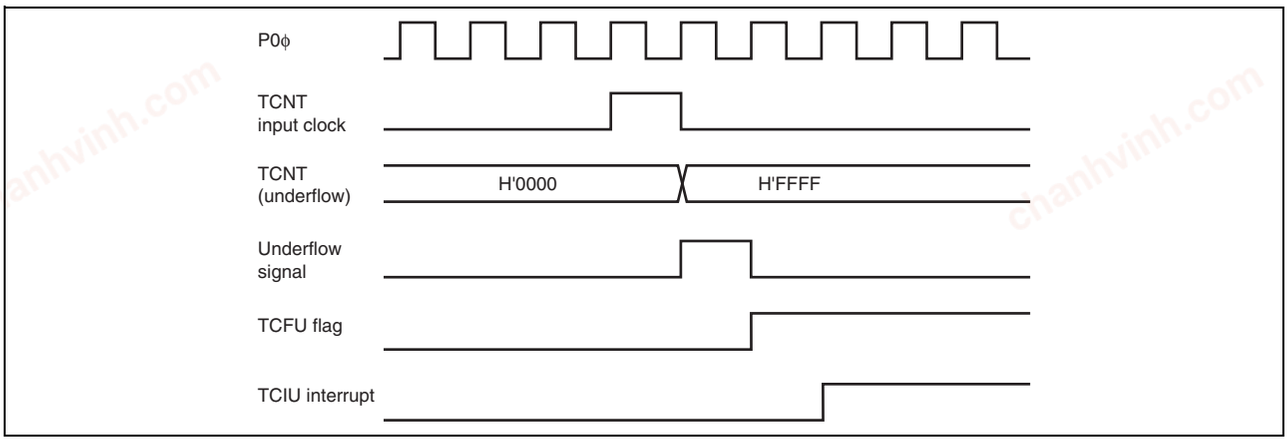


Figure 10.95 TCIU Interrupt Setting Timing

chanhvinh.com

(4) Status Flag Clearing Timing

After a status flag is read as 1 by the CPU, it is cleared by writing 0 to it. When the direct memory access controller is activated, the flag is cleared automatically. Figure 10.96 shows the timing for status flag clearing by the CPU, and Figure 10.97 shows the timing for status flag clearing by the direct memory access controller.

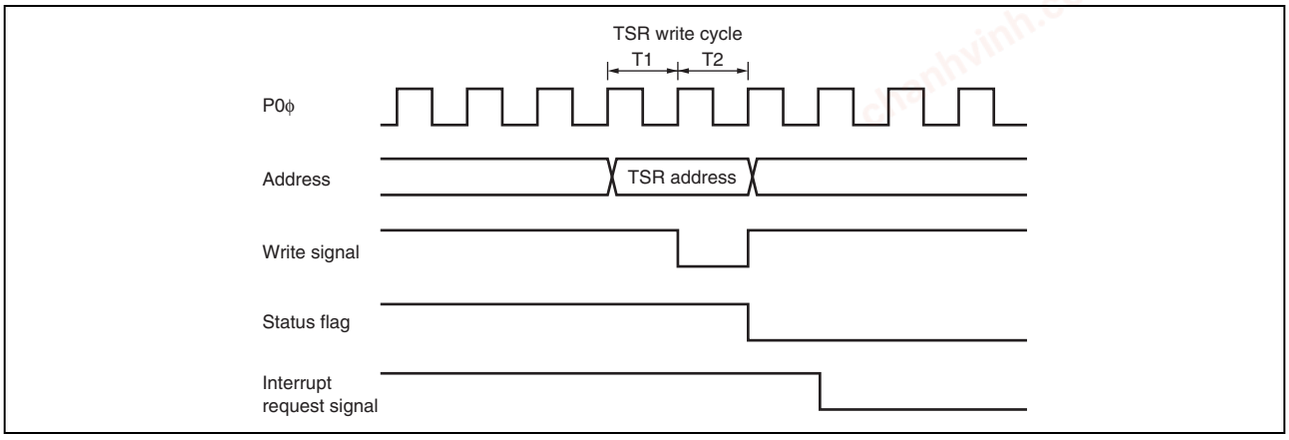


Figure 10.96 Timing for Status Flag Clearing by CPU

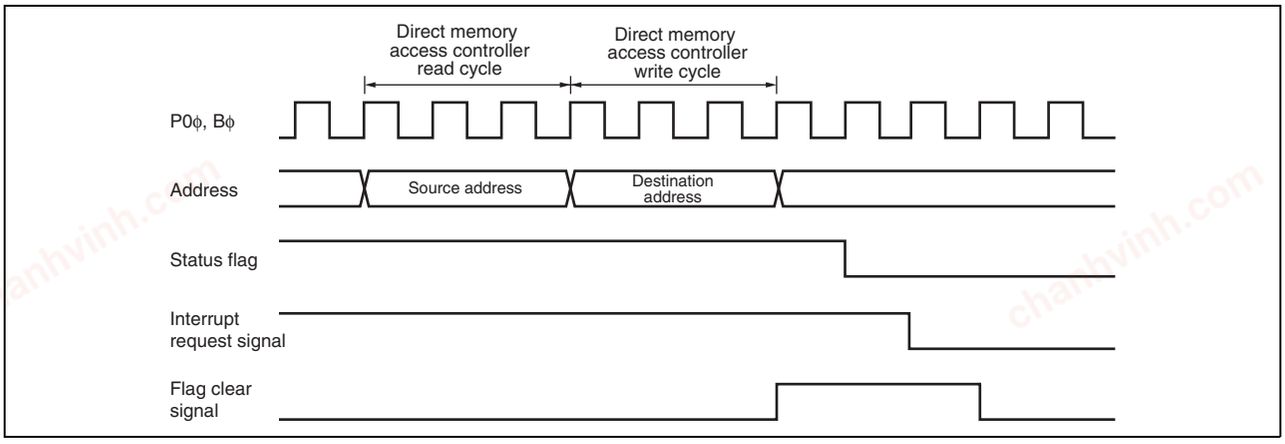


Figure 10.97 Timing for Status Flag Clearing by Direct Memory Access Controller Activation

chanhvinh.com

10.7 Usage Notes

10.7.1 Module Standby Mode Setting

Operation of this module can be disabled or enabled using the standby control register. The initial setting is for the operation to be halted. Register access is enabled by clearing module standby mode. For details, refer to section 42, Power-Down Modes.

10.7.2 Input Clock Restrictions

The input clock pulse width must be at least 1.5 states in the case of single-edge detection, and at least 2.5 states in the case of both-edge detection. This module will not operate properly at narrower pulse widths.

In phase counting mode, the phase difference and overlap between the two input clocks must be at least 1.5 states, and the pulse width must be at least 2.5 states. Figure 10.98 shows the input clock conditions in phase counting mode.

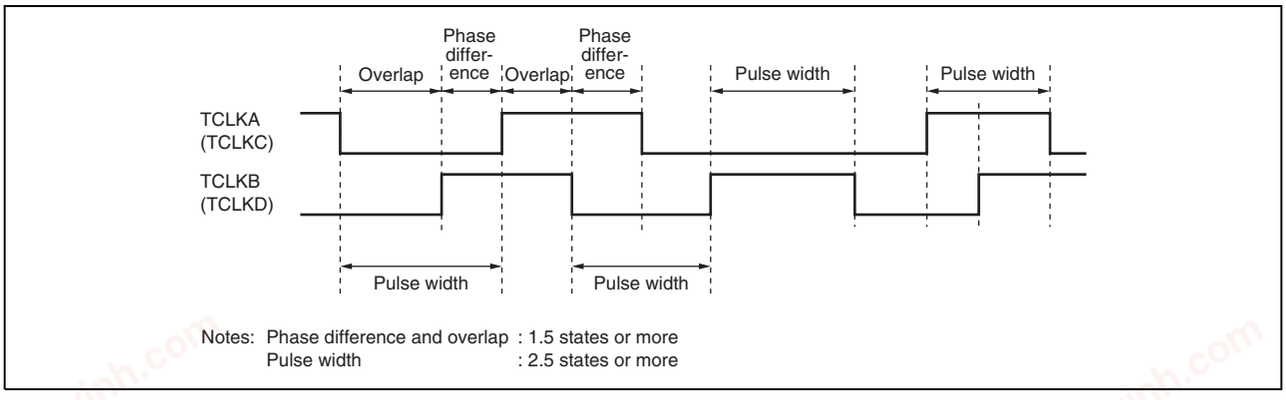


Figure 10.98 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

chanhvinh.com

10.7.3 Caution on Period Setting

When counter clearing on compare match is set, TCNT is cleared in the final state in which it matches the TGR value (the point at which the count value matched by TCNT is updated). Consequently, the actual counter frequency is given by the following formula:

$$f = \frac{P0\phi}{(N+1)}$$

Where f: Counter frequency
P0ϕ: Peripheral clock operating frequency
N: TGR set value

10.7.4 Contention between TCNT Write and Clear Operations

If the counter clear signal is generated in the T2 state of a TCNT write cycle, TCNT clearing takes precedence and the TCNT write is not performed.

Figure 10.99 shows the timing in this case.

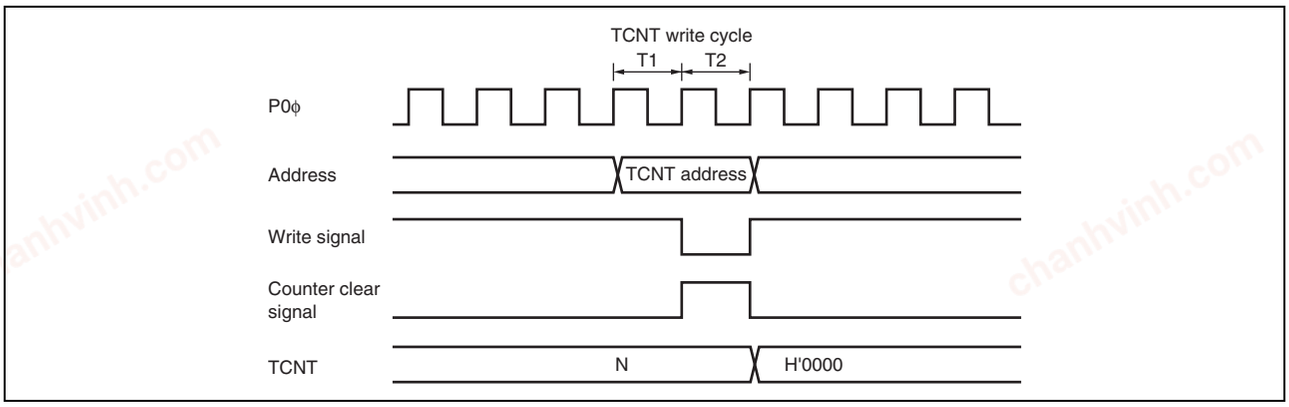


Figure 10.99 Contention between TCNT Write and Clear Operations

chanhvinh.com

10.7.5 Contention between TCNT Write and Increment Operations

If incrementing occurs in the T2 state of a TCNT write cycle, the TCNT write takes precedence and TCNT is not incremented.

Figure 10.100 shows the timing in this case.

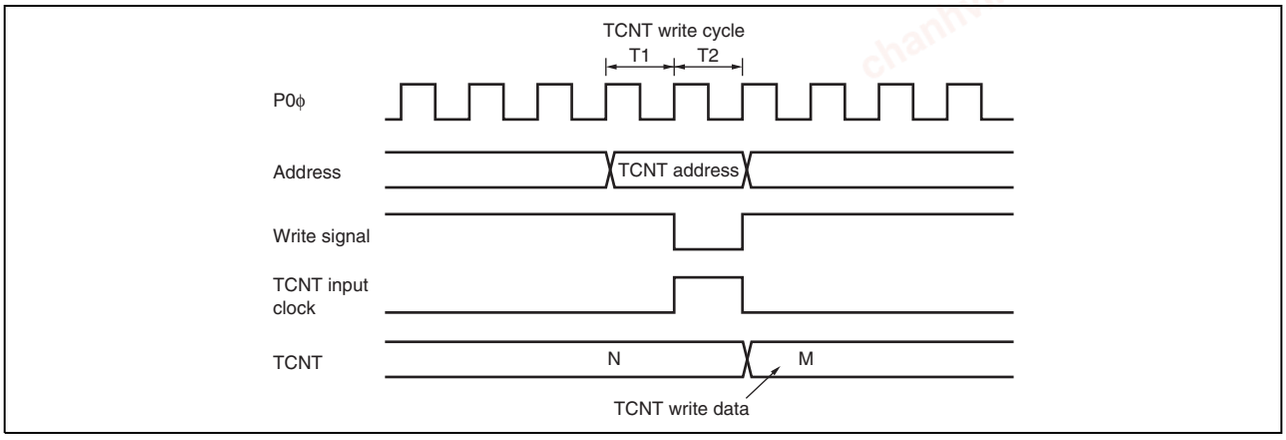


Figure 10.100 Contention between TCNT Write and Increment Operations

10.7.6 Contention between TGR Write and Compare Match

If a compare match occurs in the T2 state of a TGR write cycle, the TGR write is executed and the compare match signal is also generated.

Figure 10.101 shows the timing in this case.

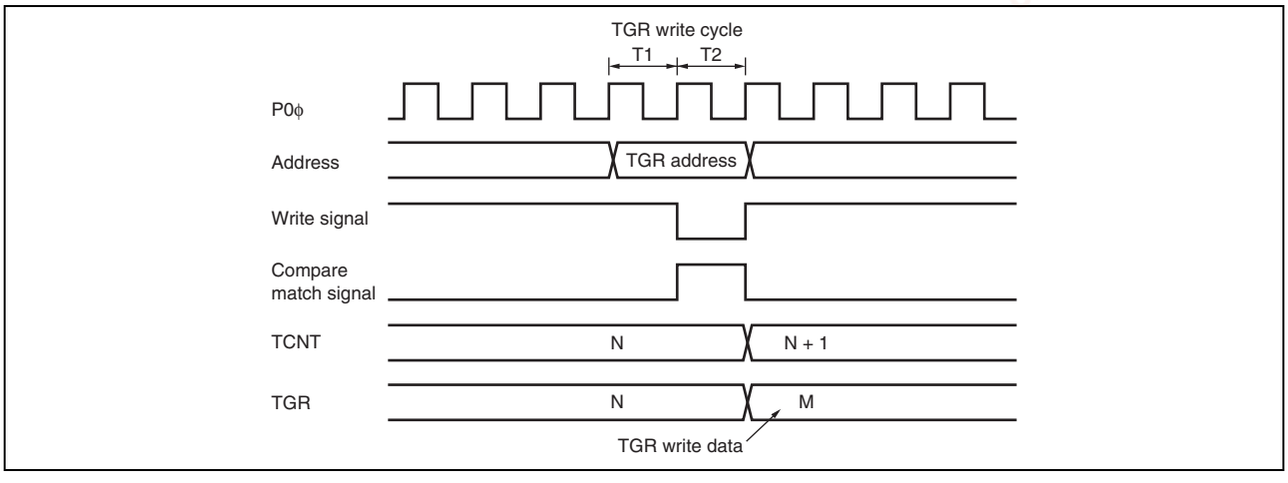


Figure 10.101 Contention between TGR Write and Compare Match

chanhvinh.com

10.7.7 Contention between Buffer Register Write and Compare Match

If a compare match occurs in the T2 state of a TGR write cycle, the data that is transferred to TGR by the buffer operation is the data after write.

Figure 10.102 shows the timing in this case.

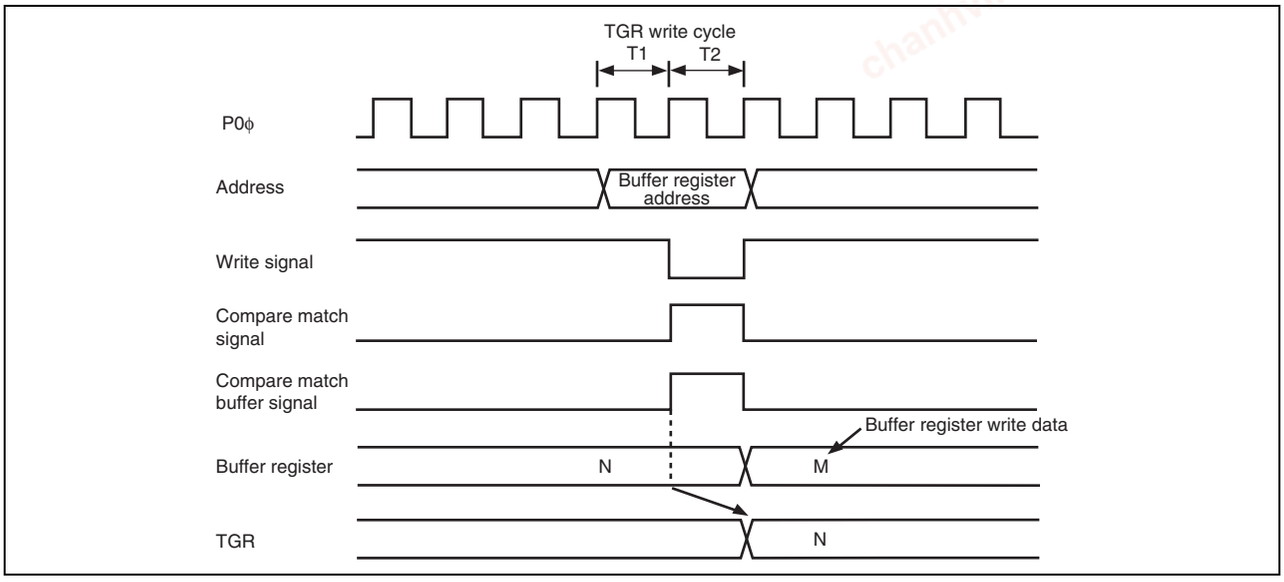


Figure 10.102 Contention between Buffer Register Write and Compare Match

chanhvinh.com

10.7.8 Contention between Buffer Register Write and TCNT Clear

When the buffer transfer timing is set at the TCNT clear by the buffer transfer mode register (TBTM), if TCNT clear occurs in the T2 state of a TGR write cycle, the data that is transferred to TGR by the buffer operation is the data before write.

Figure 10.103 shows the timing in this case.

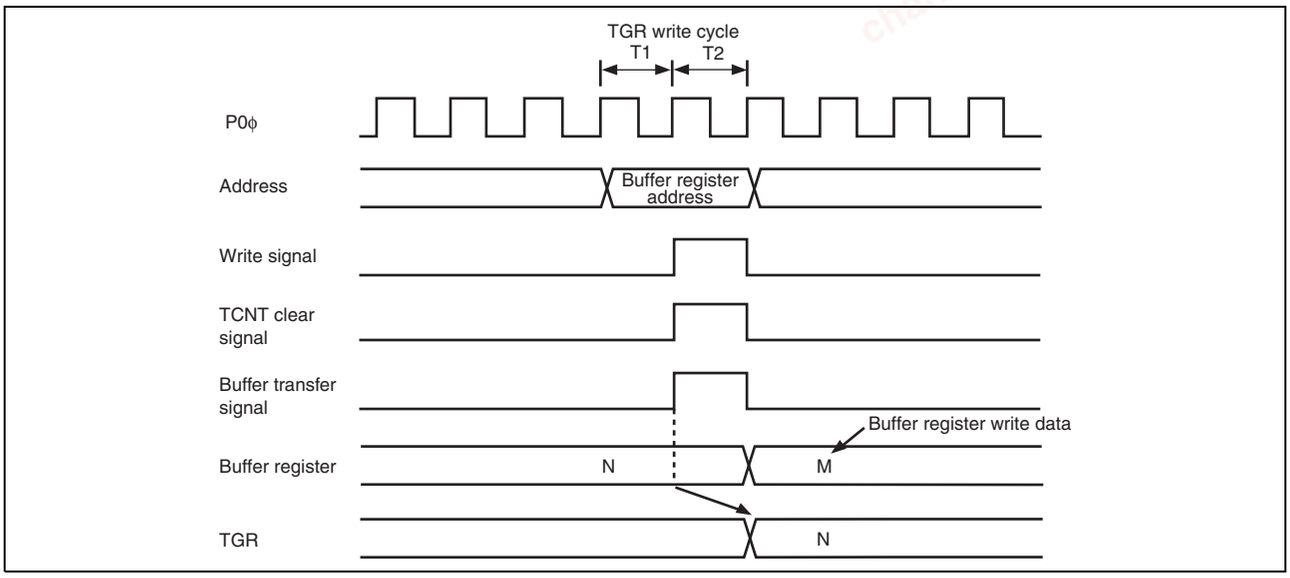


Figure 10.103 Contention between Buffer Register Write and TCNT Clear

10.7.9 Contention between TGR Read and Input Capture

If an input capture signal is generated in the T1 state of a TGR read cycle, the data that is read will be the data in the buffer before input capture transfer.

Figure 10.104 shows the timing in this case.

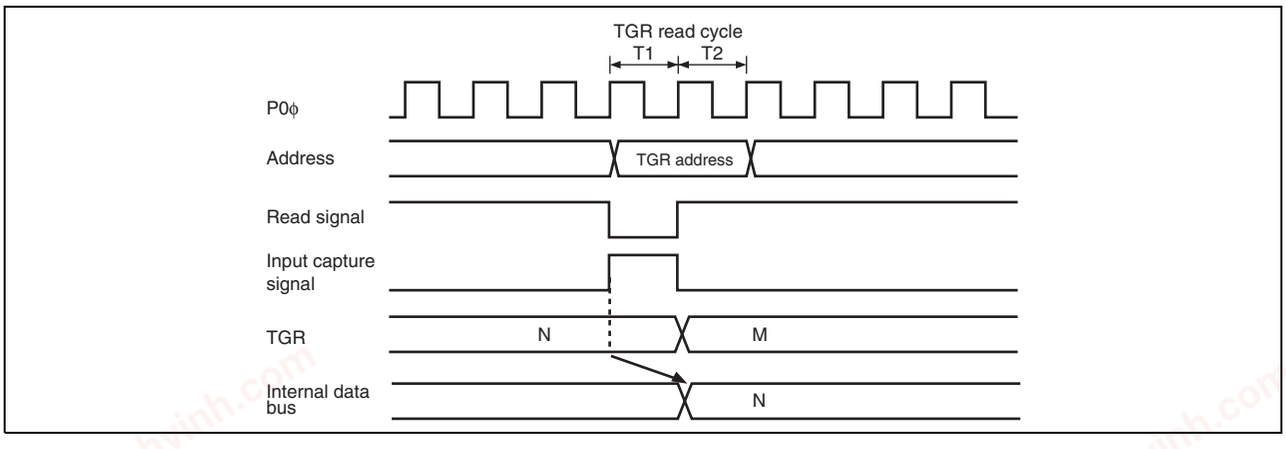


Figure 10.104 Contention between TGR Read and Input Capture

chanhvinh.com

10.7.10 Contention between TGR Write and Input Capture

If an input capture signal is generated in the T2 state of a TGR write cycle, the input capture operation takes precedence and the write to TGR is not performed.

Figure 10.105 shows the timing in this case.

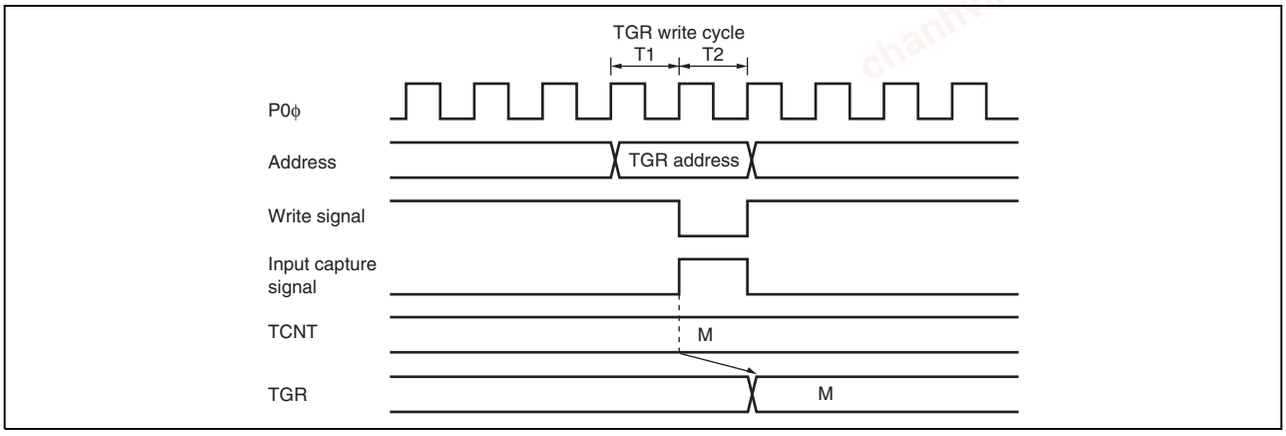


Figure 10.105 Contention between TGR Write and Input Capture

10.7.11 Contention between Buffer Register Write and Input Capture

If an input capture signal is generated in the T2 state of a buffer register write cycle, the buffer operation takes precedence and the write to the buffer register is not performed.

Figure 10.106 shows the timing in this case.

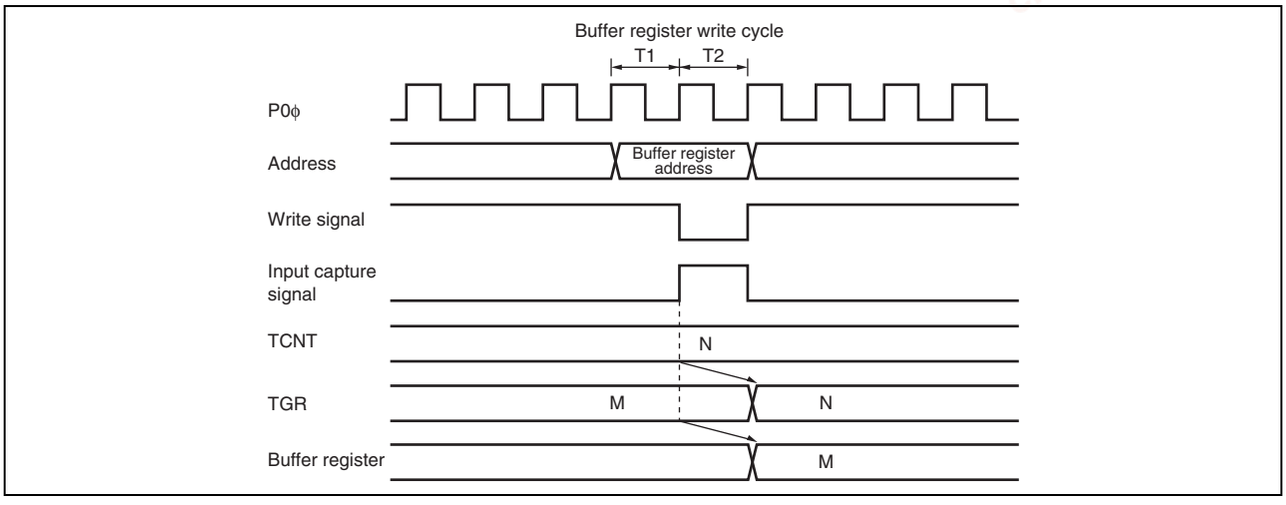


Figure 10.106 Contention between Buffer Register Write and Input Capture

chanhvinh.com

10.7.12 TCNT_2 Write and Overflow/Underflow Contention in Cascade Connection

With timer counters TCNT_1 and TCNT_2 in a cascade connection, when a contention occurs during TCNT_1 count (during a TCNT_2 overflow/underflow) in the T_2 state of the TCNT_2 write cycle, the write to TCNT_2 is conducted, and the TCNT_1 count signal is disabled. At this point, if there is match with TGRA_1 and the TCNT_1 value, a compare signal is issued. Furthermore, when the TCNT_1 count clock is selected as the input capture source of channel 0, TGRA_0 to D_0 carry out the input capture operation. In addition, when the compare match/input capture is selected as the input capture source of TGRB_1, TGRB_1 carries out input capture operation. The timing is shown in Figure 10.107.

For cascade connections, be sure to synchronize settings for channels 1 and 2 when setting TCNT clearing.

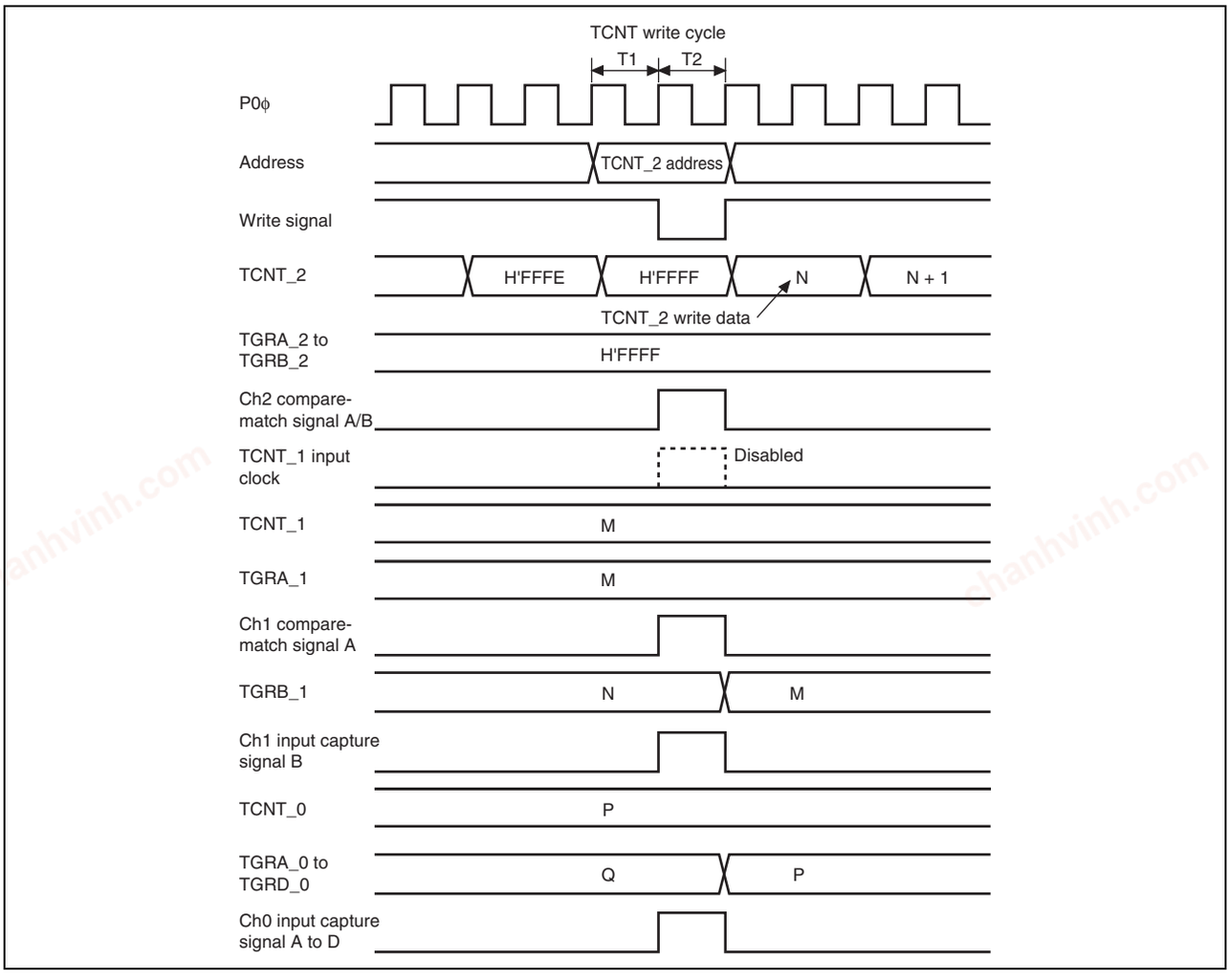


Figure 10.107 TCNT_2 Write and Overflow/Underflow Contention with Cascade Connection

chanhvinh.com

10.7.13 Counter Value during Complementary PWM Mode Stop

When counting operation is suspended with TCNT_3 and TCNT_4 in complementary PWM mode, TCNT_3 has the timer dead time register (TDDR) value, and TCNT_4 is held at H'0000.

When restarting complementary PWM mode, counting begins automatically from the initialized state. This explanatory diagram is shown in Figure 10.108.

When counting begins in another operating mode, be sure that TCNT_3 and TCNT_4 are set to the initial values.

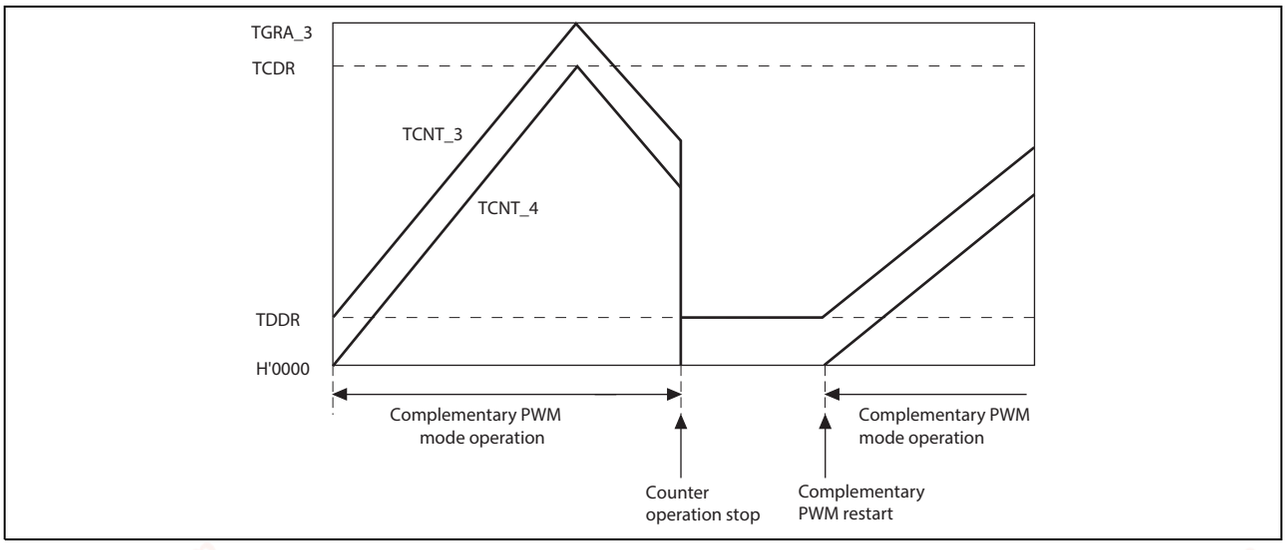


Figure 10.108 Counter Value during Complementary PWM Mode Stop

10.7.14 Buffer Operation Setting in Complementary PWM Mode

In complementary PWM mode, conduct rewrites by buffer operation for the PWM cycle setting register (TGRA_3), timer cycle data register (TCDR), and duty setting registers (TGRB_3, TGRA_4, and TGRB_4).

In complementary PWM mode, channel 3 and channel 4 buffers operate in accordance with bit settings BFA and BFB of TMDR_3. When TMDR_3's BFA bit is set to 1, TGRC_3 functions as a buffer register for TGRA_3. At the same time, TGRC_4 functions as the buffer register for TGRA_4, and TCBR functions as the TCDR's buffer register.

chanhvinh.com

10.7.15 Reset Sync PWM Mode Buffer Operation and Compare Match Flag

When setting buffer operation for reset sync PWM mode, set the BFA and BFB bits of TMDR_4 to 0. The TIOC4C pin will be unable to produce its waveform output if the BFA bit of TMDR_4 is set to 1.

In reset sync PWM mode, the channel 3 and channel 4 buffers operate in accordance with the BFA and BFB bit settings of TMDR_3. For example, if the BFA bit of TMDR_3 is set to 1, TGRC_3 functions as the buffer register for TGRA_3. At the same time, TGRC_4 functions as the buffer register for TGRA_4.

The TGFC bit and TGFD bit of TSR_3 and TSR_4 are not set when TGRC_3 and TGRD_3 are operating as buffer registers.

Figure 10.109 shows an example of operations for TGR_3, TGR_4, TIOC3, and TIOC4, with TMDR_3's BFA and BFB bits set to 1, and TMDR_4's BFA and BFB bits set to 0.

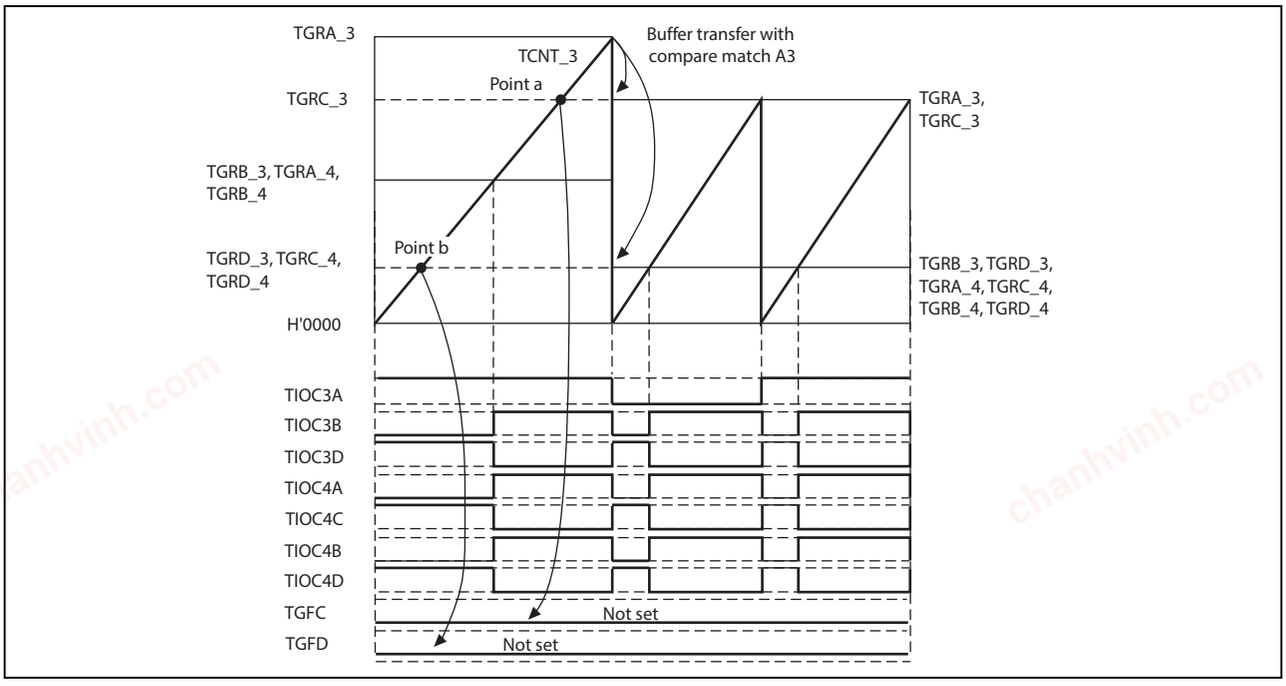


Figure 10.109 Buffer Operation and Compare-Match Flags in Reset Synchronous PWM Mode

chanhvinh.com

10.7.16 Overflow Flags in Reset Synchronous PWM Mode

When set to reset synchronous PWM mode, TCNT_3 and TCNT_4 start counting when the CST3 bit of TSTR is set to 1. At this point, TCNT_4's count clock source and count edge obey the TCR_3 setting.

In reset synchronous PWM mode, with cycle register TGRA_3's set value at H'FFFF, when specifying TGR3A compare-match for the counter clear source, TCNT_3 and TCNT_4 count up to H'FFFF, then a compare-match occurs with TGRA_3, and TCNT_3 and TCNT_4 are both cleared. At this point, TSR's overflow flag TCFV bit is not set.

Figure 10.110 shows a TCFV bit operation example in reset synchronous PWM mode with a set value for cycle register TGRA_3 of H'FFFF, when a TGRA_3 compare-match has been specified without synchronous setting for the counter clear source.

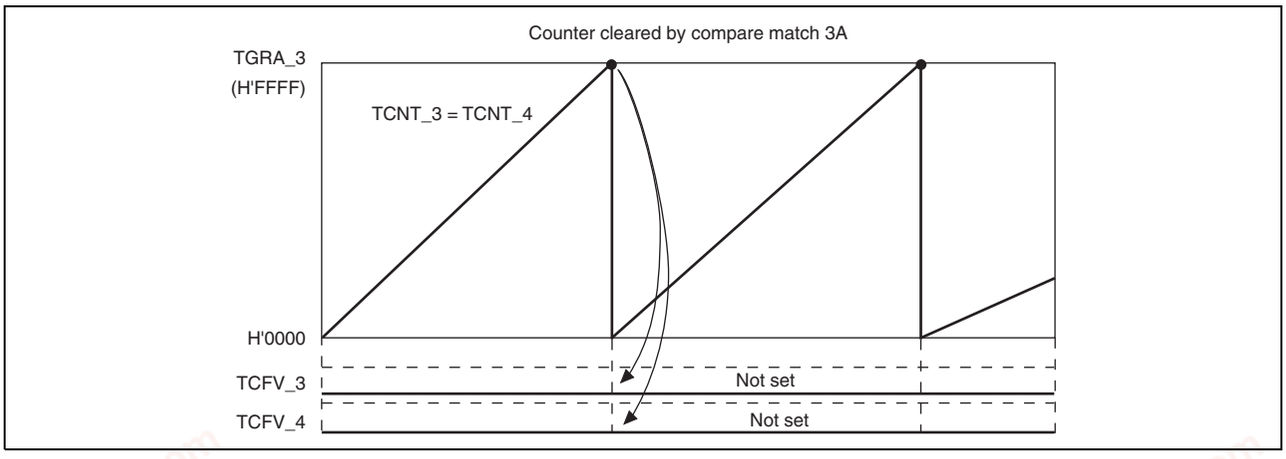


Figure 10.110 Reset Synchronous PWM Mode Overflow Flag

10.7.17 Contention between Overflow/Underflow and Counter Clearing

If overflow/underflow and counter clearing occur simultaneously, the TCFV/TCFU flag in TSR is not set and TCNT clearing takes precedence.

Figure 10.111 shows the operation timing when a TGR compare match is specified as the clearing source, and when H'FFFF is set in TGR.

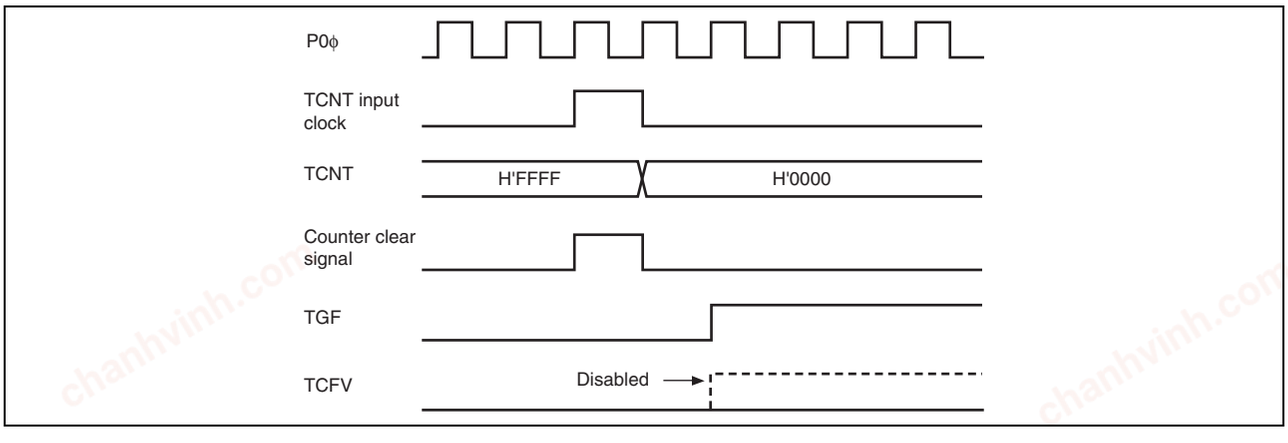


Figure 10.111 Contention between Overflow and Counter Clearing

chanhvinh.com

10.7.18 Contention between TCNT Write and Overflow/Underflow

If there is an up-count or down-count in the T2 state of a TCNT write cycle, and overflow/underflow occurs, the TCNT write takes precedence and the TCFV/TCFU flag in TSR is not set.

Figure 10.112 shows the operation timing when there is contention between TCNT write and overflow.

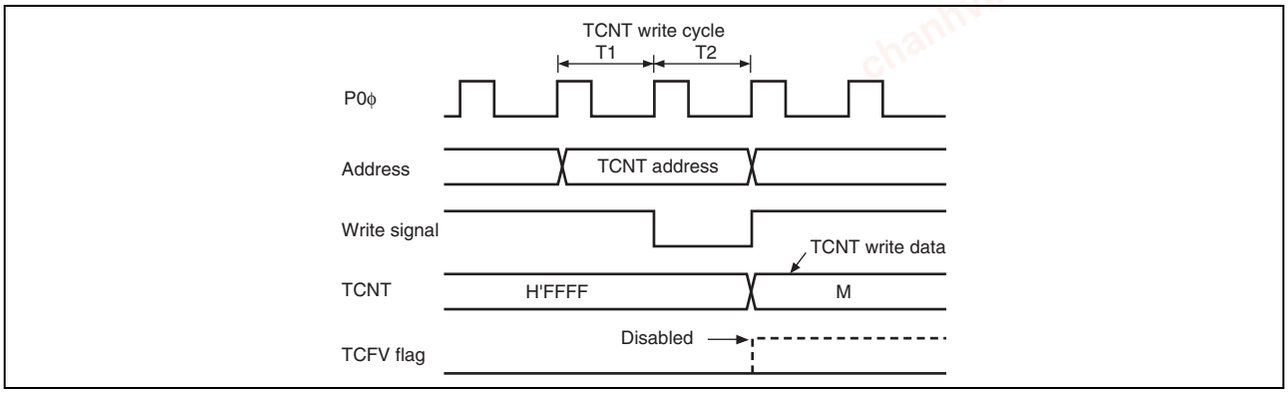


Figure 10.112 Contention between TCNT Write and Overflow

10.7.19 Cautions on Transition from Normal Operation or PWM Mode 1 to Reset-Synchronized PWM Mode

When making a transition from channel 3 or 4 normal operation or PWM mode 1 to reset-synchronized PWM mode, if the counter is halted with the output pins (TIOC3B, TIOC3D, TIOC4A, TIOC4C, TIOC4B, TIOC4D) in the high-level state, followed by the transition to reset-synchronized PWM mode and operation in that mode, the initial pin output will not be correct.

When making a transition from normal operation to reset-synchronized PWM mode, write H'11 to registers TIORH_3, TIORL_3, TIORH_4, and TIORL_4 to initialize the output pins to low level output, then set an initial register value of H'00 before making the mode transition.

When making a transition from PWM mode 1 to reset-synchronized PWM mode, first switch to normal operation, then initialize the output pins to low level output and set an initial register value of H'00 before making the transition to reset-synchronized PWM mode.

10.7.20 Output Level in Complementary PWM Mode and Reset-Synchronized PWM Mode

When channels 3 and 4 are in complementary PWM mode or reset-synchronized PWM mode, the PWM waveform output level is set with the OLSP and OLSN bits in the timer output control register (TOCR). In the case of complementary PWM mode or reset-synchronized PWM mode, TIOR should be set to H'00.

10.7.21 Interrupts in Module Standby Mode

If module standby mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the direct memory access controller activation source. Interrupts should therefore be disabled before entering module standby mode.

chanhvinh.com

10.7.22 Simultaneous Capture of TCNT_1 and TCNT_2 in Cascade Connection

When timer counters 1 and 2 (TCNT_1 and TCNT_2) are operated as a 32-bit counter in cascade connection, the cascade counter value cannot be captured successfully even if input-capture input is simultaneously done to TIOC1A and TIOC2A or to TIOC1B and TIOC2B. This is because the input timing of TIOC1A and TIOC2A or of TIOC1B and TIOC2B may not be the same when external input-capture signals to be input into TCNT_1 and TCNT_2 are taken in synchronization with the internal clock. For example, TCNT_1 (the counter for upper 16 bits) does not capture the count-up value by overflow from TCNT_2 (the counter for lower 16 bits) but captures the count value before the count-up. In this case, the values of TCNT_1 = H'FFF1 and TCNT_2 = H'0000 should be transferred to TGRA_1 and TGRA_2 or to TGRB_1 and TGRB_2, but the values of TCNT_1 = H'FFF0 and TCNT_2 = H'0000 are erroneously transferred.

10.7.23 Notes on Output Waveform Control During Synchronous Counter Clearing in Complementary PWM Mode

In complementary PWM mode, when output waveform control during synchronous counter clearing is enabled (WRE in the TWCR register set to 1), the following problems may occur when condition (1) or condition (2), below, is satisfied.

- Dead time for the PWM output pins may be too short (or nonexistent).
- Active-level output from the PWM negative-phase pins may occur outside the correct active-level output interval

Condition (1): When synchronous clearing occurs in the PWM output dead time interval within initial output suppression interval (10) (Figure 10.113).

Condition (2): When synchronous clearing occurs within initial output suppression interval (10) or (11) and TGRB_3 ≤ TDDR, TGRA_4 ≤ TDDR, or TGRB_4 ≤ TDDR is true (Figure 10.114)

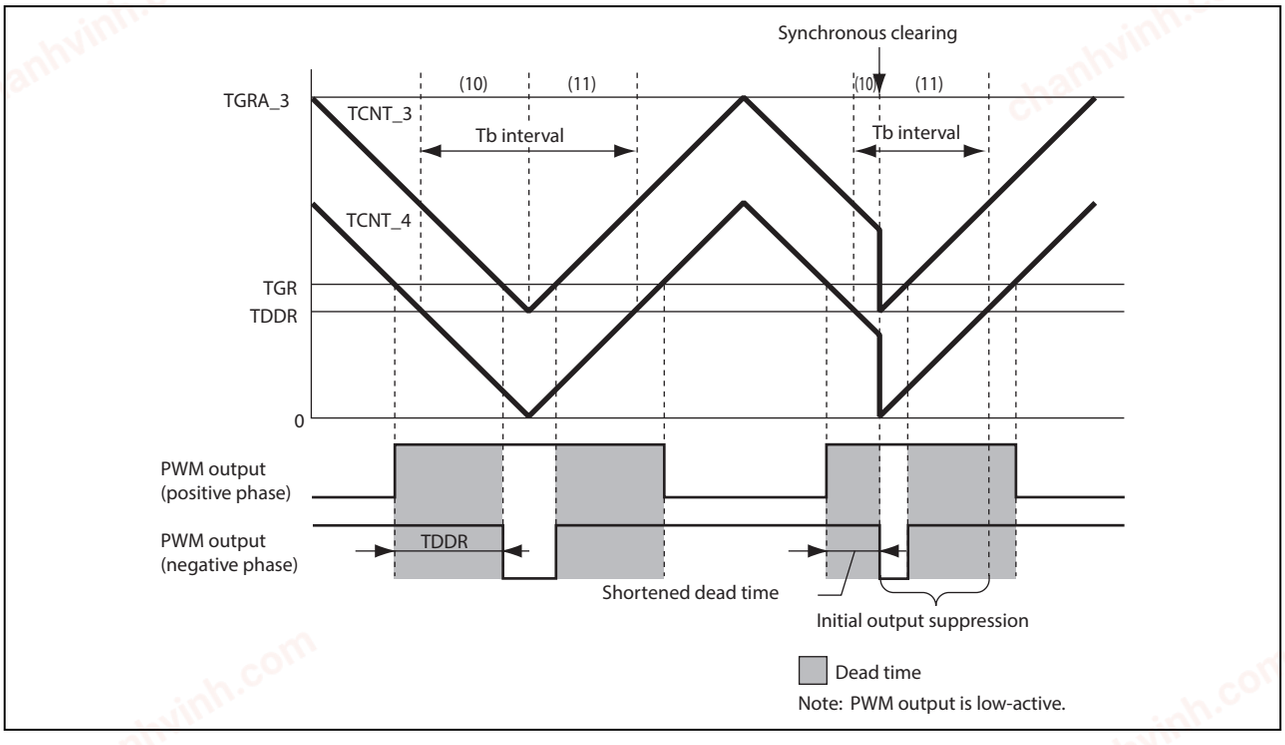


Figure 10.113 Condition (1) Synchronous Clearing Example

chanhvinh.com

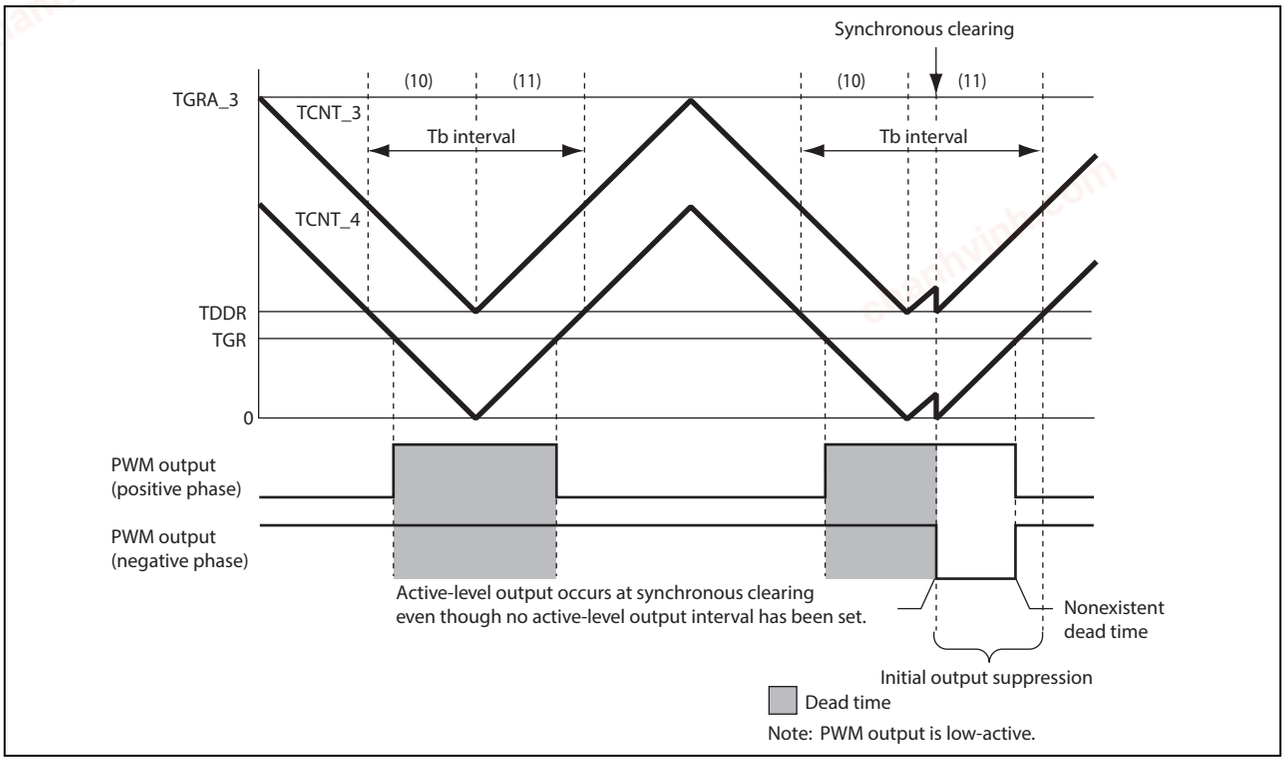


Figure 10.114 Condition (2) Synchronous Clearing Example

The following workaround can be used to avoid these problems.

When using synchronous clearing, make sure to set compare registers TGRB_3, TGRA_4, and TGRB_4 to a value twice or more the setting of dead time data register TDDR.

chanhvinh.com

10.8 Output Pin Initialization for Multi-Function Timer Pulse Unit 2

10.8.1 Operating Modes

This module has the following six operating modes. Waveform output is possible in all of these modes.

- Normal mode (channels 0 to 4)
- PWM mode 1 (channels 0 to 4)
- PWM mode 2 (channels 0 to 2)
- Phase counting modes 1 to 4 (channels 1 and 2)
- Complementary PWM mode (channels 3 and 4)
- Reset-synchronized PWM mode (channels 3 and 4)

The output pin initialization method for each of these modes is described in this section.

10.8.2 Reset Start Operation

The output pins of this module (TIOC*) are initialized low by a power-on reset or in deep standby mode. Since the pin functions are selected using the general I/O ports, when the general I/O port is set, the pin states at that point are output to the ports. When this module output is selected by the general I/O port immediately after a reset, the initial output level, low, is output directly at the port. When the active level is low, the system will operate at this point, and therefore the general
I/O port setting should be made after the initialization of the output pins is completed.

Note: Channel number and port notation are substituted for *.

chanhvinh.com

10.8.3 Operation in Case of Re-Setting Due to Error during Operation, etc.

If an error occurs during operation of this module, the module output should be cut by the system. Cutoff is performed by switching the pin output to port output with the general I/O port and outputting the inverse of the active level. The pin initialization procedures for re-setting due to an error during operation, etc., and the procedures for restarting in a different mode after re-setting, are shown below.

This module has six operating modes, as stated above. There are thus 36 mode transition combinations, but some transitions are not available with certain channel and mode combinations. Possible mode transition combinations are shown in Table 10.57.

Table 10.57 Mode Transition Combinations

	After					
Before	**Normal**	**PWM1**	**PWM2**	**PCM**	**CPWM**	**RPWM**
Normal	(1)	(2)	(3)	(4)	(5)	(6)
PWM1	(7)	(8)	(9)	(10)	(11)	(12)
PWM2	(13)	(14)	(15)	(16)	None	None
PCM	(17)	(18)	(19)	(20)	None	None
CPWM	(21)	(22)	None	None	(23) (24)	(25)
RPWM	(26)	(27)	None	None	(28)	(29)

[Legend]
Normal: Normal mode
PWM1: PWM mode 1
PWM2: PWM mode 2
PCM: Phase counting modes 1 to 4
CPWM: Complementary PWM mode
RPWM: Reset-synchronized PWM mode

chanhvinh.com

10.8.4 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation, etc.

- When making a transition to a mode (Normal, PWM1, PWM2, PCM) in which the pin output level is selected by the timer I/O control register (TIOR) setting, initialize the pins by means of a TIOR setting.
- In PWM mode 1, since a waveform is not output to the TIOC*B (TIOC *D) pin, setting TIOR will not initialize the pins. If initialization is required, carry it out in normal mode, then switch to PWM mode 1.
- In PWM mode 2, since a waveform is not output to the cycle register pin, setting TIOR will not initialize the pins. If initialization is required, carry it out in normal mode, then switch to PWM mode 2.
- In normal mode or PWM mode 2, if TGRC and TGRD operate as buffer registers, setting TIOR will not initialize the buffer register pins. If initialization is required, clear buffer mode, carry out initialization, then set buffer mode again.
- In PWM mode 1, if either TGRC or TGRD operates as a buffer register, setting TIOR will not initialize the TGRC pin. To initialize the TGRC pin, clear buffer mode, carry out initialization, then set buffer mode again.
- When making a transition to a mode (CPWM, RPWM) in which the pin output level is selected by the timer output control register (TOCR) setting, switch to normal mode and perform initialization with TIOR, then restore TIOR to its initial value, and temporarily disable channel 3 and 4 output with the timer output master enable register (TOER). Then operate the unit in accordance with the mode setting procedure (TOCR setting, TMDR setting, TOER setting).

Note: Channel number is substituted for * indicated in this article.

Pin initialization procedures are described below for the numbered combinations in Table 10.57. The active level is assumed to be low.

chanhvinh.com

(1) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Normal Mode

Figure 10.115 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in normal mode after re-setting.

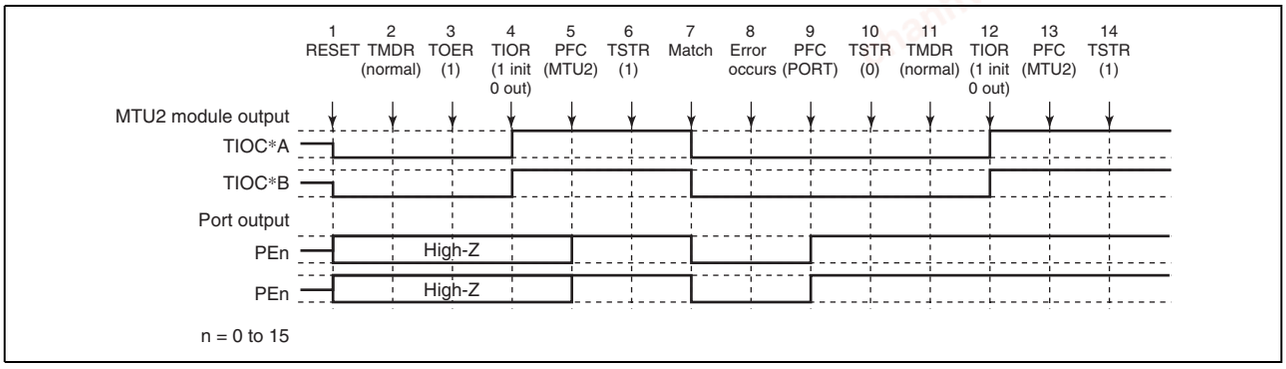


Figure 10.115 Error Occurrence in Normal Mode, Recovery in Normal Mode

1. After a reset, the module output is low and ports are in the high-impedance state.
2. After a reset, the TMDR setting is for normal mode.
3. For channels 3 and 4, enable output with TOER before initializing the pins with TIOR.
4. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence.)
5. Set the multi-function timer pulse unit 2 output with the general I/O port.
6. The count operation is started by TSTR.
7. Output goes low on compare-match occurrence.
8. An error occurs.
9. Set port output with the general I/O port and output the inverse of the active level.
10. The count operation is stopped by TSTR.
11. Not necessary when restarting in normal mode.
12. Initialize the pins with TIOR.
13. Set the multi-function timer pulse unit 2 output with the general I/O port.
14. Operation is restarted by TSTR.

chanhvinh.com

(2) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in PWM Mode 1

Figure 10.116 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in PWM mode 1 after re-setting.

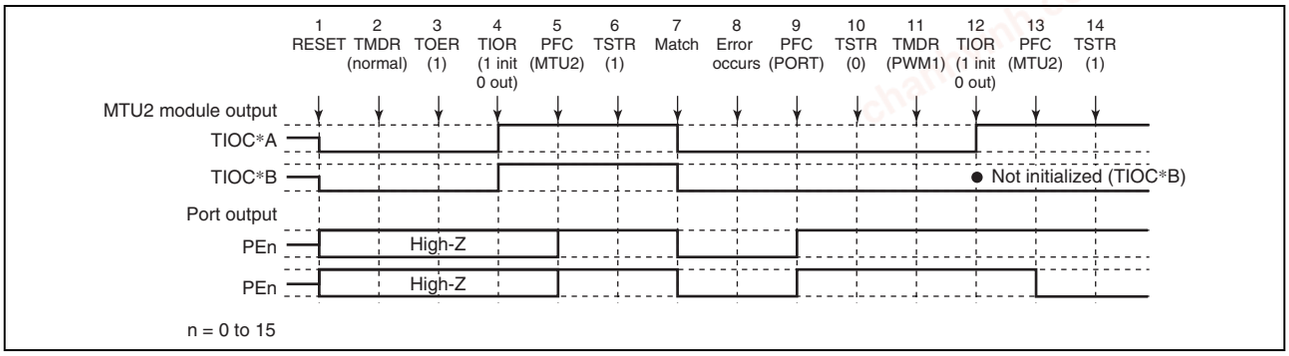


Figure 10.116 Error Occurrence in Normal Mode, Recovery in PWM Mode 1

1 to 10 are the same as in Figure 10.115.

11. Set PWM mode 1.
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized. If initialization is required, initialize in normal mode, and then switch to PWM mode 1.)
13. Set the multi-function timer pulse unit 2 output with the general I/O port.
14. Operation is restarted by TSTR.

chanhvinh.com

(3) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in PWM Mode 2

Figure 10.117 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in PWM mode 2 after re-setting.

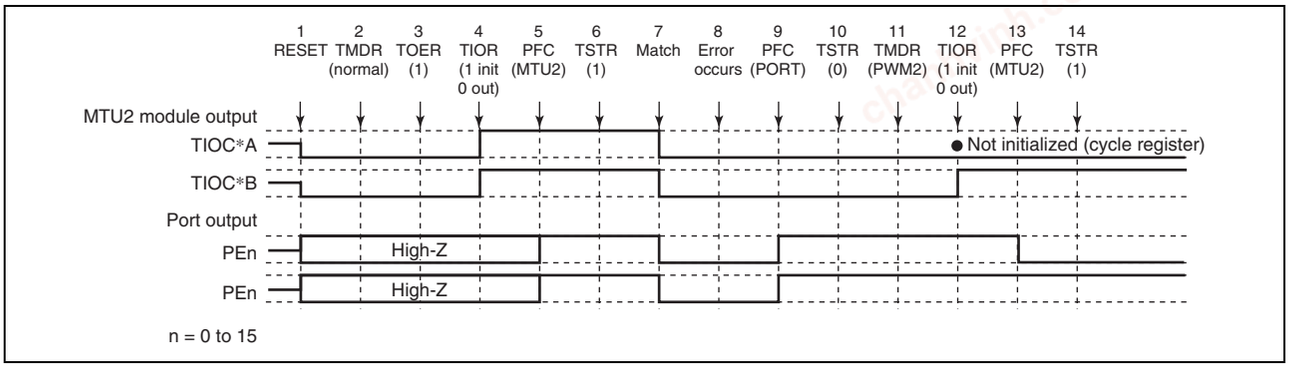


Figure 10.117 Error Occurrence in Normal Mode, Recovery in PWM Mode 2

1 to 10 are the same as in Figure 10.115.

11. Set PWM mode 2.
12. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized. If initialization is required, initialize in normal mode, and then switch to PWM mode 2.)
13. Set the multi-function timer pulse unit 2 output with the general I/O port.
14. Operation is restarted by TSTR.

Note: PWM mode 2 can only be set for channels 0 to 2, and therefore TOER setting is not necessary.

chanhvinh.com

(4) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Phase Counting Mode

Figure 10.118 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in phase counting mode after re-setting.

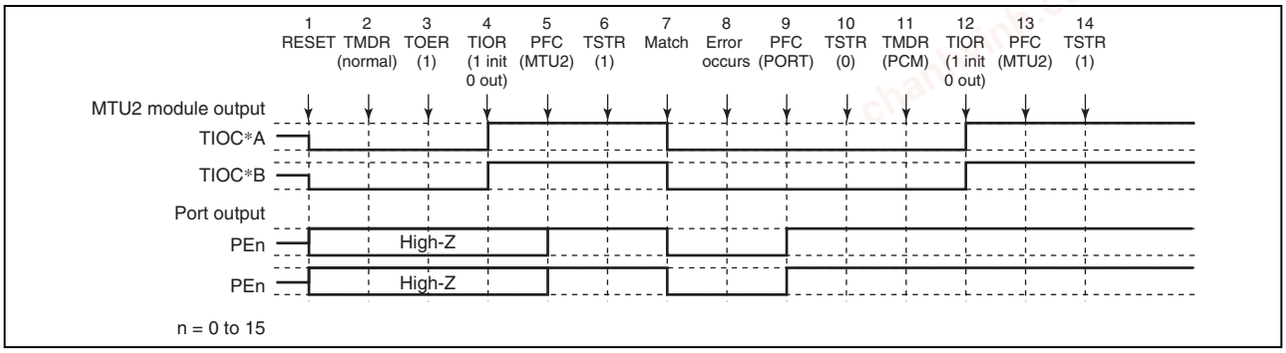


Figure 10.118 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode

1 to 10 are the same as in Figure 10.115.

11. Set phase counting mode.
12. Initialize the pins with TIOR.
13. Set the multi-function timer pulse unit 2 output with the general I/O port.
14. Operation is restarted by TSTR.

Note: Phase counting mode can only be set for channels 1 and 2, and therefore TOER setting is not necessary.

chanhvinh.com

(5) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Complementary PWM Mode

Figure 10.119 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in complementary PWM mode after re-setting.

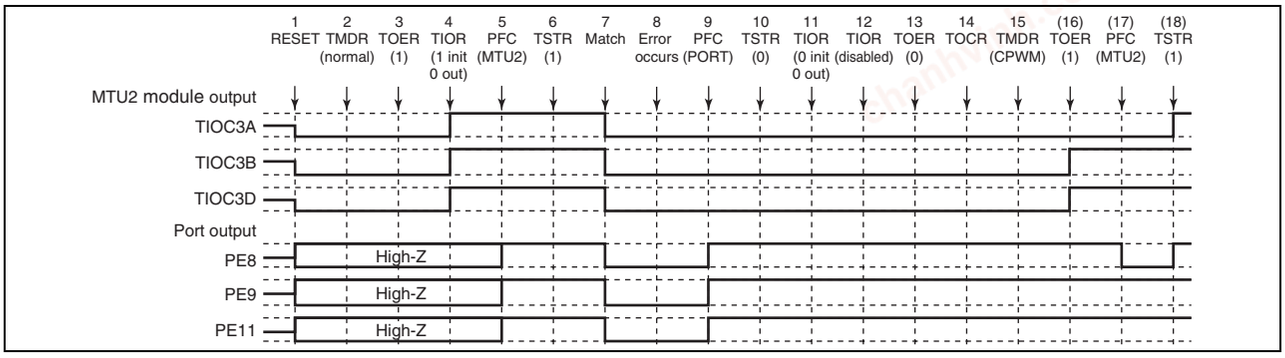


Figure 10.119 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in Figure 10.115.

11. Initialize the normal mode waveform generation section with TIOR.
12. Disable operation of the normal mode waveform generation section with TIOR.
13. Disable channel 3 and 4 output with TOER.
14. Select the complementary PWM output level and cyclic output enabling/disabling with TOCR.
15. Set complementary PWM.
16. Enable channel 3 and 4 output with TOER.
17. Set the multi-function timer pulse unit 2 output with the general I/O port.
18. Operation is restarted by TSTR.

chanhvinh.com

(6) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 10.120 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in reset-synchronized PWM mode after re-setting.

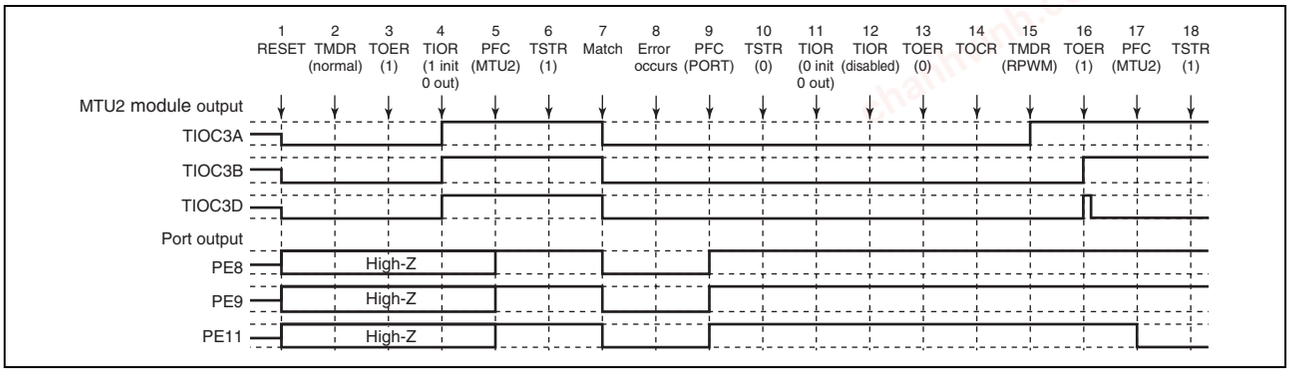


Figure 10.120 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode

1 to 13 are the same as in Figure 10.115.

14. Select the reset-synchronized PWM output level and cyclic output enabling/disabling with TOCR.
15. Set reset-synchronized PWM.
16. Enable channel 3 and 4 output with TOER.
17. Set the multi-function timer pulse unit 2 output with the general I/O port.
18. Operation is restarted by TSTR.

chanhvinh.com

(7) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Normal Mode

Figure 10.121 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in normal mode after re-setting.

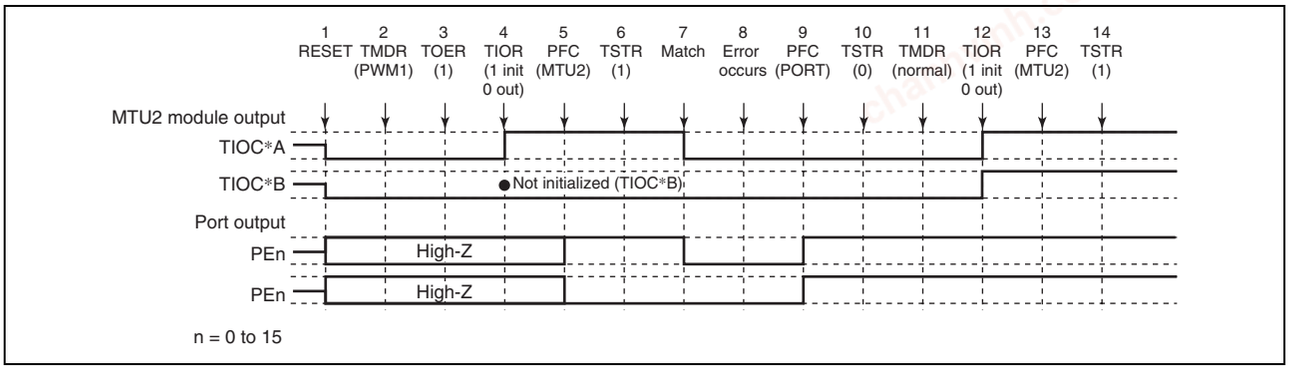


Figure 10.121 Error Occurrence in PWM Mode 1, Recovery in Normal Mode

1. After a reset, the module output is low and ports are in the high-impedance state.
2. Set PWM mode 1.
3. For channels 3 and 4, enable output with TOER before initializing the pins with TIOR.
4. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence. In PWM mode 1, the TIOC*B side is not initialized.)
5. Set the multi-function timer pulse unit 2 output with the general I/O port.
6. The count operation is started by TSTR.
7. Output goes low on compare-match occurrence.
8. An error occurs.
9. Set port output with the general I/O port and output the inverse of the active level.
10. The count operation is stopped by TSTR.
11. Set normal mode.
12. Initialize the pins with TIOR.
13. Set the multi-function timer pulse unit 2 output with the general I/O port.
14. Operation is restarted by TSTR.

chanhvinh.com

(8) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in PWM Mode 1

Figure 10.122 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in PWM mode 1 after re-setting.

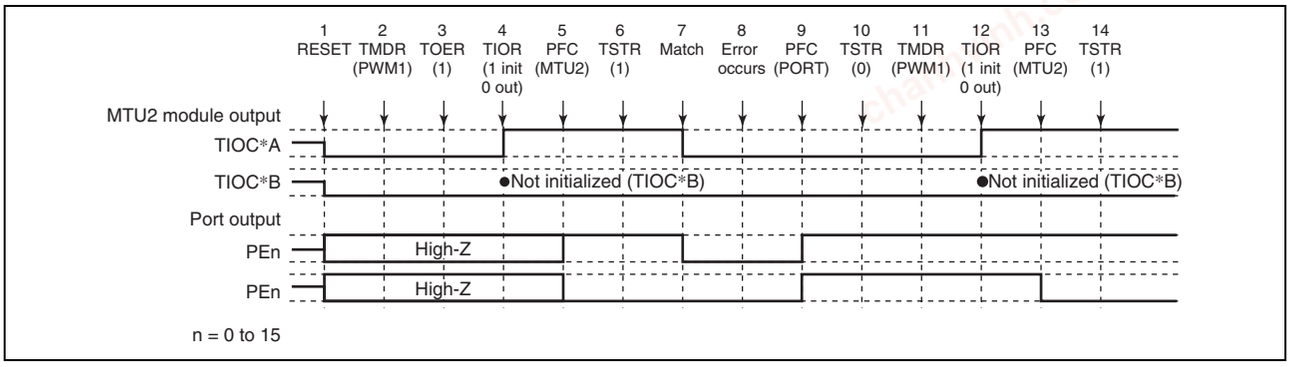


Figure 10.122 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1

1 to 10 are the same as in Figure 10.121.

11. Not necessary when restarting in PWM mode 1.
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized.)
13. Set the multi-function timer pulse unit 2 output with the general I/O port.
14. Operation is restarted by TSTR.

chanhvinh.com

(9) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in PWM Mode 2

Figure 10.123 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in PWM mode 2 after re-setting.

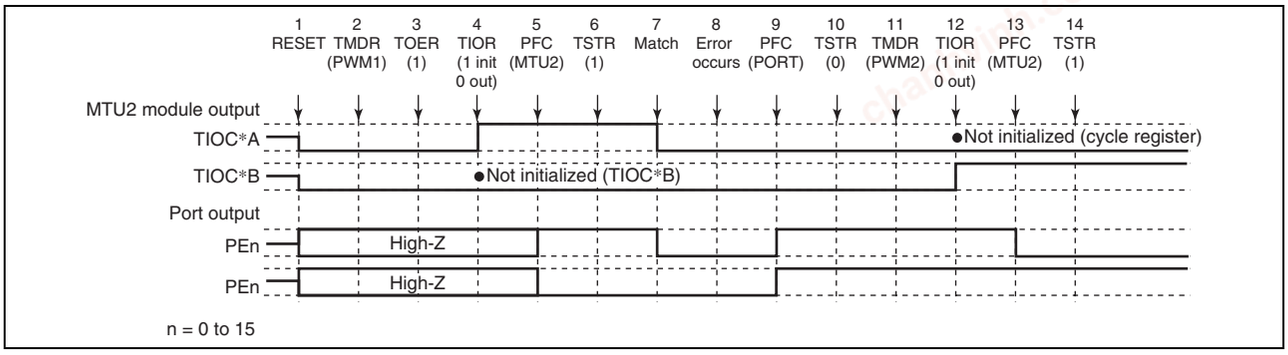


Figure 10.123 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2

1 to 10 are the same as in Figure 10.121.

11. Set PWM mode 2.
12. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)
13. Set the multi-function timer pulse unit 2 output with the general I/O port.
14. Operation is restarted by TSTR.

Note: PWM mode 2 can only be set for channels 0 to 2, and therefore TOER setting is not necessary.

chanhvinh.com

(10) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Phase Counting Mode

Figure 10.124 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in phase counting mode after re-setting.

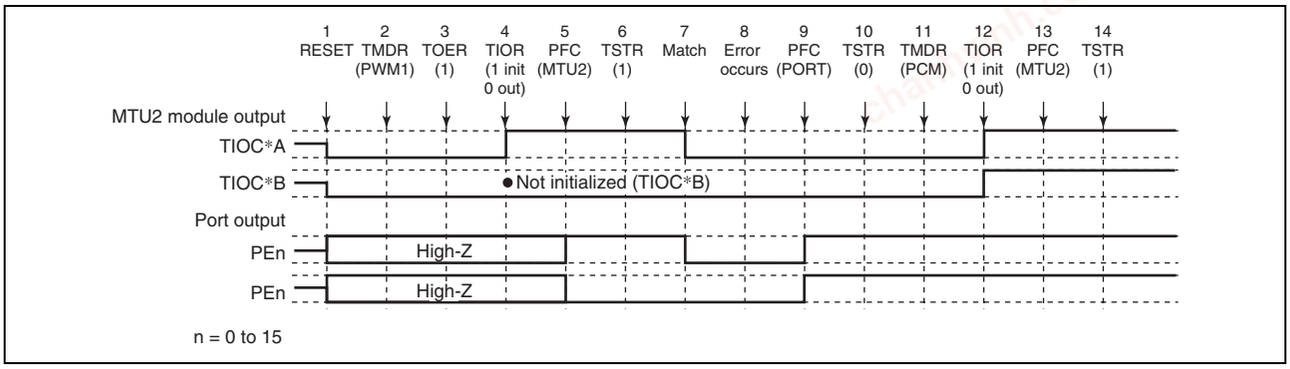


Figure 10.124 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode

1 to 10 are the same as in Figure 10.121.

11. Set phase counting mode.
12. Initialize the pins with TIOR.
13. Set the multi-function timer pulse unit 2 output with the general I/O port.
14. Operation is restarted by TSTR.

Note: Phase counting mode can only be set for channels 1 and 2, and therefore TOER setting is not necessary.

chanhvinh.com

(11) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Complementary PWM Mode

Figure 10.125 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in complementary PWM mode after re-setting.

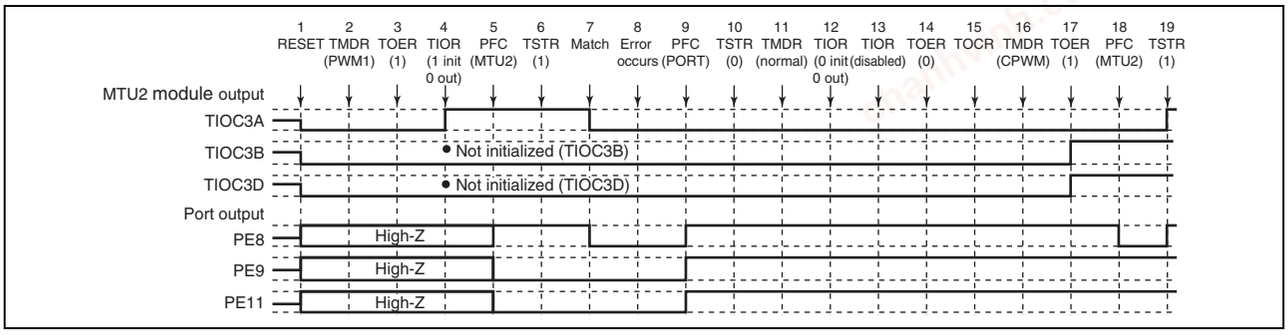


Figure 10.125 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

1 to 10 are the same as in Figure 10.121.

11. Set normal mode for initialization of the normal mode waveform generation section.
12. Initialize the PWM mode 1 waveform generation section with TIOR.
13. Disable operation of the PWM mode 1 waveform generation section with TIOR.
14. Disable channel 3 and 4 output with TOER.
15. Select the complementary PWM output level and cyclic output enabling/disabling with TOCR.
16. Set complementary PWM.
17. Enable channel 3 and 4 output with TOER.
18. Set the multi-function timer pulse unit 2 output with the general I/O port.
19. Operation is restarted by TSTR.

chanhvinh.com

(12) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 10.126 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in reset-synchronized PWM mode after re-setting.

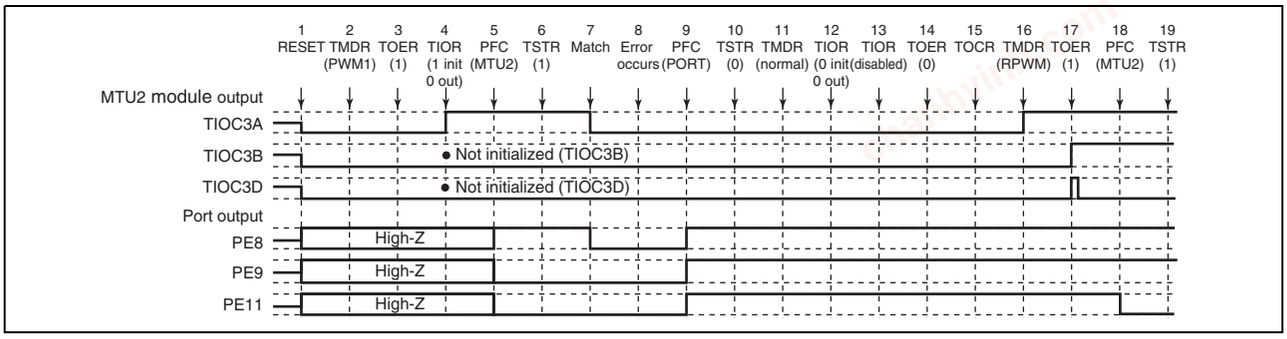


Figure 10.126 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode

1 to 14 are the same as in Figure 10.125.

15. Select the reset-synchronized PWM output level and cyclic output enabling/disabling with TOCR.
16. Set reset-synchronized PWM.
17. Enable channel 3 and 4 output with TOER.
18. Set the multi-function timer pulse unit 2 output with the general I/O port.
19. Operation is restarted by TSTR.

chanhvinh.com

(13) Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in Normal Mode

Figure 10.127 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.

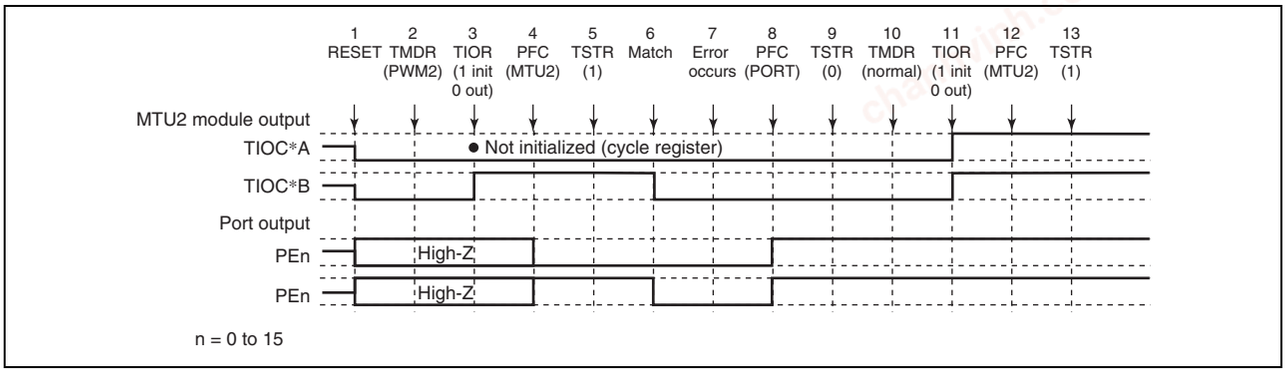


Figure 10.127 Error Occurrence in PWM Mode 2, Recovery in Normal Mode

1. After a reset, the module output is low and ports are in the high-impedance state.
2. Set PWM mode 2.
3. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence. In PWM mode 2, the cycle register pins are not initialized. In the example, TIOC *A is the cycle register.)
4. Set the multi-function timer pulse unit 2 output with the general I/O port.
5. The count operation is started by TSTR.
6. Output goes low on compare-match occurrence.
7. An error occurs.
8. Set port output with the general I/O port and output the inverse of the active level.
9. The count operation is stopped by TSTR.
10. Set normal mode.
11. Initialize the pins with TIOR.
12. Set the multi-function timer pulse unit 2 output with the general I/O port.
13. Operation is restarted by TSTR.

chanhvinh.com

(14) Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in PWM Mode 1

Figure 10.128 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in PWM mode 1 after re-setting.

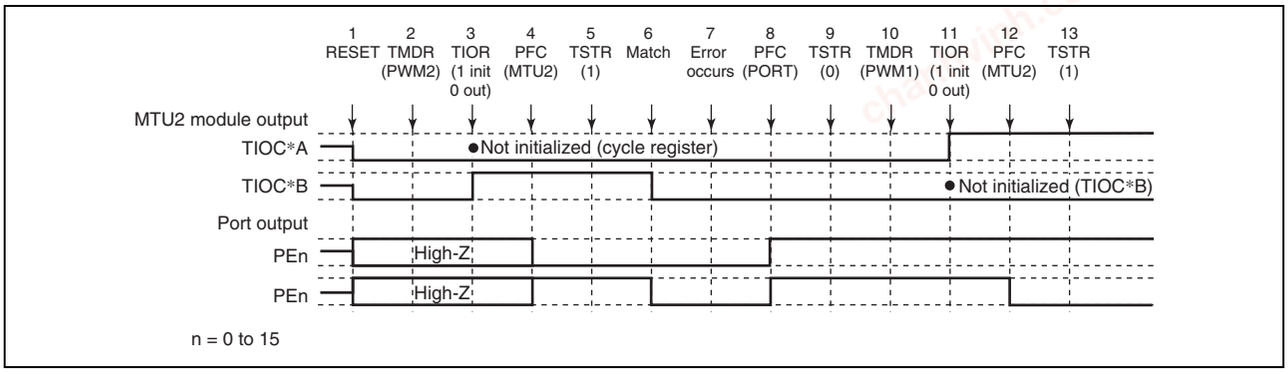


Figure 10.128 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1

1 to 9 are the same as in Figure 10.127.

10. Set PWM mode 1.
11. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized.)
12. Set the multi-function timer pulse unit 2 output with the general I/O port.
13. Operation is restarted by TSTR.

chanhvinh.com

(15) Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in PWM Mode 2

Figure 10.129 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in PWM mode 2 after re-setting.

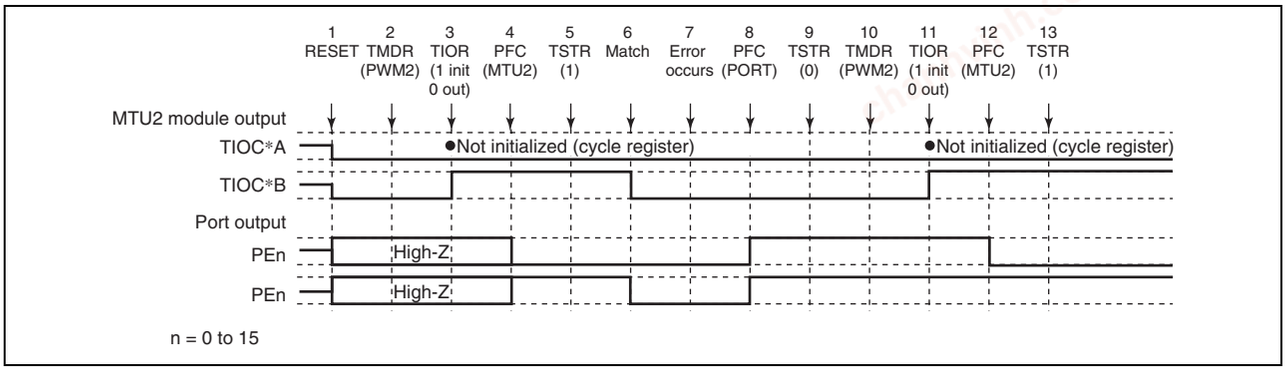


Figure 10.129 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

1 to 9 are the same as in Figure 10.127.

10. Not necessary when restarting in PWM mode 2.
11. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)
12. Set the multi-function timer pulse unit 2 output with the general I/O port.
13. Operation is restarted by TSTR.

chanhvinh.com

(16) Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in Phase Counting Mode

Figure 10.130 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in phase counting mode after re-setting.

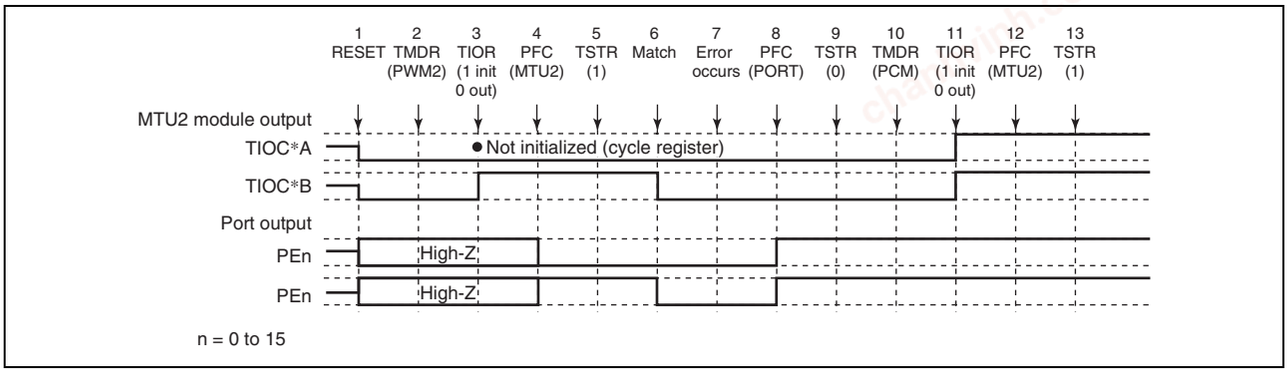


Figure 10.130 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode

1 to 9 are the same as in Figure 10.127.

10. Set phase counting mode.
11. Initialize the pins with TIOR.
12. Set the multi-function timer pulse unit 2 output with the general I/O port.
13. Operation is restarted by TSTR.

chanhvinh.com

(17) Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in Normal Mode

Figure 10.131 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in normal mode after re-setting.

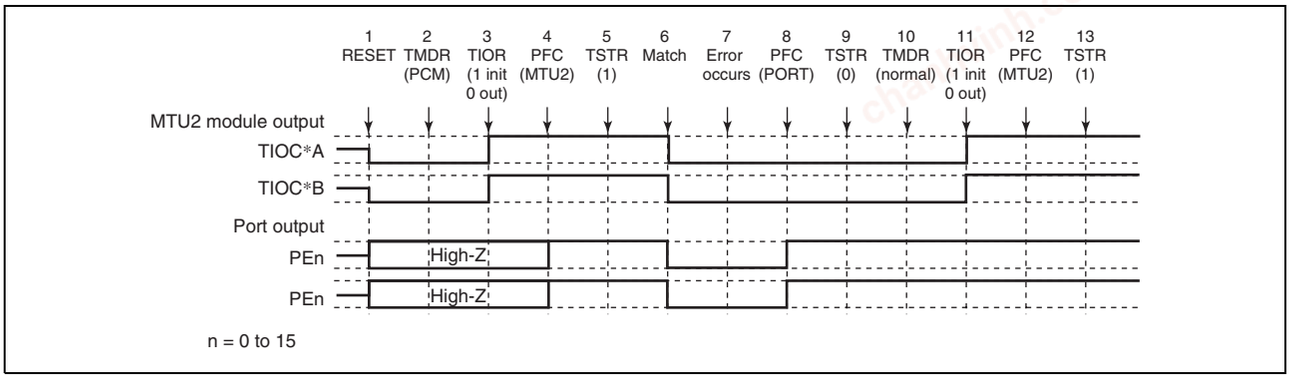


Figure 10.131 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode

1. After a reset, the module output is low and ports are in the high-impedance state.
2. Set phase counting mode.
3. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence.)
4. Set the multi-function timer pulse unit 2 output with the general I/O port.
5. The count operation is started by TSTR.
6. Output goes low on compare-match occurrence.
7. An error occurs.
8. Set port output with the general I/O port and output the inverse of the active level.
9. The count operation is stopped by TSTR.
10. Set in normal mode.
11. Initialize the pins with TIOR.
12. Set the multi-function timer pulse unit 2 output with the general I/O port.
13. Operation is restarted by TSTR.

chanhvinh.com

(18) Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in PWM Mode 1

Figure 10.132 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in PWM mode 1 after re-setting.

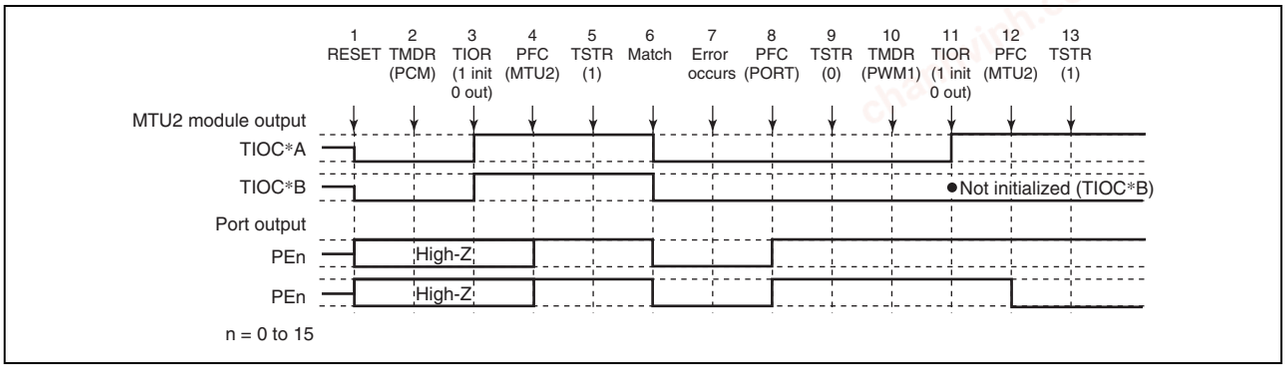


Figure 10.132 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1

1 to 9 are the same as in Figure 10.131.

10. Set PWM mode 1.
11. Initialize the pins with TIOR. (In PWM mode 1, the TIOC *B side is not initialized.)
12. Set the multi-function timer pulse unit 2 output with the general I/O port.
13. Operation is restarted by TSTR.

chanhvinh.com

(19) Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in PWM Mode 2

Figure 10.133 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in PWM mode 2 after re-setting.

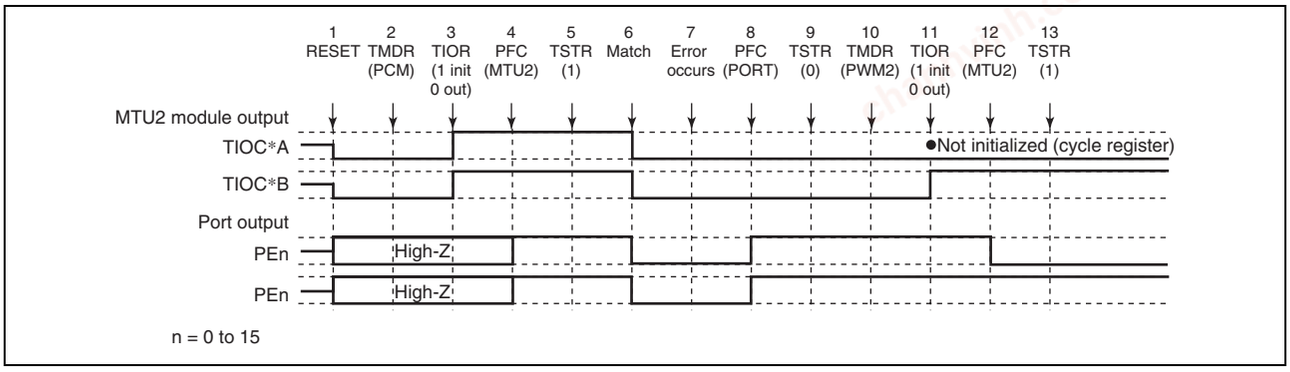


Figure 10.133 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2

1 to 9 are the same as in Figure 10.131.

10. Set PWM mode 2.
11. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)
12. Set the multi-function timer pulse unit 2 output with the general I/O port.
13. Operation is restarted by TSTR.

chanhvinh.com

(20) Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in Phase Counting Mode

Figure 10.134 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in phase counting mode after re-setting.

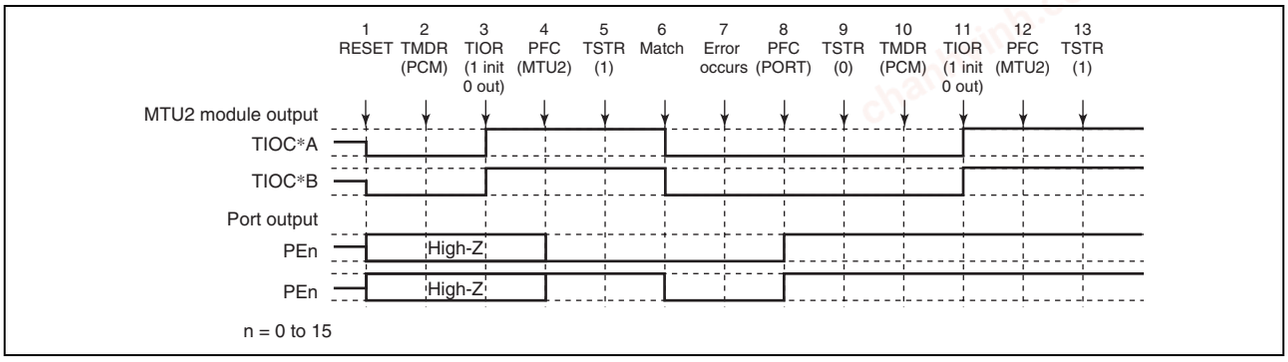


Figure 10.134 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode

1 to 9 are the same as in Figure 10.131.

10. Not necessary when restarting in phase counting mode.
11. Initialize the pins with TIOR.
12. Set the multi-function timer pulse unit 2 output with the general I/O port.
13. Operation is restarted by TSTR.

chanhvinh.com

(21) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Normal Mode

Figure 10.135 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in normal mode after re-setting.

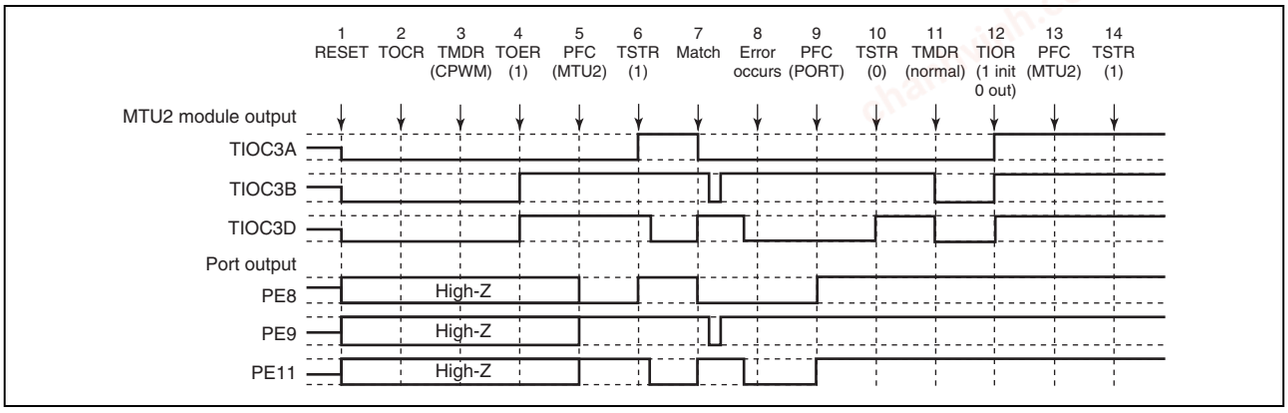


Figure 10.135 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode

1. After a reset, the module output is low and ports are in the high-impedance state.
2. Select the complementary PWM output level and cyclic output enabling/disabling with TOCR.
3. Set complementary PWM.
4. Enable channel 3 and 4 output with TOER.
5. Set the multi-function timer pulse unit 2 output with the general I/O port.
6. The count operation is started by TSTR.
7. The complementary PWM waveform is output on compare-match occurrence.
8. An error occurs.
9. Set port output with the general I/O port and output the inverse of the active level.
10. The count operation is stopped by TSTR. (This module outputs the same value as the complementary PWM output initial value.)
11. Set normal mode. (This module outputs a low-level signal.)
12. Initialize the pins with TIOR.
13. Set the multi-function timer pulse unit 2 output with the general I/O port.
14. Operation is restarted by TSTR.

chanhvinh.com

(22) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in PWM Mode 1

Figure 10.136 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in PWM mode 1 after re-setting.

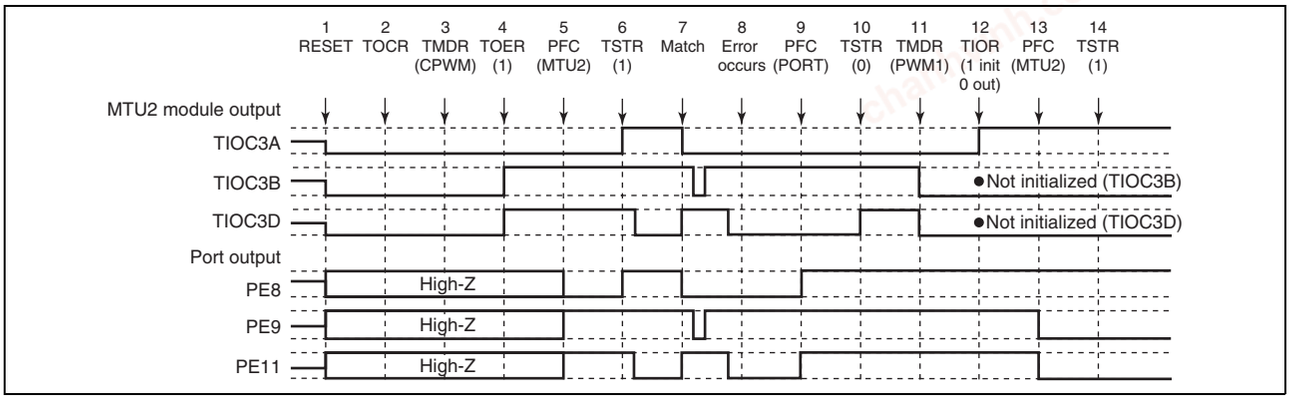


Figure 10.136 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1

1 to 10 are the same as in Figure 10.135.

11. Set PWM mode 1. (This module outputs a low-level signal.)
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC *B side is not initialized.)
13. Set the multi-function timer pulse unit 2 output with the general I/O port.
14. Operation is restarted by TSTR.

chanhvinh.com

(23) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Complementary PWM Mode

Figure 10.137 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (when operation is restarted using the cycle and duty settings at the time the counter was stopped).

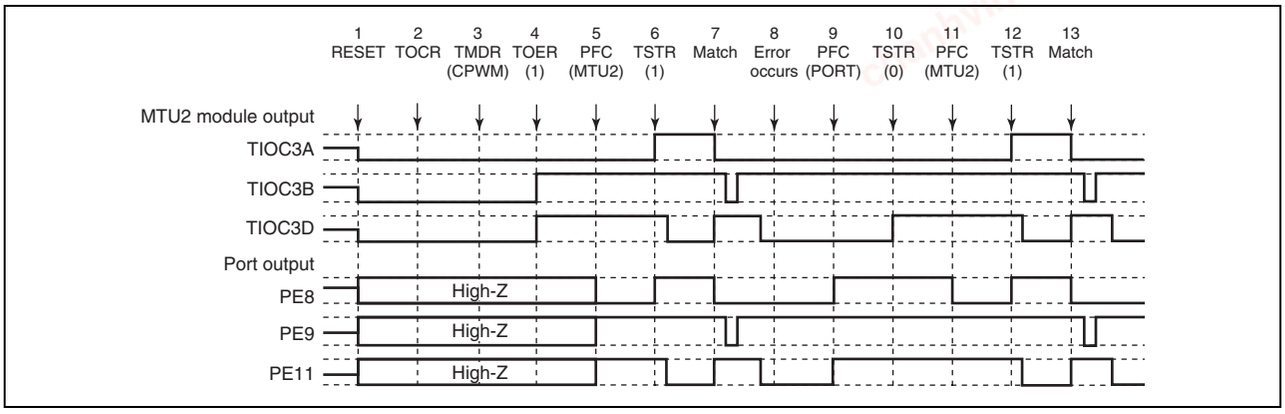


Figure 10.137 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in Figure 10.135.

11. Set the multi-function timer pulse unit 2 output with the general I/O port.
12. Operation is restarted by TSTR.
13. The complementary PWM waveform is output on compare-match occurrence.

chanhvinh.com

(24) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Complementary PWM Mode

Figure 10.138 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (when operation is restarted using completely new cycle and duty settings).

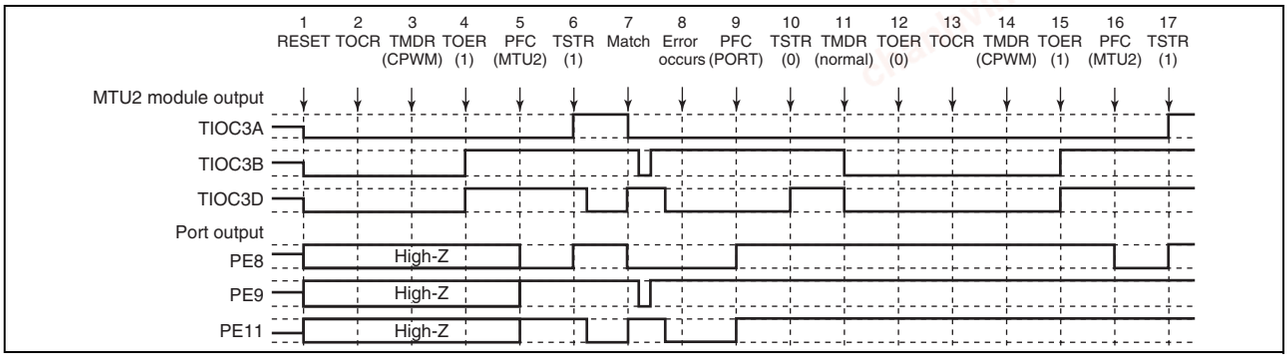


Figure 10.138 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in Figure 10.135.

11. Set normal mode and make new settings. (This module outputs a low-level signal.)
12. Disable channel 3 and 4 output with TOER.
13. Select the complementary PWM mode output level and cyclic output enabling/disabling with TOCR.
14. Set complementary PWM.
15. Enable channel 3 and 4 output with TOER.
16. Set the multi-function timer pulse unit 2 output with the general I/O port.
17. Operation is restarted by TSTR.

chanhvinh.com

(25) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 10.139 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in reset-synchronized PWM mode.

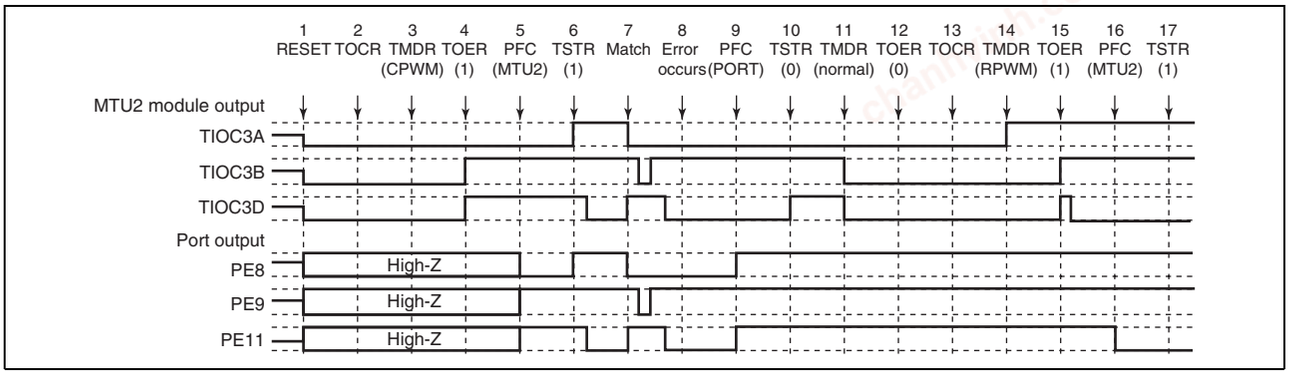


Figure 10.139 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode

1 to 10 are the same as in Figure 10.135.

11. Set normal mode. (This module outputs a low-level signal.)
12. Disable channel 3 and 4 output with TOER.
13. Select the reset-synchronized PWM mode output level and cyclic output enabling/disabling with TOCR.
14. Set reset-synchronized PWM.
15. Enable channel 3 and 4 output with TOER.
16. Set the multi-function timer pulse unit 2 output with the general I/O port.
17. Operation is restarted by TSTR.

chanhvinh.com

(26) Operation when Error Occurs during Reset-Synchronized PWM Mode Operation, and Operation is Restarted in Normal Mode

Figure 10.140 shows an explanatory diagram of the case where an error occurs in reset-synchronized PWM mode and operation is restarted in normal mode after re-setting.

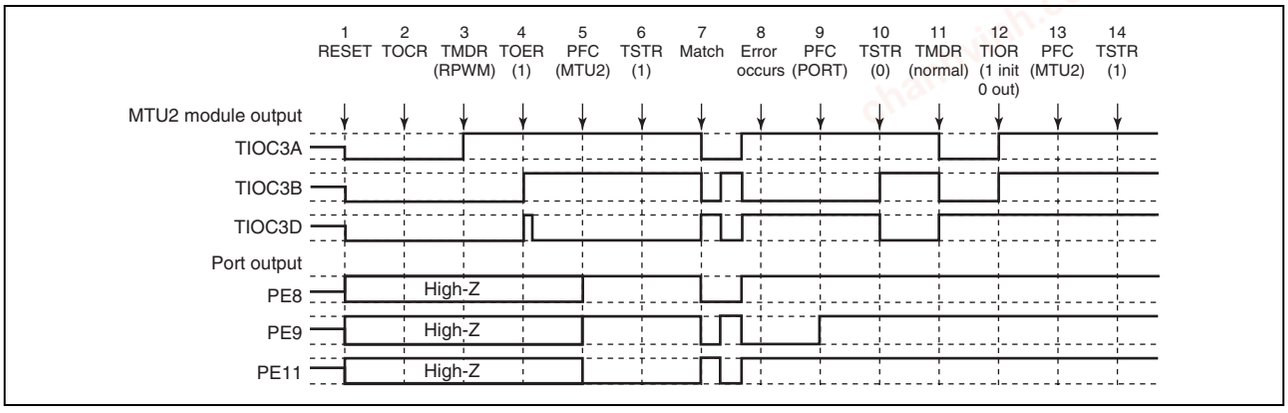


Figure 10.140 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode

1. After a reset, the module output is low and ports are in the high-impedance state.
2. Select the reset-synchronized PWM output level and cyclic output enabling/disabling with TOCR.
3. Set reset-synchronized PWM.
4. Enable channel 3 and 4 output with TOER.
5. Set the multi-function timer pulse unit 2 output with the general I/O port.
6. The count operation is started by TSTR.
7. The reset-synchronized PWM waveform is output on compare-match occurrence.
8. An error occurs.
9. Set port output with the general I/O port and output the inverse of the active level.
10. The count operation is stopped by TSTR. (This module outputs the same value as the reset-synchronized PWM output initial value.)
11. Set normal mode. (The positive phase output from this module is low, and negative phase output is high.)
12. Initialize the pins with TIOR.
13. Set the multi-function timer pulse unit 2 output with the general I/O port.
14. Operation is restarted by TSTR.

chanhvinh.com

(27) Operation when Error Occurs during Reset-Synchronized PWM Mode Operation, and Operation is Restarted in PWM Mode 1

Figure 10.141 shows an explanatory diagram of the case where an error occurs in reset-synchronized PWM mode and operation is restarted in PWM mode 1 after re-setting.

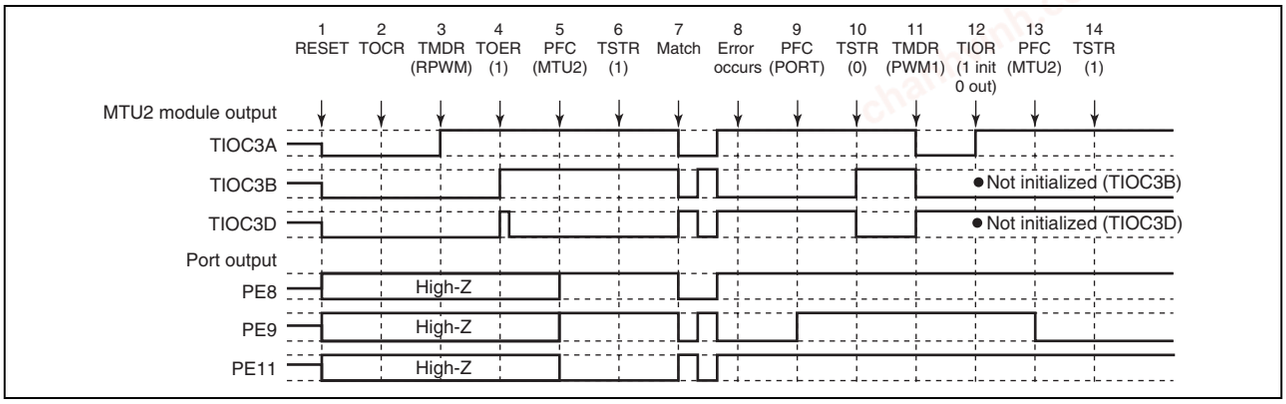


Figure 10.141 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1

1 to 10 are the same as in Figure 10.140.

11. Set PWM mode 1. (The positive phase output from this module is low, and negative phase output is high.)
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC *B side is not initialized.)
13. Set the multi-function timer pulse unit 2 output with the general I/O port.
14. Operation is restarted by TSTR.

chanhvinh.com

(28) Operation when Error Occurs during Reset-Synchronized PWM Mode Operation, and Operation is Restarted in Complementary PWM Mode

Figure 10.142 shows an explanatory diagram of the case where an error occurs in reset-synchronized PWM mode and operation is restarted in complementary PWM mode after re-setting.

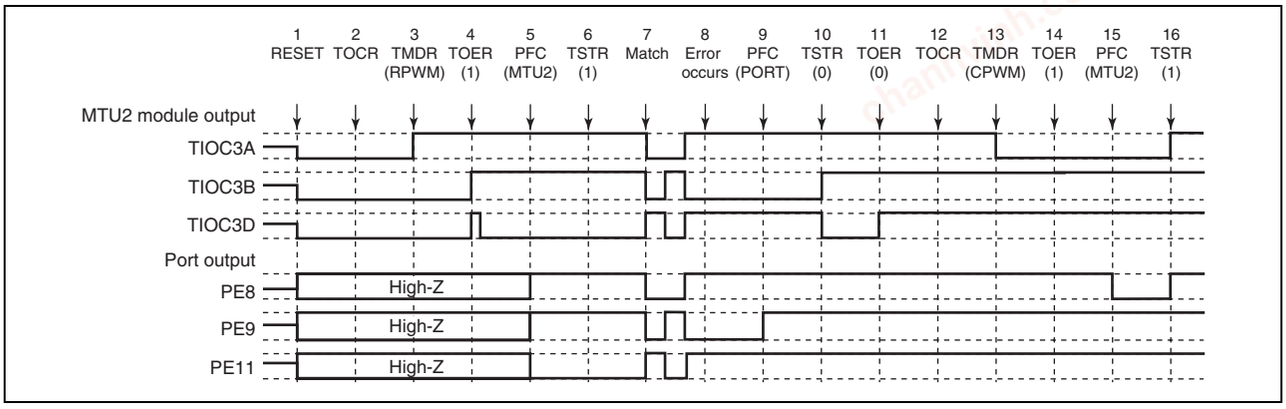


Figure 10.142 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in Figure 10.140.

11. Disable channel 3 and 4 output with TOER.
12. Select the complementary PWM output level and cyclic output enabling/disabling with TOCR.
13. Set complementary PWM. (The cyclic output pin of this module outputs a low-level signal.)
14. Enable channel 3 and 4 output with TOER.
15. Set the multi-function timer pulse unit 2 output with the general I/O port.
16. Operation is restarted by TSTR.

chanhvinh.com

(29) Operation when Error Occurs during Reset-Synchronized PWM Mode Operation, and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 10.143 shows an explanatory diagram of the case where an error occurs in reset-synchronized PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

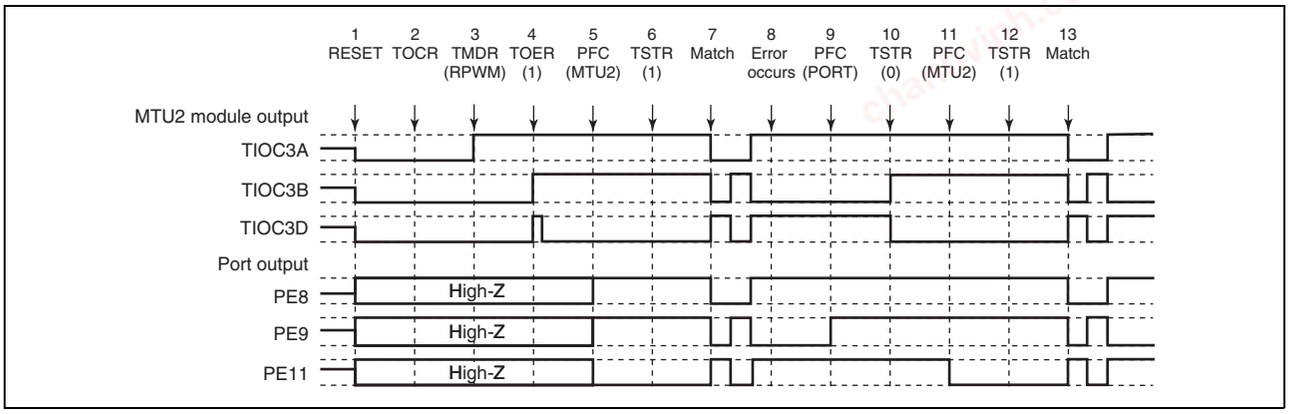


Figure 10.143 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode

1 to 10 are the same as in Figure 10.140.

11. Set the multi-function timer pulse unit 2 output with the general I/O port.
12. Operation is restarted by TSTR.
13. The reset-synchronized PWM waveform is output on compare-match occurrence.

chanhvinh.com

11. OS Timer

11.1 Functional Overview

The OS timer has the following features.

- Two operating modes
  - Interval timer mode
  - Free-running comparison mode
- Choice between startup of DMA by compare match and generation of interrupt

11.1.1 Features of OSTM

Channels

This product has the following number of channels of the OS timer.

Table 11.1 Channels of OS timer

	OS Timer
Number of Channels	2
Name	OSTMn

Meaning of n

Throughout this section, the individual channels of the OS timer are identified by the index "n" (n = 0, 1), for example OSTMnTO for the OS timer n output register.

Register address

The register addresses of the OS timer are given as offsets from the individual base addresses <OSTMn_base>.

The register base addresses of each OSTMn are listed in the following table.

Table 11.2 Register Base Addresses

Base Address Name	Base Address
<OSTM0_base>	FCFE C000$_H$
<OSTM1_base>	FCFE C400$_H$

Interrupts

The OS timers can generate the following interrupt requests.

Table 11.3 OSTMn Interrupt Requests

OSTMn Signal	Function	Startup of Direct Memory Access Controller
OSTM0TINT	OSTM0 interrupt	√
OSTM1TINT	OSTM1 interrupt	√

chanhvinh.com

11.2 Registers

The OS timer is controlled and operated by the following registers.

11.2.1 Registers Overview

The list of OSTMn (n = 0, 1) registers and the memory addresses are as follows.

For the base addresses, see the Table 11.2.

For the actual addresses, the offset values indicated in the following table are added to the base addresses.

Register Name	Function	R/W	Reset Value	Access Unit (bit)			Address
				8	16	32	
OSTMnCMP	OSTM compare register	R/W	0000 0000_H	—	—	√	<OSTMn_base> + 00_H
OSTMnCNT	OSTM counter register	R	FFFF $FFFF_H$	—	—	√	<OSTMn_base> + 04_H
OSTMnTE	OSTM count enable status register	R	00_H	√	—	—	<OSTMn_base> + 10_H
OSTMnTS	OSTM count start trigger register	W	00_H	√	—	—	<OSTMn_base> + 14_H
OSTMnTT	OSTM count stop trigger register	W	00_H	√	—	—	<OSTMn_base> + 18_H
OSTMnCTL	OSTM control register	R/W	00_H	√	—	—	<OSTMn_base> + 20_H

chanhvinh.com

11.2.2 Details of OSTM Registers

11.2.2.1 OSTMnCMP — OSTM Compare Register

Depending on the mode of operation, this register holds the start value for the down-counter or the value for comparison with that of the counter.

Access: This register is readable/writable in 32-bit units.

Address: <OSTMn_base>

Initial value: $0000\ 0000_H$

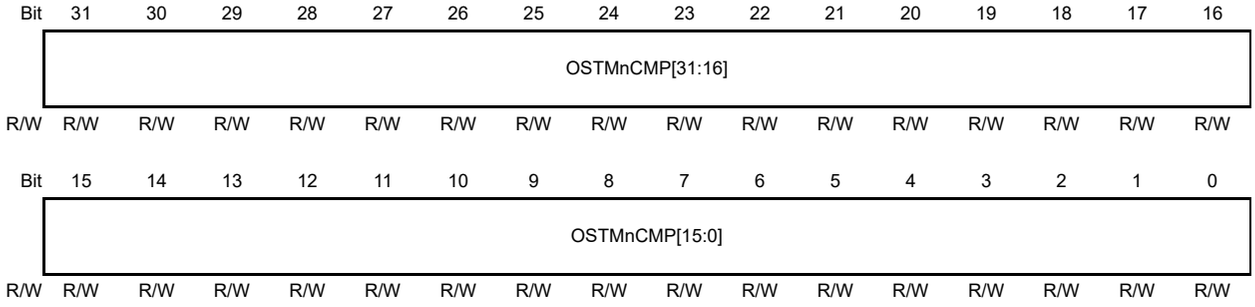


Table 11.4 OSTMnCMP register contents

Bit Position	Bit Name	Function
31 to 0	OSTMnCMP [31:0]	• In interval timer mode: start value of the down-counter • In free-running comparison mode: value for comparison

chanhvinh.com

11.2.2.2 OSTMnCNT — OSTM Counter Register

This register indicates the counter value of the timer.

Access: This register is readable in 32-bit units.

Address: OSTMn_base> + 4_H

Initial value: The initial value depends on the operating mode of the OS timer. Refer to Table 11.6, Correspondence between Operating Mode, Counting Direction and Initial Value.

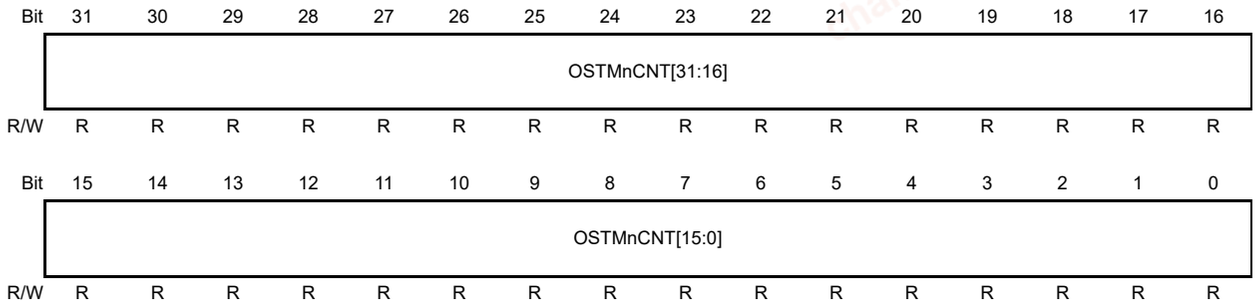


Table 11.5 OSTMnCNT register contents

Bit Position	Bit Name	Function
31 to 0	OSTMnCNT [31:0]	32-bit counter value

The following table shows the correspondence between operating mode, counting direction and initial value. The initial value is the value read from the counter after a change to the operating mode.

Table 11.6 Correspondence between Operating Mode, Counting Direction and Initial Value

Timer Operating Mode	OSTMnCTL.OSTMnMD1	Counting Direction	Initial value
Interval timer mode	0 *1	Down	FFFF FFFF$_H$
Free-running comparison mode	1	Up	0000 0000$_H$

Note 1. Value after reset

chanhvinh.com

11.2.2.3 OSTMnTE — OSTM Count Enable Status Register

This register indicates whether the counter is enabled or disabled.

Access: This register is readable in 8-bit units.

Address: <OSTMn_base>+ 10_H

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	OSTMnTE
R/W	R	R	R	R	R	R	R	R

Table 11.7 OSTMnTE register contents

Bit Position	Bit Name	Function
7 to 1	Reserved	Reserved These bits are always read as 0. The write value should always be 0.
0	OSTMnTE	This bit indicates whether the counter is enabled or disabled. 0: Counter disabled 1: Counter enabled This bit is set to 1 in response to OSTMnTS.OSTMnTS being set to 1. This bit is reset to 0 in response to OSTMnTT.OSTMnTT being set to 1.

NOTE

When OSTMnTE = 0, the counter retains its value.

If the counter is restarted, it

- restarts counting down from the value in the OSTMnCMP register if it is in interval timer mode or
- restarts counting up from the counter value 0000 0000_H if it is in free running comparison mode.

11.2.2.4 OSTMnTS — OSTM Count Start Trigger Register

This register starts the counter.

Access: This register is writable in 8-bit units. It is always read as 00_H.

Address: <OSTMn_base>+ 14_H

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	OSTMnTS
R/W	R	R	R	R	R	R	R	W

Table 11.8 OSTMnTS register contents

Bit Position	Bit Name	Function
7 to 1	Reserved	Reserved These bits are always read as 0. The write value should always be 0.
0	OSTMnTS	This bit starts the counter. 0: This setting has no effect. 1: Starts the counter and sets OSTMnTE.OSTMnTE = 1. • In interval timer mode, a forced restart is executed if this bit is set while OSTMnTE.OSTMnTE = 1. • In free-running comparison mode, setting this bit is ignored as long as OSTMnTE.OSTMnTE = 1.

chanhvinh.com

11.2.2.5 OSTMnTT — OSTM Count Stop Trigger Register

This register stops the counter.

Access: This register is writable in 8-bit units. It is always read as 00_H.

Address: <OSTMn_base>+ 18_H

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	OSTMnTT
R/W	R	R	R	R	R	R	R	W

Table 11.9 OSTMnTT register contents

Bit Position	Bit Name	Function
7 to 1	Reserved	Reserved These bits are always read as 0. The write value should always be 0.
0	OSTMnTT	Stops the counter. 0: This setting has no effect. 1: Stops the counter and clears the OSTMnTE.OSTMnTE bit.

11.2.2.6 OSTMnCTL — OSTM Control Register

This register specifies the operating mode for the counter and controls enabling/disabling of OSTMnTINT interrupt requests when counting starts.

Access: This register is readable/writable in 8-bit units. Writing to this register is only possible if the counter is disabled (OSTMnTOE.OSTMnTOE = 0).

Address: <OSTMn_base>+ 20_H

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	OSTMnMD1	OSTMnMD0
R/W	R	R	R	R	R	R	R/W	R/W

Table 11.10 OSTMnCTL register contents

Bit Position	Bit Name	Function
7 to 2	Reserved	Reserved These bits are always read as 0. The write value should always be 0.
1	OSTMnMD1	Specifies the operating mode for the counter. 0: Interval timer mode 1: Free-running comparison mode
0	OSTMnMD0	Controls enabling/disabling of OSTMnTINT interrupt requests when counting starts. 0: Disables the interrupts when counting starts. 1: Enables the interrupts when counting starts.

chanhvinh.com

11.3 Functional Description

Each OS timer is a 32-bit timer/counter.

The settings for operating mode specify the direction of counting (up or down) and the generation of interrupt requests.

11.3.1 Block Diagram

The following block diagram shows the main components of OSTM.

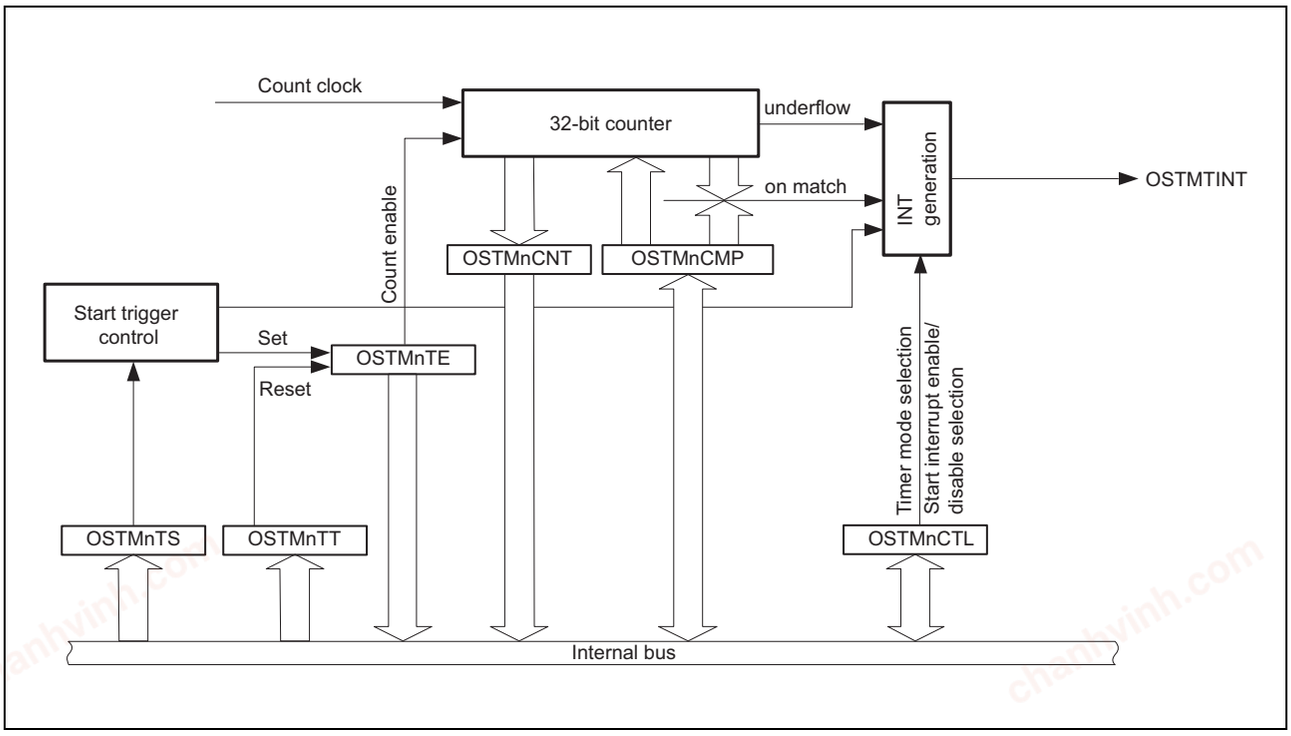


Figure 11.1 Block Diagram of OSTM

chanhvinh.com

11.3.2 Count Clock

The count clock of OSTMn is P0ϕ.

11.3.3 Generation of Interrupt Request

An OSTMnTINT interrupt request is generated whenever the counter reaches 0000 0000_H (in interval timer mode) or matches the comparison value (in free-running comparison mode).

An interrupt request can also be generated on starting and restarting of the counter. This is controlled by the OSTMnCTL.OSTMnMD0 bit.

This operation is shown in the following figure.

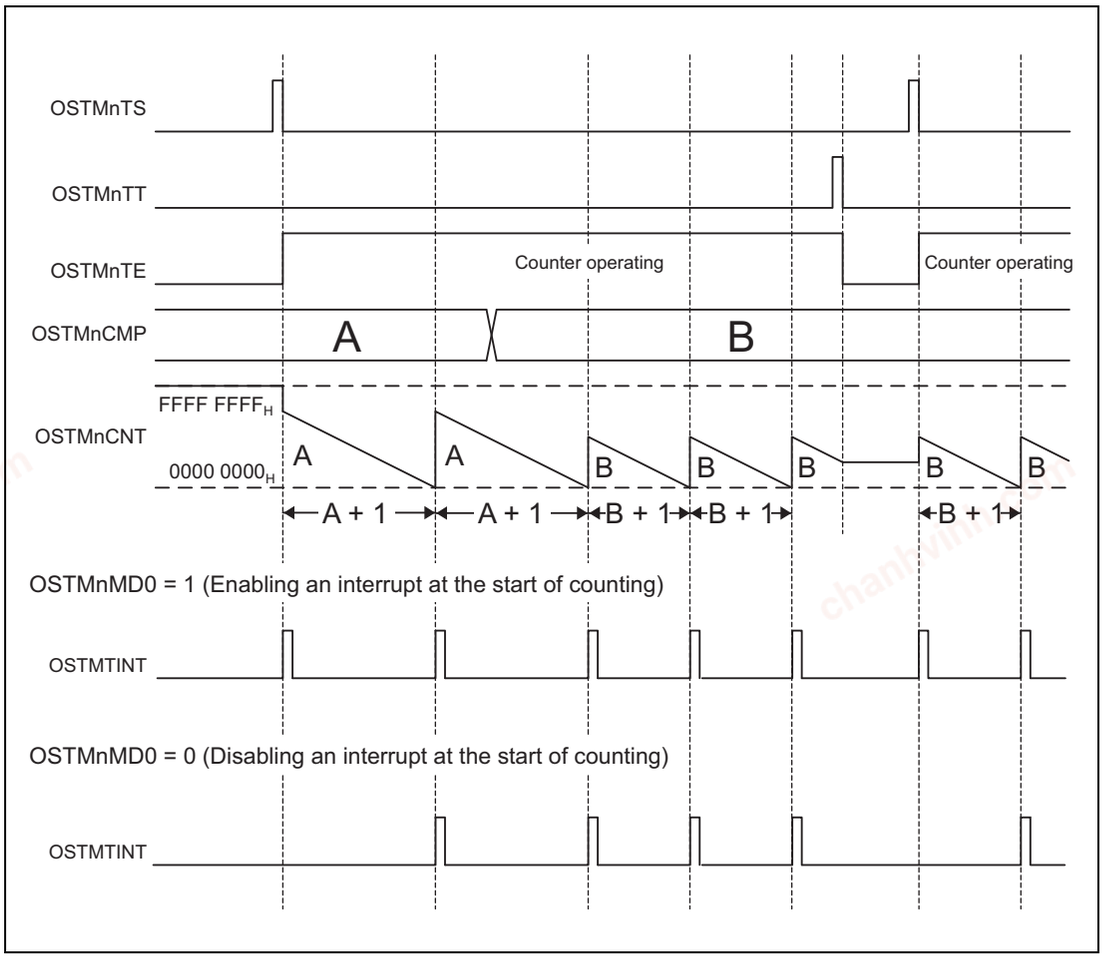


Figure 11.2 Generating an Interrupt when Counting Starts (in Interval Timer Mode)

chanhvinh.com

11.3.4 Starting and Stopping the Timer

The OS timer is started and stopped as follows.

Starting the timer

The timer is started in either of the following way:

- setting the OSTMnTS.OSTMnTSF bit to 1

Status bit OSTMnTE.OSTMnTE is set to 1.

The counter starts to count up or down in accord with the settings for operating mode.

Stopping the timer

Setting the OSTMnTT.OSTMnTT bit to 1 stops the timer.

This also clears the OSTMnTE.OSTMnTE status flag.

11.3.5 Interval Timer Mode

Select the interval timer mode when an OS timer is to be used as a reference timer for generating interrupt requests at a fixed interval.

11.3.5.1 Basic Operation in Interval Timer Mode

In interval timer mode, the timer counts down from the value specified in the OSTMnCMP register. An OSTMnTINT interrupt request is generated when the counter reaches $0000\ 0000_H$.

Select interval timer mode by setting OSTMnCTL.OSTMnMD1 = 0.

New values can be written to the OSTMnCMP register at any time. If it is rewritten during count operation, the counter loads the new OSTMnCMP value when the next $0000\ 0000_H$ is reached.

Cycles of OSTMnTINT output

The cycle of OSTMnTINT output is as follows.

- OSTMnTINT generation cycle = counter-clock cycle × (OSTMnCMP + 1)

chanhvinh.com

The following figure shows the basic operation of OSTM when counter-start interrupts is enabled in interval timer mode.

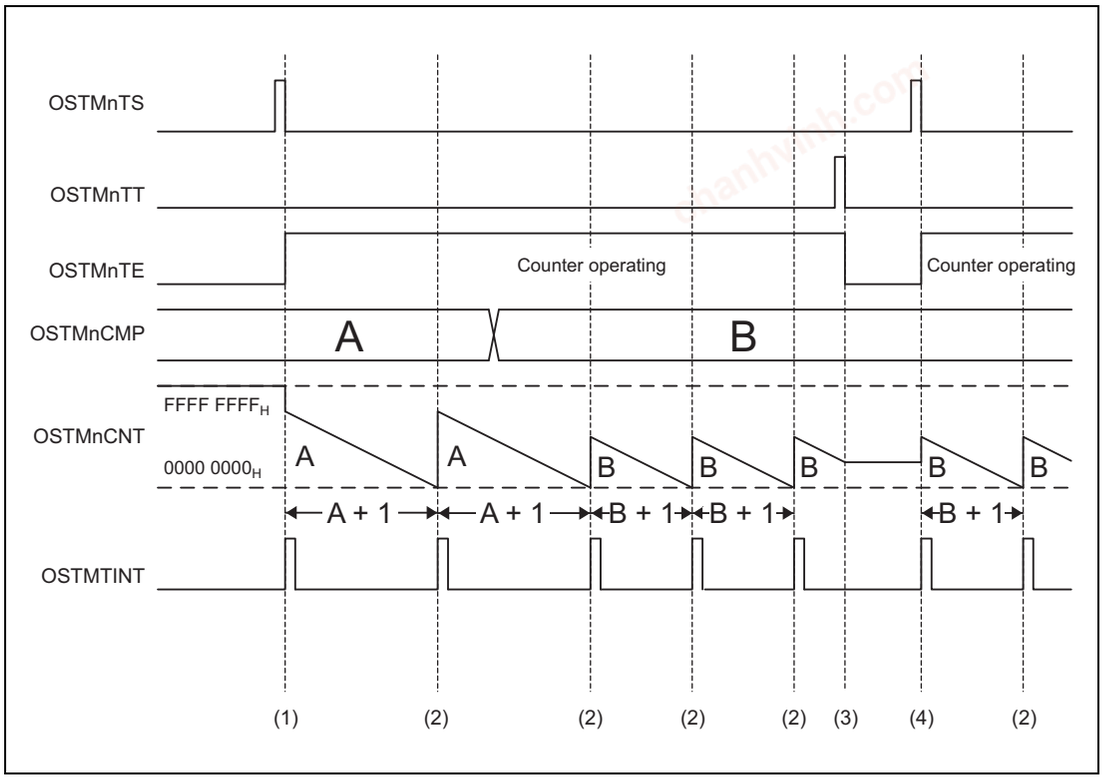


Figure 11.3 Timing Diagram of OSTM in Interval Timer Mode

The timing diagram above shows the following:

(1) The counter starts counting when OSTMnTS.OSTMnTS = 1. The OSTMnTE.OSTMnTE bit is set to indicate enabling of the counter.
The counter starts counting down from the value of OSTMnCMP.
If OSTMnCTL.OSTMnMD0 is 1, OSTMTINT interrupt requests are generated at the start of counting. The OSTMnCNT register contains the current value as the counter.

(2) When the counter reaches 0000 0000$_H$, an OSTMTINT interrupt request is generated. The counter loads the new start value from OSTMnCMP and continues counting down.

(3) When the counter is stopped (OSTMnTT.OSTMnTT = 1), the OSTMnTE.OSTMnTE bit is cleared to indicate disabling of the counter. The counter retains its current value until it is restarted.

(4) When counting is restarted (OSTMnTS.OSTMnTS = 1), the counter loads the new start value from OSTMnCMP and starts counting down.

Forced restart

The counter is forcibly restarted by setting OSTMnTS.OSTMnTS = 1 during counting.

The counter loads the start value from the OSTMnCMP register and continues to count down.

chanhvinh.com

The following figure shows the forced restart of the OS Timer in interval timer mode, with counter-start interrupts enabled (OSTMnCTL.OSTMnMD0 = 1).

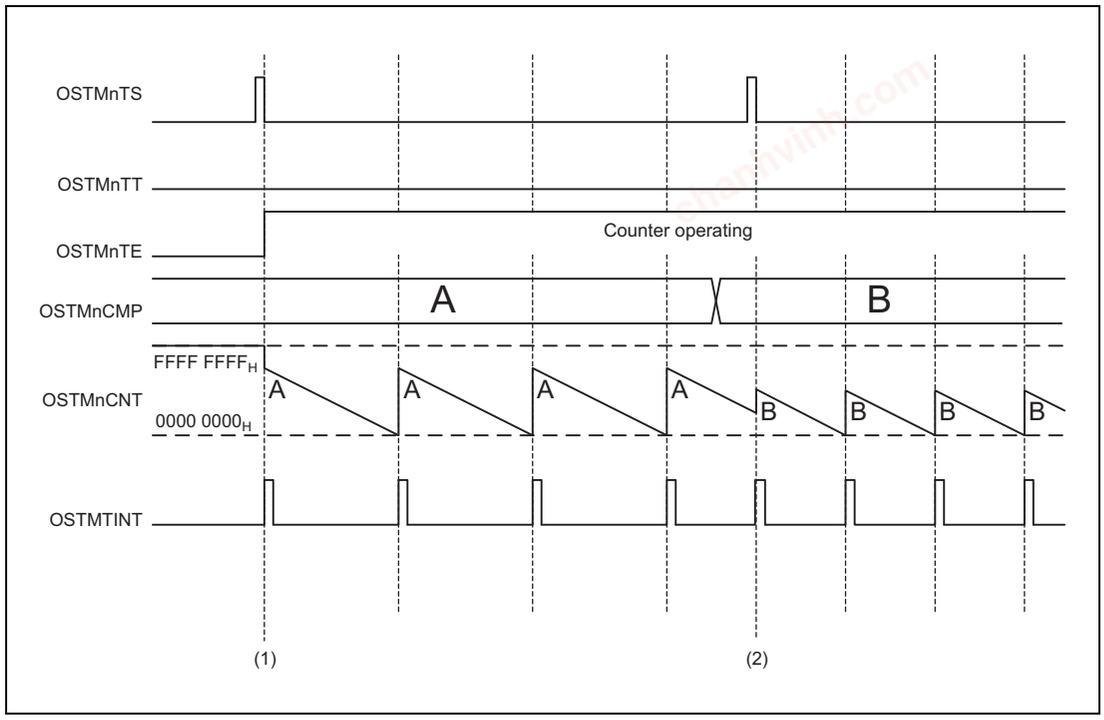


Figure 11.4 Timing Diagram of Forced Restart in Interval Timer Mode

Operations shown in the above timing diagram are as follows.

(1) The counter is started and stopped as described under Figure 11.3, Timing Diagram of OSTM in Interval Timer Mode.

(2) Setting OSTMnTS.OSTMnTS = 1 restarts the counter while counting is in progress (i.e. while OSTMnTE.OSTMnTE = 1).
The counter immediately restarts counting down, starting with the current value of OSTMnCMP.
When OSTMnCTL.OSTMnMD0 = 1, an OSTMTINT interrupt request is generated when counting starts.

chanhvinh.com

11.3.5.2 Operation when OSTMnCMP = 0000 0000$_H$

When OSTMnCMP = 0000 0000$_H$, OSTM behaves as follows.

- When the counter is enabled, the OSTMTINT interrupt request is always set to 1.

The following figure shows operations of OSTM when OSTMnCMP = 0000 0000$_H$, and counter-start interrupts are enabled.

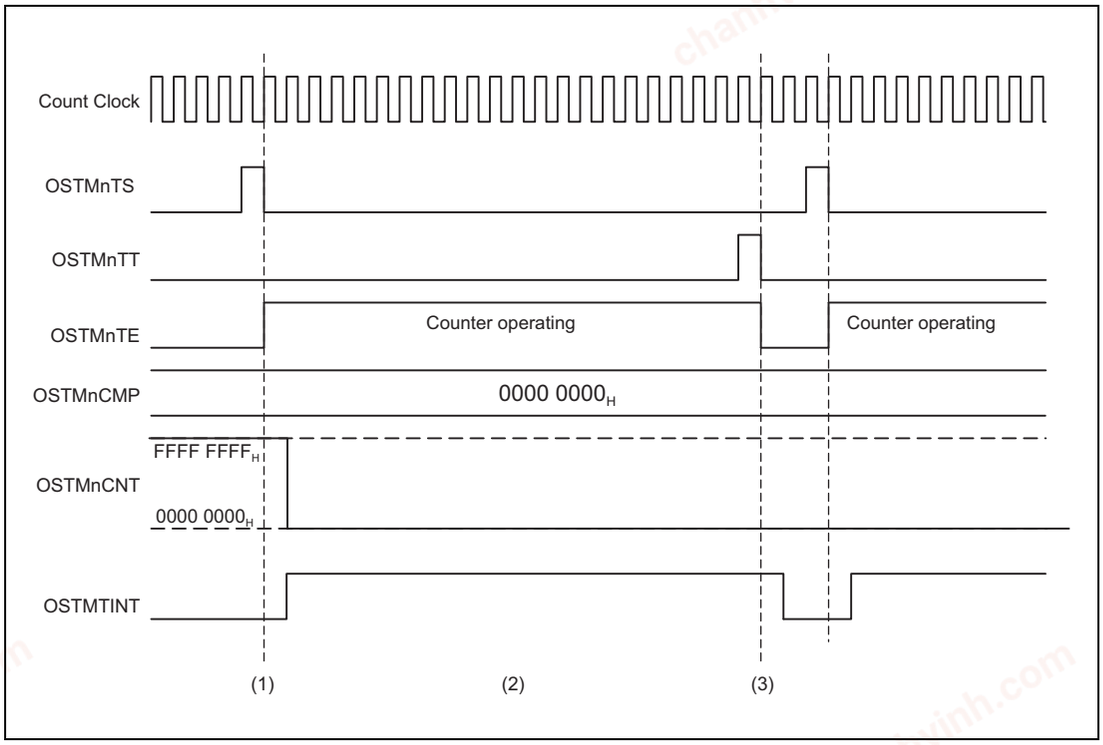


Figure 11.5 Timing Diagram when OSTMnCMP = 0000 0000$_H$ in Interval Timer Mode

The timing diagram above shows the following operations:

(1) The counter is reloaded with the value in OSTMnCMP as soon as it starts counting, so the value 0000 0000$_H$ is retained in OSTMnCMP.

(2) The OSTMTINT interrupt request is continuously asserted.

(3) After the counter stops, the OSTMTINT interrupt request signal is deasserted.

(4) When interrupts on starting of the counter are disabled, no interrupt is generated when counting starts.

chanhvinh.com

11.3.6 Free-Running Comparison Mode

11.3.6.1 Basic Operation in Free-Running Comparison Mode

In free-running comparison mode, the counter counts up from 0000 0000$_H$ to FFFF FFFF$_H$. An OSTMnTINT interrupt request is output when the current value of the counter matches the value of the OSTMnCMP register. The free-running comparison mode is selected by setting the OSTMnCTL.OSTMnMD1 bit to 1.

New values can be written to the OSTMnCMP register at any time.

The following figure shows the basic operation of OSTM in free-run compare mode with the start of counting enabled (OSTMnCTL.OSTMnMD0 = 1).

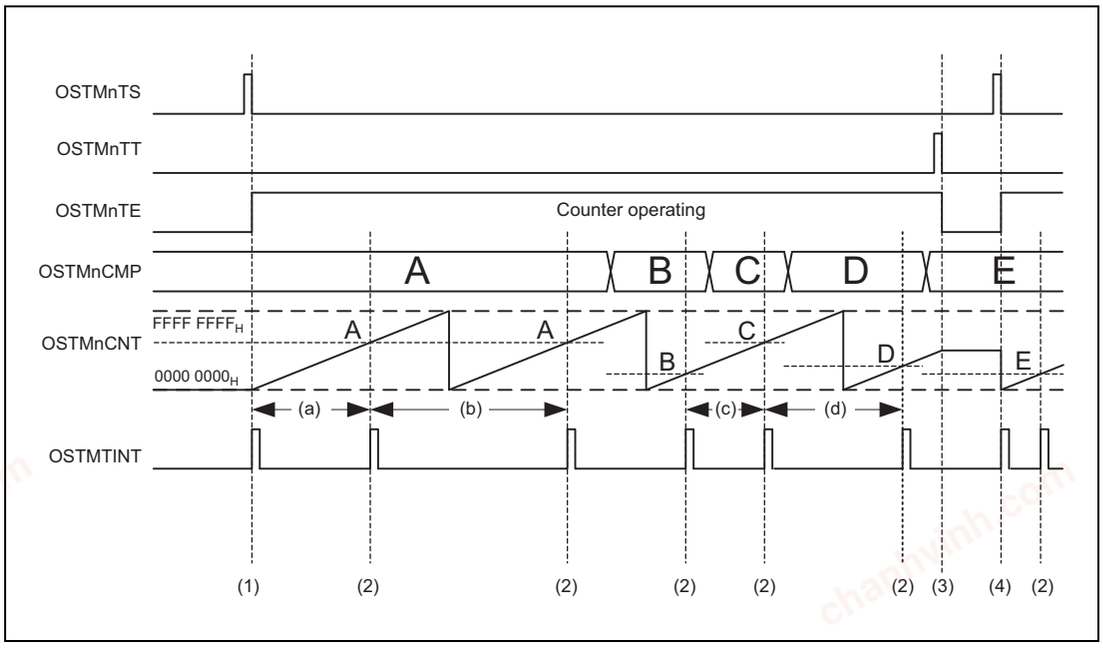


Figure 11.6 Timing Diagram of OSTM in Free-Run Compare Mode

The timing diagram above shows the following:

(1) The counter starts counting when OSTMnTS.OSTMnTS = 1.
The OSTMnTE.OSTMnTE bit is set to indicate enabling of the counter. The counter counts up from 0000 0000$_H$ to FFFF FFFF$_H$. The OSTMnCNT register is the counter, so it contains the current value.
When OSTMnCTL.OSTMnMD0 = 1, an OSTMTINT interrupt request is generated at the start of counting.

(2) When the current counter value matches the value in the OSTMnCMP register, an OSTMTINT interrupt request is generated.

(3) When the counter is stopped (OSTMnTT.OSTMnTT = 1), the OSTMnTE.OSTMnTE bit is cleared to indicate disabling of the counter.
The counter retains its current value until it is restarted.

(4) Counting by the counter restarts from 0000 0000$_H$ when OSTMnTS.OSTMnTS = 1.

chanhvinh.com

OSTMTINT period

The OSTMTINT generation period is different at the start of counting and depends on the old and new compare values if OSTMnCMP is rewritten during operation.

Table 11.11 OSTMTINT Generation Timing

Old Value for Comparison	New Value for Comparison	Counter Value at Time of Rewriting	Period of OSTMTINT Generation	Label in Timing Diagram
Counter starts			(A + 1) × counter clock period	(a)
A	A	No rewriting	(FFFF FFFF$_H$ + 1) × counter clock period	(b)
B	C > B	B < counter value < C	(C – B) × counter clock period	(c)
C	D < C	Counter value > D,C	(FFFF FFFF$_H$ – C + D + 1) × counter clock period	(d)

Forced restart

Forced restarting does not proceed during counting even if the OSTMnTS.OSTMnTS bit is set.

The counter ignores the attempted setting and continues counting.

11.3.6.2 Operation when OSTMnCMP = 0000 0000$_H$

The following figure shows the operation of OSTM when OSTMnCMP = 0000 0000$_H$, and counter-start interrupts are enabled (OSTMnCTL.OSTMnMD0 = 1).

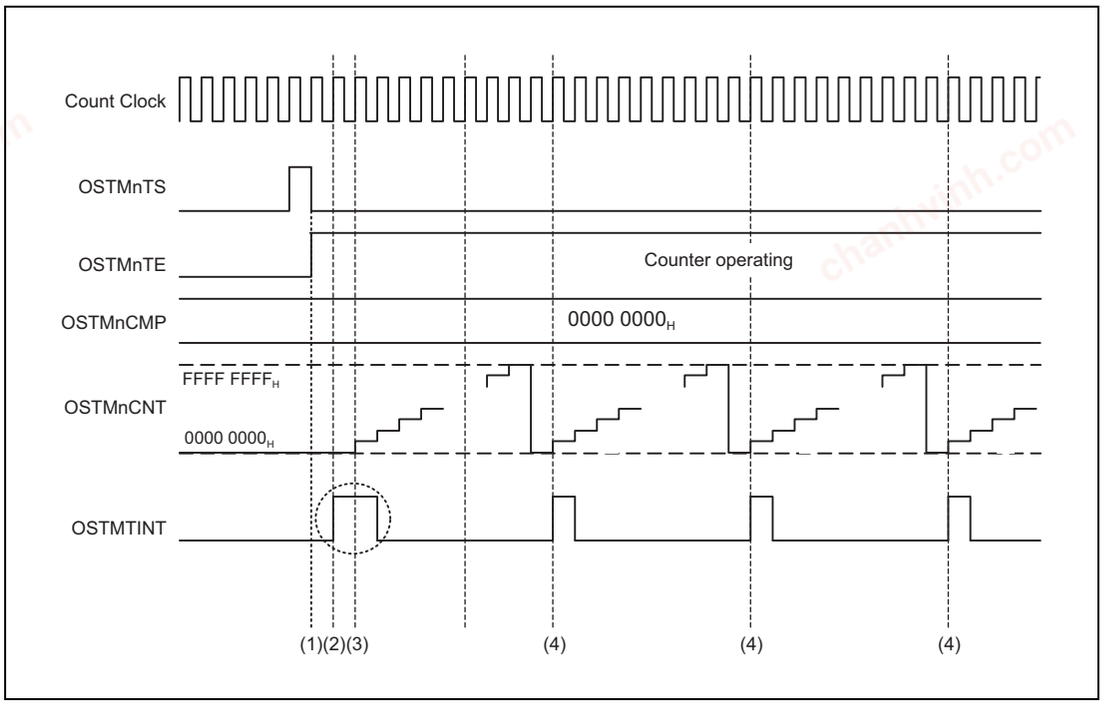


Figure 11.7 Timing Diagram when OSTMnCMP = 0000 0000$_H$ in Free-Run Compare Mode

The timing diagram above shows the following operations.

(1) Once the counter starts, it counts up from 0000 0000$_H$ to FFFF FFFF$_H$.

(2) An OSTMTINT interrupt request is generated when counting starts.

(3) If the current counter value matches OSTMnCMP, an OSTMTINT interrupt request is generated. If OSTMnCMP = 0000 0000$_H$ in the above case, OSTMTINT is generated over two clock cycles.

chanhvinh.com

(4) Every (FFFF FFFF$_H$ + 1) clock cycles the OSTMTINT interrupt request is asserted.

When interrupts on starting of the counter are disabled, no interrupt is generated when counting starts.

chanhvinh.com

12. Watchdog Timer

This LSI includes the watchdog timer, which externally outputs an overflow signal ($\overline{\text{WDTOVF}}$) on overflow of the counter when the value of the counter has not been updated because of a system malfunction. This module can simultaneously generate an internal reset signal for the entire LSI.
This module is a single channel timer that counts up the clock oscillation settling period when the system leaves software standby mode. It can also be used as a general watchdog timer or interval timer.

12.1 Features

- Can be used to ensure the clock oscillation settling time
  This module is used in leaving software standby mode.
- Can switch between watchdog timer mode and interval timer mode.
- Outputs $\overline{\text{WDTOVF}}$ signal in watchdog timer mode
  When the counter overflows in watchdog timer mode, the $\overline{\text{WDTOVF}}$ signal is output externally. It is possible to select whether to reset the LSI internally when this happens. The internal reset signal is used as the power-on reset signal.
- Interrupt generation in interval timer mode
  An interval timer interrupt is generated when the counter overflows.
- Choice of eight counter input clocks
  Eight clocks (P0ϕ × 1 to P0ϕ × 1/16384) that are obtained by dividing the peripheral clock can be selected.

Figure 12.1 shows a block diagram.

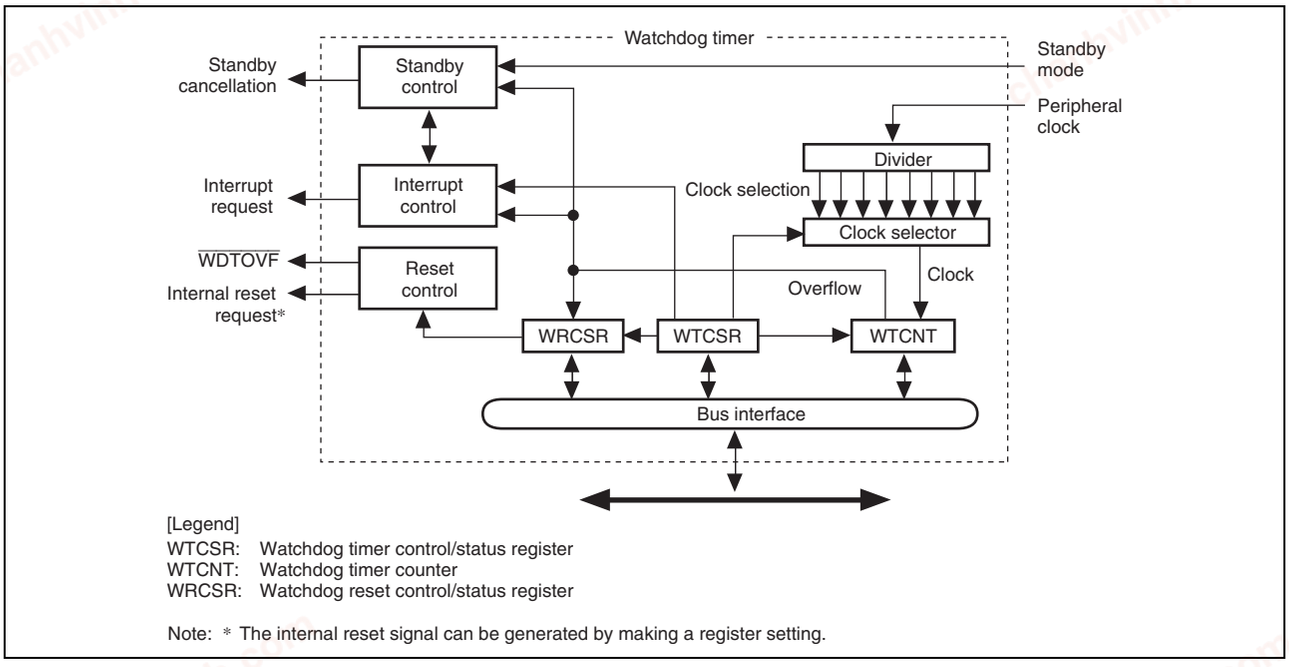


Figure 12.1 Block Diagram

chanhvinh.com

12.2 Input/Output Pin

Table 12.1 shows the pin configuration.

Table 12.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Watchdog timer overflow	$\overline{\text{WDTOVF}}$	Output	Outputs the counter overflow signal in watchdog timer mode

12.3 Register Descriptions

Table 12.2 shows the register configuration.

Table 12.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Watchdog timer counter	WTCNT	R/W	H'00	H'FCFE0002	16*
Watchdog timer control/status register	WTCSR	R/W	H'18	H'FCFE0000	16*
Watchdog reset control/status register	WRCSR	R/W	H'1F	H'FCFE0004	16*

Note: * For the access size, see section 12.3.4, Notes on Register Access.

chanhvinh.com

12.3.1 Watchdog Timer Counter (WTCNT)

WTCNT is an 8-bit readable/writable register that is incremented by cycles of the selected clock signal. When an overflow occurs, it generates a watchdog timer overflow signal ($\overline{\text{WDTOVF}}$) in watchdog timer mode and an interrupt in interval timer mode.

Use 16-bit access to write to WTCNT, writing H'5A in the upper byte. Use 8-bit access to read from WTCNT.

Note: The method for writing to WTCNT differs from that for other registers to prevent erroneous writes. See section 12.3.4, Notes on Register Access for details.

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

12.3.2 Watchdog Timer Control/Status Register (WTCSR)

WTCSR is an 8-bit readable/writable register composed of bits to select the clock used for the count, overflow flags, and timer enable bit.

When used to count the clock oscillation settling time for canceling software standby mode, it retains its value after counter overflow.

Use 16-bit access to write to WTCSR, writing H'A5 in the upper byte. Use 8-bit access to read from WTCSR.

Note: The method for writing to WTCSR differs from that for other registers to prevent erroneous writes. See section 12.3.4, Notes on Register Access for details.

Bit:	7	6	5	4	3	2	1	0
	IOVF	WT/$\overline{\text{IT}}$	TME	-	-	CKS[2:0]		
Initial value:	0	0	0	1	1	0	0	0
R/W:	R/(W)	R/W	R/W	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	IOVF	0	R/(W)	Interval Timer Overflow Indicates that WTCNT has overflowed in interval timer mode. This flag is not set in watchdog timer mode. 0: No overflow 1: WTCNT overflow in interval timer mode [Clearing condition] • When 0 is written to IOVF after reading IOVF
6	WT/$\overline{\text{IT}}$	0	R/W	Timer Mode Select Selects whether to use this module as a watchdog timer or an interval timer. 0: Use as interval timer 1: Use as watchdog timer Note: When the WTCNT overflows in watchdog timer mode, the $\overline{\text{WDTOVF}}$ signal is output externally. If this bit is modified when this module is running, the up-count may not be performed correctly.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
5	TME	0	R/W	Timer Enable Starts and stops timer operation. Clear this bit to 0 when using this module in software standby mode or when changing the clock frequency. 0: Timer disabled Count-up stops and WTCNT value is retained 1: Timer enabled
4,3	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
2 to 0	CKS[2:0]	000	R/W	Clock Select These bits select the clock to be used for the WTCNT count from the eight types obtainable by dividing the peripheral clock (P0ϕ). The overflow period that is shown inside the parenthesis in the table is the value when the peripheral clock (P0ϕ) is 33.33 MHz.

Bits 2 to 0	Clock Ratio	Overflow Cycle
000:	1 x P0ϕ	7.7 µs
001:	1/64 x P0ϕ	490 µs
010:	1/128 x P0ϕ	979 µs
011:	1/256 x P0ϕ	2.0 ms
100:	1/512 x P0ϕ	3.9 ms
101:	1/1024 x P0ϕ	7.8 ms
110:	1/4096 x P0ϕ	31 ms
111:	1/16384 x P0ϕ	125 ms

Note: If bits CKS[2:0] are modified when this module is running, the up-count may not be performed correctly. Ensure that these bits are modified only when this module is not running.

chanhvinh.com

12.3.3 Watchdog Reset Control/Status Register (WRCSR)

WRCSR is an 8-bit readable/writable register that controls output of the internal reset signal generated by watchdog timer counter (WTCNT) overflow.

Note: The method for writing to WRCSR differs from that for other registers to prevent erroneous writes. See section 12.3.4, Notes on Register Access for details.

Bit:	7	6	5	4	3	2	1	0
	WOVF	RSTE	-	-	-	-	-	-
Initial value:	0	0	0	1	1	1	1	1
R/W:	R/(W)	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	WOVF	0	R/(W)	Watchdog Timer Overflow Indicates that the WTCNT has overflowed in watchdog timer mode. This bit is not set in interval timer mode. 0: No overflow 1: WTCNT has overflowed in watchdog timer mode [Clearing condition] • When 0 is written to WOVF after reading WOVF
6	RSTE	0	R/W	Reset Enable Selects whether to generate a signal to reset the LSI internally if WTCNT overflows in watchdog timer mode. In interval timer mode, this setting is ignored. 0: Not reset when WTCNT overflows* 1: Reset when WTCNT overflows Note: * LSI not reset internally, but WTCNT and WTCSR reset within this module.
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4 to 0	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.

chanhvinh.com

12.3.4 Notes on Register Access

The watchdog timer counter (WTCNT), watchdog timer control/status register (WTCSR), and watchdog reset control/status register (WRCSR) are more difficult to write to than other registers. The procedures for reading or writing to these registers are given below.

(1) Writing to WTCNT and WTCSR

These registers must be written by a 16-bit transfer instruction. They cannot be written by an 8- or 32-bit transfer instruction.

When writing to WTCNT, set the upper byte to H'5A and transfer the lower byte as the write data, as shown in Figure 12.2. When writing to WTCSR, set the upper byte to H'A5 and transfer the lower byte as the write data. This transfer procedure writes the lower byte data to WTCNT or WTCSR.

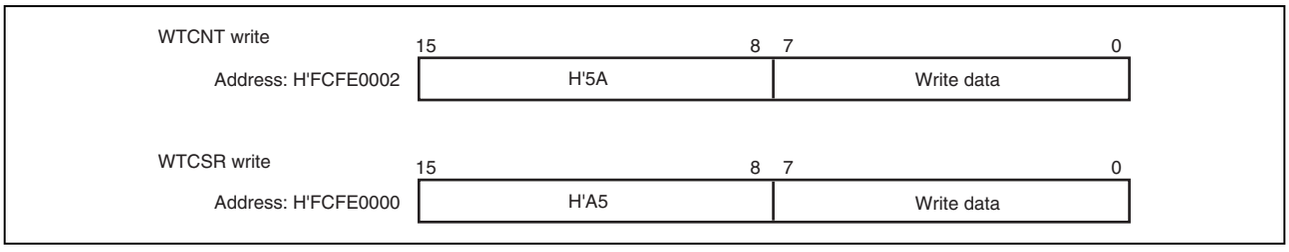


Figure 12.2 Writing to WTCNT and WTCSR

(2) Writing to WRCSR

WRCSR must be written by a 16-bit access to address H'FCFE0004. It cannot be written by 8- or 32-bit transfer instructions.

Procedures for writing 0 to WOVF (bit 7) and for writing to RSTE (bit 6) are different, as shown in Figure 12.3.

To write 0 to the WOVF bit, the write data must be H'A5 in the upper byte and H'00 in the lower byte. This clears the WOVF bit to 0. The RSTE bit is not affected. To write to the RSTE bit, the upper byte must be H'5A and the lower byte must be the write data. The value of bit 6 of the lower byte is transferred to the RSTE bit. The WOVF bit is not affected.

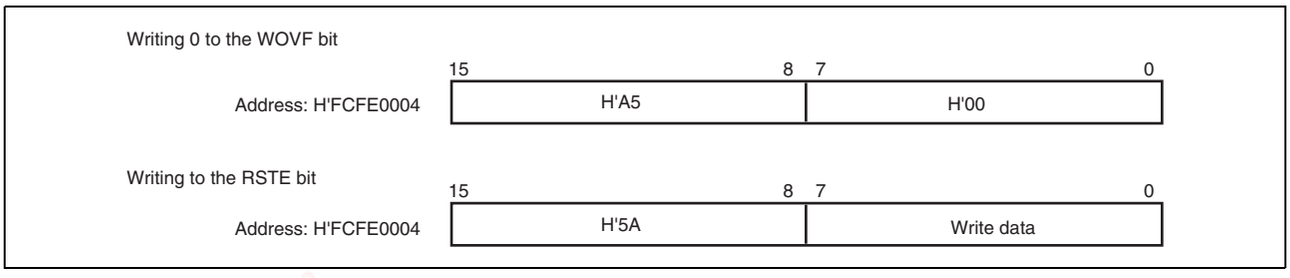


Figure 12.3 Writing to WRCSR

chanhvinh.com

(3) Reading from WTCNT, WTCSR, and WRCSR

WTCNT, WTCSR, and WRCSR are read in a method similar to other registers. WTCSR is allocated to address H'FCFE0000, WTCNT to address H'FCFE0002, and WRCSR to address H'FCFE0004. Eight-bit transfer instructions must be used for reading from these registers.

chanhvinh.com

12.4 Usage

12.4.1 Canceling Software Standby Mode

This module can be used to cancel software standby mode with an interrupt such as an NMI interrupt. The procedure is described below. (This module does not operate when resets are used for canceling, so keep the $\overline{\text{RES}}$ pin low until clock oscillation settles.)

1. Before making a transition to software standby mode, always clear the TME bit in WTCSR to 0. When the TME bit is 1, an erroneous reset or interval timer interrupt may be generated when the count overflows.
2. Set the type of count clock used in the CKS[2:0] bits in WTCSR and the initial value of the counter in WTCNT. These values should ensure that the time till count overflow is equal to or longer than the clock oscillation settling time.
3. After setting the STBY and DEEP bits of the standby control register 1 (STBCR1: see section 42, Power-Down Modes) to 1 and 0 respectively, the execution of a WFI instruction puts the system in software standby mode and clock operation then stops.
4. This module starts counting by detecting the edge change of the NMI signal.
5. When the module count overflows, the clock pulse generator starts supplying the clock and this LSI resumes operation. The WOVF flag in WRCSR is not set when this happens.

12.4.2 Using Watchdog Timer Mode

1. Set the WT/$\overline{\text{IT}}$ bit in WTCSR to 1, the type of count clock in the CKS[2:0] bits in WTCSR, whether this LSI is to be reset internally or not in the RSTE bit in WRCSR, and the initial value of the counter in WTCNT.
2. Set the TME bit in WTCSR to 1 to start the count in watchdog timer mode.
3. While operating in watchdog timer mode, rewrite the counter periodically to H'00 to prevent the counter from overflowing.
4. When the counter overflows, this module sets the WOVF flag in WRCSR to 1, and the $\overline{\text{WDTOVF}}$ signal is output externally (Figure 12.4). The $\overline{\text{WDTOVF}}$ signal can be used to reset the system. The $\overline{\text{WDTOVF}}$ signal is output for 64 × P0ϕ clock cycles.
5. If the RSTE bit in WRCSR is set to 1, a signal to reset the inside of this LSI can be generated simultaneously with the $\overline{\text{WDTOVF}}$ signal. The internal reset signal is output for 128 × P0ϕ clock cycles.
6. When an overflow reset of this module is generated simultaneously with a reset input on the $\overline{\text{RES}}$ pin, the $\overline{\text{RES}}$ pin reset takes priority, and the WOVF bit in WRCSR is cleared to 0.

chanhvinh.com

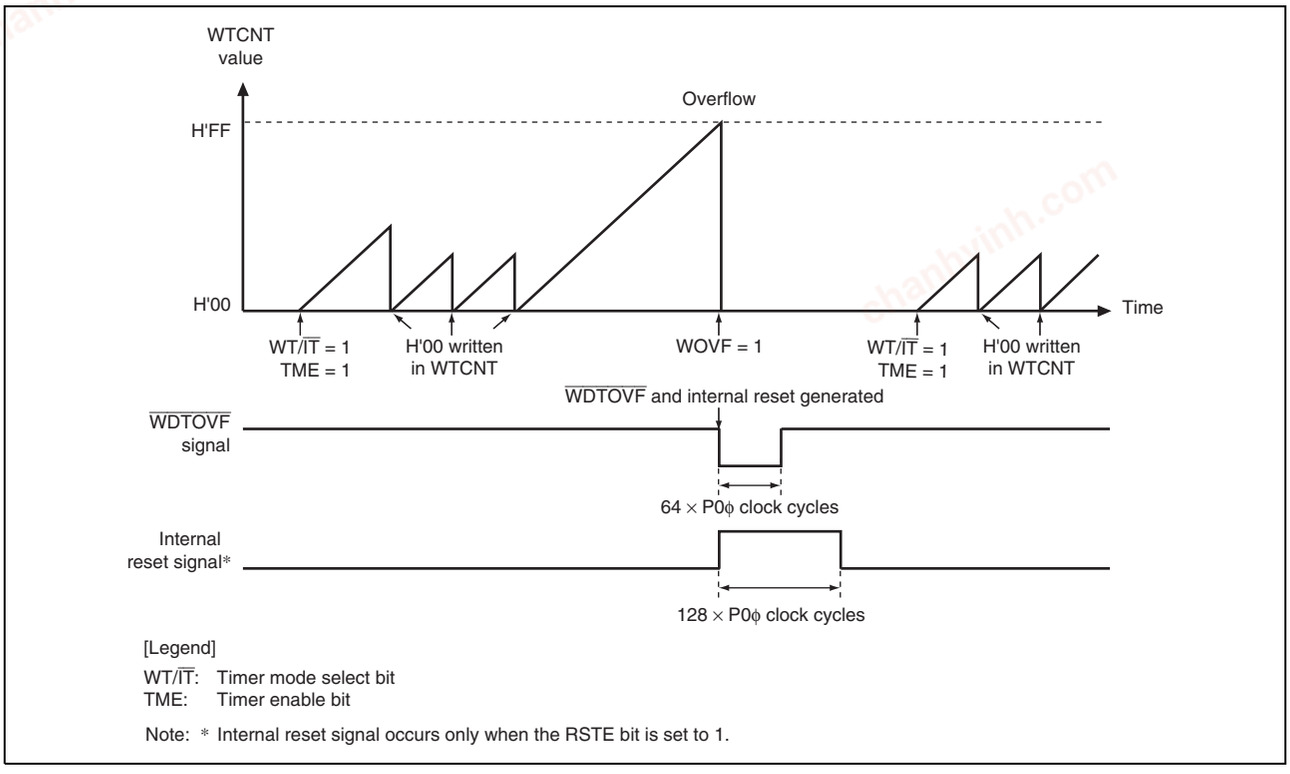


Figure 12.4 Operation in Watchdog Timer Mode

12.4.3 Using Interval Timer Mode

When operating in interval timer mode, interval timer interrupts are generated at every overflow of the counter. This enables interrupts to be generated at set periods.

1. Clear the WT/$\overline{IT}$ bit in WTCSR to 0, set the type of count clock in the CKS[2:0] bits in WTCSR, and set the initial value of the counter in WTCNT.
2. Set the TME bit in WTCSR to 1 to start the count in interval timer mode.
3. When the counter overflows, this module sets the IOVF bit in WTCSR to 1 and an interval timer interrupt request is sent to the interrupt controller. The counter then resumes counting.

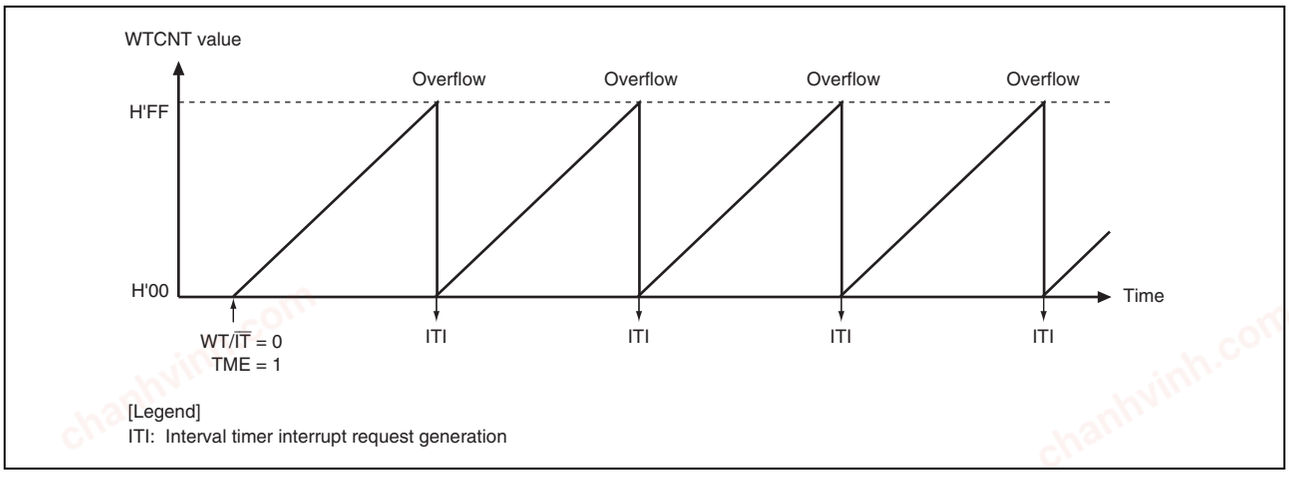


Figure 12.5 Operation in Interval Timer Mode

chanhvinh.com

12.5 Usage Notes

Pay attention to the following points when using this module in either the interval timer or watchdog timer mode.

12.5.1 Timer Variation

After timer operation has started, the period from the power-on reset point to the first count up timing of WTCNT varies depending on the time period that is set by the TME bit of WTCSR. The shortest such time period is thus one cycle of the peripheral clock, P0ϕ, while the longest is the result of frequency division according to the value in the CKS[2:0] bits. The timing of subsequent incrementation is in accord with the selected frequency division ratio. Accordingly, this time difference is referred to as timer variation.

This also applies to the timing of the first incrementation after WTCNT has been written to during timer operation.

12.5.2 Prohibition against Setting H'FF to WTCNT

When the value in WTCNT reaches H'FF, this module assumes that an overflow has occurred. Accordingly, when H'FF is set in WTCNT, an interval timer interrupt or reset will occur immediately, regardless of the current clock selection by the CKS[2:0] bits.

12.5.3 Interval Timer Overflow Flag

When the value in WTCNT is H'FF, the IOVF flag in WTCSR cannot be cleared.

Only clear the IOVF flag when the value in WTCNT has either become H'00 or been changed to a value other than H'FF.

12.5.4 System Reset by $\overline{\text{WDTOVF}}$ Signal

If the $\overline{\text{WDTOVF}}$ signal is input to the $\overline{\text{RES}}$ pin of this LSI, this LSI cannot be initialized correctly.

Avoid input of the $\overline{\text{WDTOVF}}$ signal to the $\overline{\text{RES}}$ pin of this LSI through glue logic circuits. To reset the entire system with the $\overline{\text{WDTOVF}}$ signal, use the circuit shown in Figure 12.6.

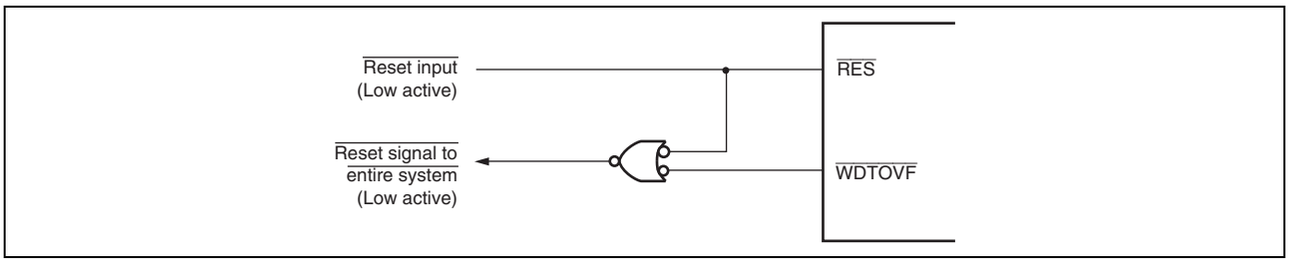


Figure 12.6 Example of System Reset Circuit Using $\overline{\text{WDTOVF}}$ Signal

12.5.5 Internal Reset in Watchdog Timer Mode

When an internal reset is generated due to an overflow of the watchdog timer counter (WTCNT) in watchdog timer mode, the watchdog reset control/status register (WRCSR) is not initialized, so the WOVF bit retains the value 1. As long as the WOVF bit is 1, an internal reset will not be generated even if the WTCNT overflows again.

chanhvinh.com

13. Realtime Clock

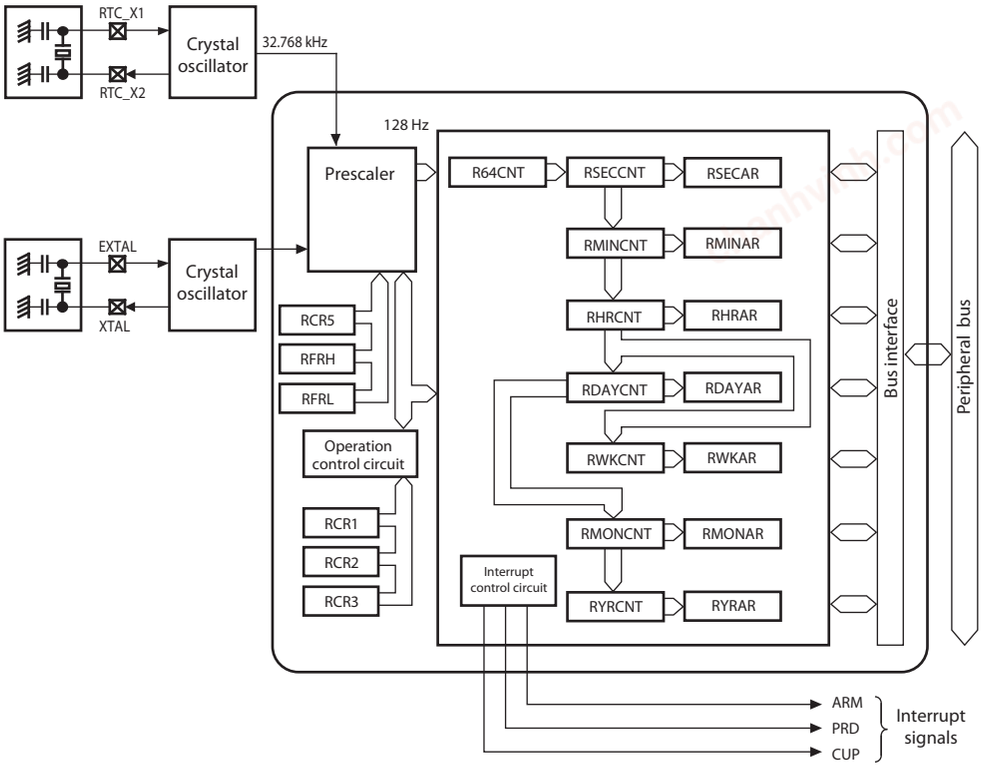
This LSI has a realtime clock and a 32.768-kHz crystal oscillator.

13.1 Features

- Clock and calendar functions (BCD format): Seconds, minutes, hours, day of the week, day, month, and year.
- 1-Hz to 64-Hz timer (binary format)
  64-Hz counter indicates the state of the divider circuit between 64 Hz and 1 Hz
- Start/stop function
- 30-second adjust function
- Alarm interrupt: Comparison with seconds, minutes, hours, day of the week, day, month, or year can be selected as a condition for the alarm interrupt
- Periodic interrupt: the interrupt cycle may be 1/64 second, 1/16 second, 1/4 second, 1/2 second, 1 second, or 2 seconds
- Carry interrupt: a carry interrupt indicates that a second counter carry is generated or a 64-Hz counter carry is generated reading the 64-Hz counter
- Automatic leap year adjustment
- The external clock signal input for internal operation or the external clock signal input dedicated to the clock operation can be selected as the operating clock signal for the clock function.
- Recovery from deep standby mode can be performed by an alarm interrupt.

chanhvinh.com

Figure 13.1 shows the block diagram.



[Legend]

RSECCNT:	Second counter	RSECAR:	Second alarm register
RMINCNT:	Minute counter	RMINAR:	Minute alarm register
RHRCNT:	Hour counter	RHRAR:	Hour alarm register
RWKCNT:	Day of week counter	RWKAR:	Day of week alarm register
RDAYCNT:	Day counter	RDAYAR:	Day alarm register
RMONCNT:	Month counter	RMONAR:	Month alarm register
RYRCNT:	Year counter	RYRAR:	Year alarm register
R64CNT:	64-Hz counter	RCR1:	Control register 1
RFRH/L:	Frequency register	RCR2:	Control register 2
		RCR3:	Control register 3
		RCR5:	Control register 5

Figure 13.1 Block Diagram

chanhvinh.com

13.2 Input/Output Pin

Table 13.1 shows the pin configuration.

Table 13.1 Pin Configuration

Pin Name	Symbol	I/O	Description
Realtime clock crystal resonator pin/external clock	RTC_X1	Input	Connects a 32.768-kHz crystal resonator for this module. External clock can be input to the RTC_X1 pin.
	RTC_X2	Output	
Internal clock crystal resonator/external clock	EXTAL	Input	Connects crystal resonator used for internal operation. For details, see section 6, Clock Pulse Generator.
	XTAL	Output	

13.3 Register Descriptions

Table 13.2 shows the register configuration.

Table 13.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
64-Hz counter	R64CNT	R	H'xx	H'FCFF1000	8
Second counter	RSECCNT	R/W	H'xx	H'FCFF1002	8
Minute counter	RMINCNT	R/W	H'xx	H'FCFF1004	8
Hour counter	RHRCNT	R/W	H'xx	H'FCFF1006	8
Day of week counter	RWKCNT	R/W	H'0x	H'FCFF1008	8
Day counter	RDAYCNT	R/W	H'xx	H'FCFF100A	8
Month counter	RMONCNT	R/W	H'xx	H'FCFF100C	8
Year counter	RYRCNT	R/W	H'xxxx	H'FCFF100E	16
Second alarm register	RSECAR	R/W	H'xx	H'FCFF1010	8
Minute alarm register	RMINAR	R/W	H'xx	H'FCFF1012	8
Hour alarm register	RHRAR	R/W	H'xx	H'FCFF1014	8
Day of week alarm register	RWKAR	R/W	H'xx	H'FCFF1016	8
Day alarm register	RDAYAR	R/W	H'xx	H'FCFF1018	8
Month alarm register	RMONAR	R/W	H'xx	H'FCFF101A	8
Year alarm register	RYRAR	R/W	H'xxxx	H'FCFF1020	16
Control register 1	RCR1	R/W	H'xx	H'FCFF101C	8
Control register 2	RCR2	R/W	H'01	H'FCFF101E	8
Control register 3	RCR3	R/W	H'x0	H'FCFF1024	8
Control register 5	RCR5	R/W	H'0x	H'FCFF1026	8
Frequency register	RFRH	R/W	H'xxxx	H'FCFF102A	16
	RFRL	R/W	H'xxxx	H'FCFF102C	16

chanhvinh.com

13.3.1 64-Hz Counter (R64CNT)

R64CNT indicates the state of the divider circuit between 64 Hz and 1 Hz.

Reading this register, when carry from 128-Hz divider stage is generated, sets the CF bit in the control register 1 (RCR1) to 1, which indicates that the carrying and reading the 64-Hz counter are performed at the same time. In this case, the R64CNT should be read again after writing 0 to the CF bit in RCR1 since the read value is not valid.

Setting the RESET or ADJ bit in the control register 2 (RCR2) to 1 initializes the divider circuit and the R64CNT.

Bit:	7	6	5	4	3	2	1	0
	-	1Hz	2Hz	4Hz	8Hz	16Hz	32Hz	64Hz
Initial value:	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W:	R	R	R	R	R	R	R	R

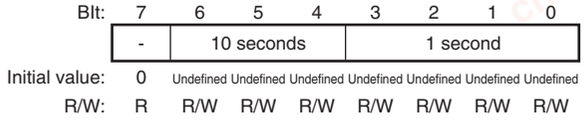
Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	1 Hz	Undefined	R	Indicate the state of the divider circuit between 64 Hz and 1 Hz.
5	2 Hz	Undefined	R	
4	4 Hz	Undefined	R	
3	8 Hz	Undefined	R	
2	16 Hz	Undefined	R	
1	32 Hz	Undefined	R	
0	64 Hz	Undefined	R	

chanhvinh.com

13.3.2 Second Counter (RSECCNT)

RSECCNT is the counter used for setting/counting the BCD-coded second value. The count operation is performed by a carry for each second of the 64-Hz counter.

The assignable range is from 00 through 59 (practically in BCD); otherwise an operation error will occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.

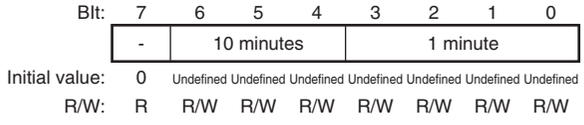


Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	10 seconds	Undefined	R/W	Counting Ten's Position of Seconds Counts from 0 to 5 for 60-seconds counting.
3 to 0	1 second	Undefined	R/W	Counting One's Position of Seconds Counts from 0 to 9, one number per second. When a carry is generated, 1 is added to the ten's position.

13.3.3 Minute Counter (RMINCNT)

RMINCNT is the counter used for setting/counting the BCD-coded minute value. The count operation is performed by a carry for each minute of the second counter.

The assignable range is from 00 through 59 (practically in BCD); otherwise an operation error will occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.



Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0.The write value should always be 0.
6 to 4	10 minutes	Undefined	R/W	Counting Ten's Position of Minutes Counts from 0 to 5 for 60-minutes counting.
3 to 0	1 minute	Undefined	R/W	Counting One's Position of Minutes Counts from 0 to 9, one number per minute. When a carry is generated, 1 is added to the ten's position.

chanhvinh.com

13.3.4 Hour Counter (RHRCNT)

RHRCNT is the counter used for setting/counting the BCD-coded hour value. The count operation is performed by a carry for each 1 hour of the minute counter.

The assignable range is from 00 through 23 (practically in BCD); otherwise an operation error will occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.

Bit:	7	6	5	4	3	2	1	0
	-	-	10 hours		1 hour			
Initial value:	0	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	10 hours	Undefined	R/W	Counting Ten's Position of Hours Counts from 0 to 2 for ten's position of hours.
3 to 0	1 hour	Undefined	R/W	Counting One's Position of Hours Counts from 0 to 9, one number per hour. When a carry is generated, 1 is added to the ten's position.

13.3.5 Day of Week Counter (RWKCNT)

RWKCNT is the counter used for setting/counting the BCD-coded day-of-week value. The count operation is performed by a carry for each day of the hour counter.

The assignable range is from 0 through 6 (practically in BCD); otherwise an operation error will occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	Day		
Initial value:	0	0	0	0	0	Undefined	Undefined	Undefined
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	Day	Undefined	R/W	Day-of-Week Counting Day-of-week is indicated with a binary code. 000: Sunday 001: Monday 010: Tuesday 011: Wednesday 100: Thursday 101: Friday 110: Saturday 111: Reserved (setting prohibited)

chanhvinh.com

13.3.6 Day Counter (RDAYCNT)

RDAYCNT is the counter used for setting/counting the BCD-coded day value. The count operation is performed by a carry for each day of the hour counter.

The assignable range is from 01 through 31 (practically in BCD); otherwise an operation error will occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.

The assignable range changes depending on the month and in leap years. Confirm the correct setting. A leap year is determined by checking if the year counter (RYRCNT) value is divisible by 400, 100, and 4.

Bit:	7	6	5	4	3	2	1	0
	-	-	10 days		1 day			
Initial value:	0	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	10 days	Undefined	R/W	Counting Ten's Position of Days
3 to 0	1 day	Undefined	R/W	Counting One's Position of Days Counts from 0 to 9, one number per day. When a carry is generated, 1 is added to the ten's position.

13.3.7 Month Counter (RMONCNT)

RMONCNT is the counter used for setting/counting the BCD-coded month value. The count operation is performed by a carry for each month of the date counter.

The assignable range is from 01 through 12 (practically in BCD); otherwise an operation error will occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	10 months	1 month			
Initial value:	0	0	0	Undefined	Undefined	Undefined	Undefined	Undefined
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

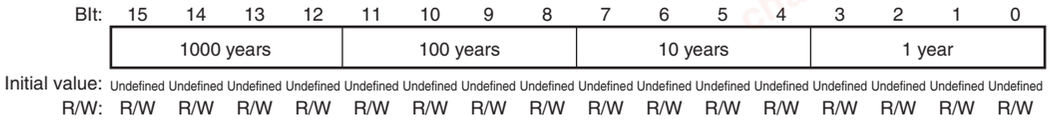
Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	10 months	Undefined	R/W	Counting Ten's Position of Months
3 to 0	1 month	Undefined	R/W	Counting One's Position of Months Counts from 0 to 9, one number per month. When a carry is generated, 1 is added to the ten's position.

chanhvinh.com

13.3.8 Year Counter (RYRCNT)

RYRCNT is the counter used for setting/counting the BCD-coded year value. The count operation is performed by a carry for each year of the month counter.

The assignable range is from 0000 through 9999 (practically in BCD); otherwise an operation error will occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.

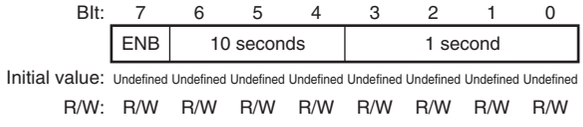


Bit	Bit Name	Initial Value	R/W	Description
15 to 12	1000 years	Undefined	R/W	Counting Thousand's Position of Years
11 to 8	100 years	Undefined	R/W	Counting Hundred's Position of Years
7 to 4	10 years	Undefined	R/W	Counting Ten's Position of Years
3 to 0	1 year	Undefined	R/W	Counting One's Position of Years

13.3.9 Second Alarm Register (RSECAR)

RSECAR is an alarm register corresponding to the BCD-coded second counter RSECCNT. When the ENB bit is set to 1, a comparison with the RSECCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The assignable range is from 00 through 59 + ENB bits (practically in BCD); otherwise an operation error will occur.



Bit	Bit Name	Initial Value	R/W	Description
7	ENB	Undefined	R/W	When this bit is set to 1, a comparison with the RSECCNT value is performed.
6 to 4	10 seconds	Undefined	R/W	Ten's position of seconds setting value
3 to 0	1 second	Undefined	R/W	One's position of seconds setting value

chanhvinh.com

13.3.10 Minute Alarm Register (RMINAR)

RMINAR is an alarm register corresponding to the BCD-coded minute counter RMINCNT. When the ENB bit is set to 1, a comparison with the RMINCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The assignable range is from 00 through 59 + ENB bits (practically in BCD); otherwise an operation error will occur.

Bit:	7	6	5	4	3	2	1	0
	ENB	10 minutes			1 minute			
Initial value:	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	Undefined	R/W	When this bit is set to 1, a comparison with the RMINCNT value is performed.
6 to 4	10 minutes	Undefined	R/W	Ten's position of minutes setting value
3 to 0	1 minute	Undefined	R/W	One's position of minutes setting value

13.3.11 Hour Alarm Register (RHRAR)

RHRAR is an alarm register corresponding to the BCD-coded hour counter RHRCNT. When the ENB bit is set to 1, a comparison with the RHRCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The assignable range is from 00 through 23 + ENB bits (practically in BCD); otherwise an operation error will occur.

Bit:	7	6	5	4	3	2	1	0
	ENB	-	10 hours		1 hour			
Initial value:	Undefined	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	Undefined	R/W	When this bit is set to 1, a comparison with the RHRCNT value is performed.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5, 4	10 hours	Undefined	R/W	Ten's position of hours setting value
3 to 0	1 hour	Undefined	R/W	One's position of hours setting value

chanhvinh.com

13.3.12 Day of Week Alarm Register (RWKAR)

RWKAR is an alarm register corresponding to the BCD-coded day of week counter RWKCNT. When the ENB bit is set to 1, a comparison with the RWKCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The assignable range is from 0 through 6 + ENB bits (practically in BCD); otherwise an operation error will occur.

Bit:	7	6	5	4	3	2	1	0
	ENB	-	-	-	-	Day		
Initial value:	Undefined	0	0	0	0	Undefined	Undefined	Undefined
R/W:	R/W	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	Undefined	R/W	When this bit is set to 1, a comparison with the RWKCNT value is performed.
6 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	Day	Undefined	R/W	Day of Week Setting Value 000: Sunday 001: Monday 010: Tuesday 011: Wednesday 100: Thursday 101: Friday 110: Saturday 111: Reserved (setting prohibited)

13.3.13 Day Alarm Register (RDAYAR)

RDAYAR is an alarm register corresponding to the BCD-coded day counter RDAYCNT. When the ENB bit is set to 1, a comparison with the RDAYCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The assignable range is from 01 through 31 + ENB bits (practically in BCD); otherwise an operation error will occur.

Bit:	7	6	5	4	3	2	1	0
	ENB	-	10 days		1 day			
Initial value:	Undefined	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

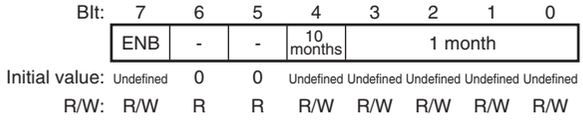
Bit	Bit Name	Initial Value	R/W	Description
7	ENB	Undefined	R/W	When this bit is set to 1, a comparison with the RDAYCNT value is performed.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5, 4	10 days	Undefined	R/W	Ten's position of days setting value
3 to 0	1 day	Undefined	R/W	One's position of days setting value

chanhvinh.com

13.3.14 Month Alarm Register (RMONAR)

RMONAR is an alarm register corresponding to the BCD-coded month counter RMONCNT. When the ENB bit is set to 1, a comparison with the RMONCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

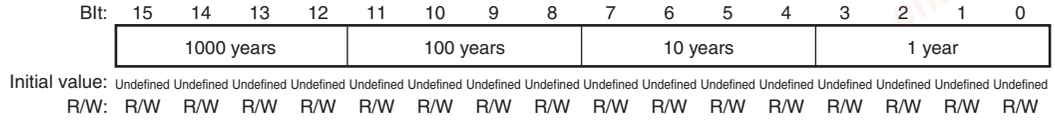
The assignable range is from 01 through 12 + ENB bits (practically in BCD); otherwise an operation error will occur.



Bit	Bit Name	Initial Value	R/W	Description
7	ENB	Undefined	R/W	When this bit is set to 1, a comparison with the RMONCNT value is performed.
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	10 months	Undefined	R/W	Ten's position of months setting value
3 to 0	1 month	Undefined	R/W	One's position of months setting value

13.3.15 Year Alarm Register (RYRAR)

RYRAR is an alarm register corresponding to the BCD-coded year counter RYRCNT. The assignable range is from 0000 through 9999 (practically in BCD); otherwise an operation error will occur.



Bit	Bit Name	Initial Value	R/W	Description
15 to 12	1000 years	Undefined	R/W	Thousand's position of years setting value
11 to 8	100 years	Undefined	R/W	Hundred's position of years setting value
7 to 4	10 years	Undefined	R/W	Ten's position of years setting value
3 to 0	1 year	Undefined	R/W	One's position of years setting value

chanhvinh.com

13.3.16 Control Register 1 (RCR1)

RCR1 is a register that affects carry flags and alarm flags. It also selects whether to generate interrupts for each flag. The CF flag remains undefined until the divider circuit is reset (the RESET and ADJ bits in RCR2 are set to 1). When using the CF flag, make sure to reset the divider circuit beforehand.

The AF flag remains undefined until the value is set to an alarm register and a counter. When using the AF flag, make sure to set the alarm register and counter beforehand.

Bit:	7	6	5	4	3	2	1	0
	CF	-	-	CIE	AIE	-	-	AF
Initial value:	Undefined	0	0	0	0	0	0	Undefined
R/W:	R/W	R	R	R/W	R/W	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	CF	Undefined	R/W	Carry Flag Status flag that indicates that a carry has occurred. CF is set to 1 when a second counter carry is generated or a 64-Hz counter carry is generated while reading the 64-Hz counter A value read from the count register at this time cannot be guaranteed; another read is required. 0: No carry of second counter or no carry of 64-Hz counter while reading 64-Hz counter [Clearing condition] When 0 is written to CF 1: Carry of second counter or carry of 64-Hz counter while reading 64-Hz counter [Setting condition] When a second counter carry is generated or a 64-Hz counter carry is generated while reading the 64-Hz counter, or 1 is written to CF
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	CIE	0	R/W	Carry Interrupt Enable Flag When the carry flag (CF) is set to 1, the CIE bit enables an interrupt. 0: A carry interrupt is not generated when the CF flag is set to 1 1: A carry interrupt is generated when the CF flag is set to 1
3	AIE	0	R/W	Alarm Interrupt Enable Flag When the alarm flag (AF) is set to 1, the AIE bit enables an interrupt. 0: An alarm interrupt is not generated when the AF flag is set to 1 1: An alarm interrupt is generated when the AF flag is set to 1
2, 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	AF	Undefined	R/W	Alarm Flag The AF flag is set when the alarm time, which is set by the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, or RYRAR whose ENB bit is set to 1), and counters match. 0: Alarm registers and counters not match [Clearing condition] When 0 is written to AF 1: Alarm registers and counters match* [Setting condition] When alarm registers (only the registers with ENB bit set to 1) and counters match Note: * Writing 1 holds previous value.

chanhvinh.com

13.3.17 Control Register 2 (RCR2)

RCR2 is a register for periodic interrupt control, 30-second adjustment, divider circuit RESET, and count control.

RCR2 is initialized by a power-on reset or in deep standby mode. The RTCEN bit is only initialized by a power-on reset using the $\overline{RES}$ pin.

Bit:	7	6	5	4	3	2	1	0
	PEF	PES[2:0]			RTCEN	ADJ	RESET	START
Initial value:	0	0	0	0	1	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	PEF	0	R/W	Periodic Interrupt Flag Indicates that an interrupt is generated with the period designated by the PES2 to PES0 bits. When set to 1, PEF generates periodic interrupts. 0: Interrupts not generated with the period designated by the bits PES2 to PES0. [Clearing condition] When 0 is written to PEF 1: Interrupts generated with the period designated by the PES2 to PES0 bits. [Setting condition] When an interrupt is generated with the period designated by the bits PES0 to PES2 or when 1 is written to the PEF flag
6 to 4	PES[2:0]	000	R/W	Interrupt Enable Flags These bits specify the period of the periodic interrupt. 000: No periodic interrupts generated 001: Setting prohibited 010: Periodic interrupt generated every 1/64 second 011: Periodic interrupt generated every 1/16 second 100: Periodic interrupt generated every 1/4 second 101: Periodic interrupt generated every 1/2 second 110: Periodic interrupt generated every 1 second 111: Periodic interrupt generated every 2 seconds
3	RTCEN	1	R/W	RTC_X1 Clock Control Controls the function of the RTC_X1 pin. 0: Halts the on-chip crystal oscillator/disables the external clock input. 1: Runs the on-chip crystal oscillator/enables the external clock input. Note: The on-chip crystal oscillator selected with the RCKSEL[1:0] bits in the RCR5 register runs. This bit must not be set to 1 when the RCKSEL[1:0] bits are set to 01.
2	ADJ	0	R/W	30-Second Adjustment When 1 is written to the ADJ bit, times of 29 seconds or less will be rounded down to 00 seconds and 30 seconds or more up to 1 minute. The divider circuit (prescaler and R64CNT) will be simultaneously reset. The ADJ bit is automatically reset to 0; there is no need to write 0 to this bit. This bit is always read as 0. 0: Normal clock operation 1: 30-second adjustment
1	RESET	0	R/W	Reset Writing 1 to this bit initializes the divider circuit, the R64CNT register, the alarm register, the RCR3 register, bits CF and AF in RCR1, and bit PEF in RCR2. In this case, the RESET bit is automatically reset to 0 after 1 is written to and the above registers are reset. Thus, there is no need to write 0 to this bit. This bit is always read as 0. 0: Normal clock operation 1: Divider circuit is reset.
0	START	1	R/W	Start Halts and restarts the counter (clock). 0: Second, minute, hour, day, week, month, and year counters halt. 1: Second, minute, hour, day, week, month, and year counters run normally.

chanhvinh.com

13.3.18 Control Register 3 (RCR3)

When the ENB bit is set to 1, RCR3 performs a comparison with the RYRCNT. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

Bit:	7	6	5	4	3	2	1	0
	ENB	-	-	-	-	-	-	-
Initial value:	Undefined	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	Undefined	R/W	When this bit is set to 1, comparison of the year alarm register (RYRAR) and the year counter (RYRCNT) is performed.
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

13.3.19 Control Register 5 (RCR5)

When the RCKSEL[1:0] bits in RCR5 are set to 00 and 01, the RTC_X1 clock pulses (32.768 kHz) and the EXTAL clock pulses are used for clock counting, respectively.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	RCKSEL[1:0]	
Initial value:	0	0	0	0	0	0	Undefined	Undefined
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	RCKSEL[1:0]	Undefined	R/W	Operation clock select Operation clock can be selected from RTC_X1 and EXTAL. The setting of these bits should not be switched during operation. 00: Selects RTC_X1 (32.768 kHz). 01: Selects EXTAL. 10: Reserved. 11: Setting prohibited.

chanhvinh.com

13.3.20 Frequency Register H/L (RFRH/L)

RFRH/L is a 16-bit readable/writable register.

The "frequency comparison value" is set in RFC[18:0] so that a 128-Hz clock is generated when the realtime clock operates at the EXTAL clock frequency.

Change the "frequency comparison value" according to the EXTAL clock frequency. The calculation method is shown below. When the RCKSEL bits in RCR5 are set to 00, setting this register is not necessary.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SEL64	-	-	-	-	-	-	-	-	-	-	-	-	RFC[18:16]		
Initial value:	Undefined	0	0	0	0	0	0	0	0	0	0	0	0	Undefined	Undefined	Undefined
R/W:	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFC[15:0]															
Initial value:	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	SEL64	Undefined	R/W	64-Hz Divider Select Indicates the EXTAL clock frequency is not dividable by 128 Hz but is dividable by 64 Hz. 0: EXTAL clock frequency is dividable by 128 Hz. 1: EXTAL clock frequency is not dividable by 128 Hz but is dividable by 64 Hz.
30 to 19	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
18 to 0	RFC[18:0]	Undefined	R/W	Frequency comparison value Sets the comparison value to generate operation clock from the EXTAL clock frequency.

(1) Method for calculating "frequency comparison value".

- When EXTAL clock frequency is dividable by 128 Hz:
  RFC[18:0] = (EXTAL clock frequency)/128
  Clear the SEL64 bit to 0.
- When EXTAL clock frequency is not dividable by 128 Hz but is dividable by 64 Hz:
  RFC[18:0] = (EXTAL clock frequency)/64
  Set the SEL64 bit to 1.

(2) Setting Example

Table 13.3 Setting Example

Clock Frequency		SEL64 Setting Value	RFC Setting Value
EXTAL	10 MHz	0	H'1312D
	11 MHz	1	H'29F63
	12 MHz	0	H'16E36
	13 MHz	1	H'31975

chanhvinh.com

13.4 Operation

A usage example of this module is shown below.

13.4.1 Initial Settings of Registers after Power-On and Oscillation Stabilization Time

All the registers should be initialized after the power is turned on. When the RTC_X1 crystal oscillator is used, oscillation stabilization time is necessary after changing the RTCEN bit in RCR2 from 0 to 1. During oscillation stabilization time, various configurations for or operation of the real time clock must not be performed. For details on oscillation stabilization time, refer to section 47, Electrical Characteristics.

13.4.2 Setting Time

Figure 13.2 shows how to set the time when the clock is stopped.

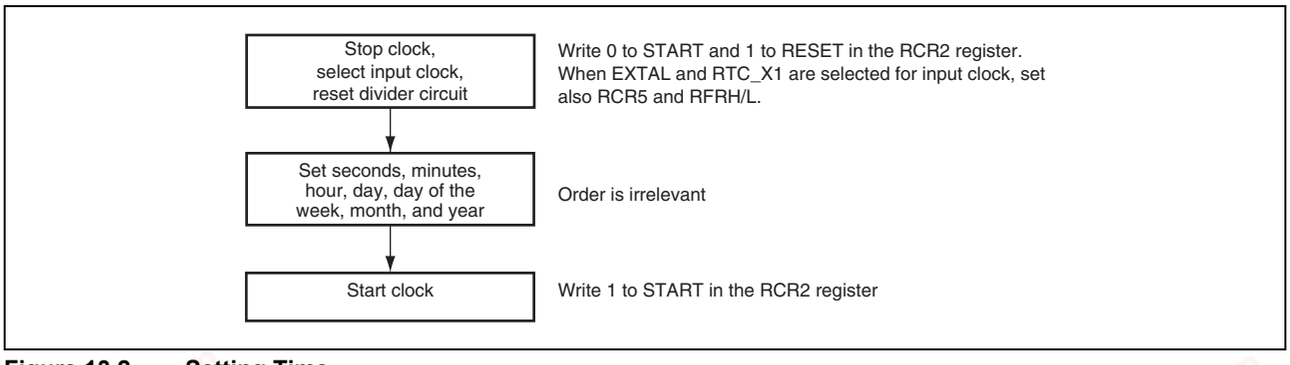


Figure 13.2 Setting Time

chanhvinh.com

13.4.3 Reading Time

Figure 13.3 shows how to read the time.

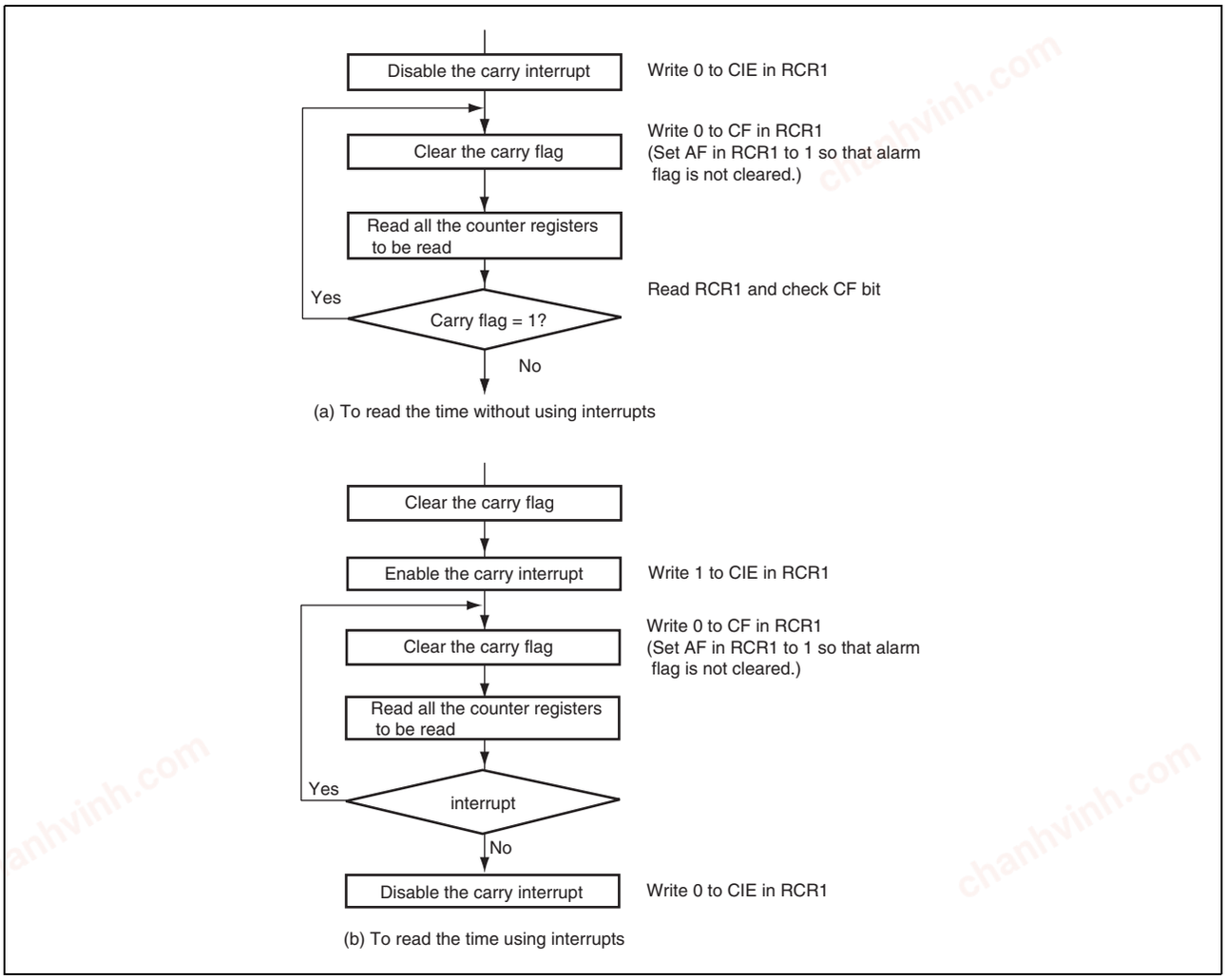


Figure 13.3 Reading Time

If a carry occurs while reading the time, the correct time will not be obtained, so it must be read again. Part (a) in Figure 13.3 shows the method of reading the time without using interrupts; part (b) in Figure 13.3 shows the method using carry interrupts. To keep programming simple, method (a) should normally be used.

chanhvinh.com

13.4.4 Alarm Function

Figure 13.4 shows how to use the alarm function.

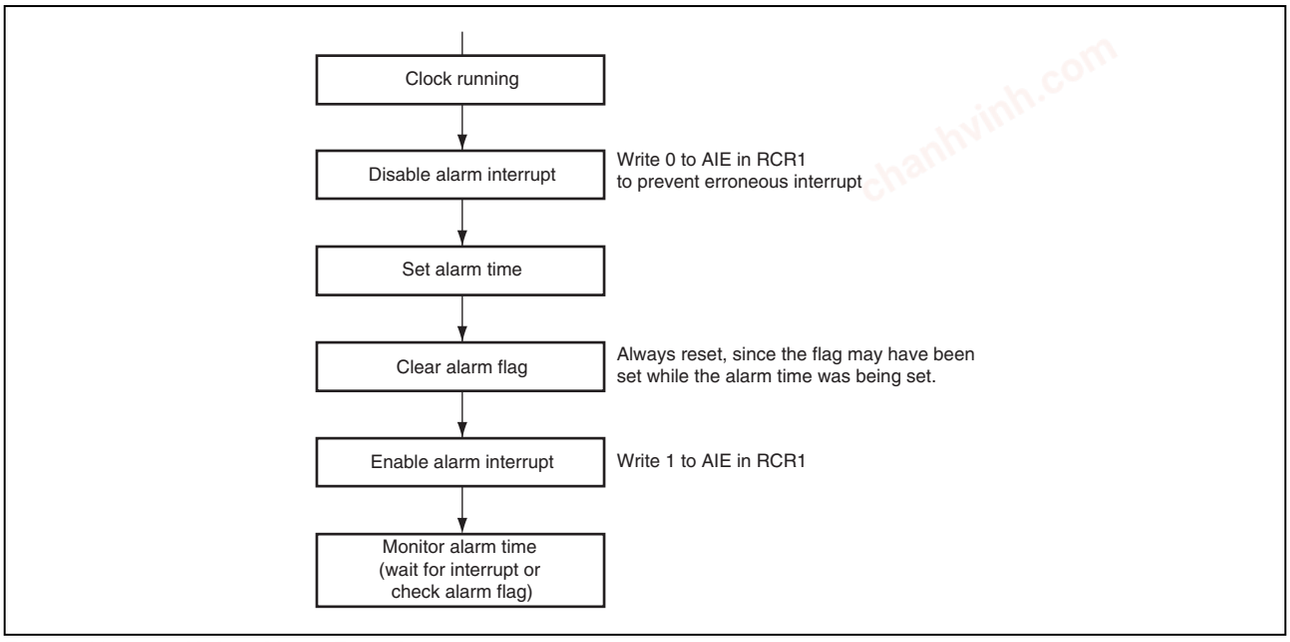


Figure 13.4 Using Alarm Function

Alarms can be generated using seconds, minutes, hours, day of the week, day, month, year, or any combination of these. Set the ENB bit in the register on which the alarm is placed to 1, and then set the alarm time in the lower bits. Clear the ENB bit in the register on which the alarm is not placed to 0.

When the clock and alarm times match, 1 is set in the AF bit in RCR1. Alarm detection can be checked by reading this bit, but normally it is done by interrupt. If 1 is set in the AIE bit in RCR1, an interrupt is generated when an alarm occurs.

The alarm flag is set when the clock and alarm times match. However, the alarm flag can be cleared by writing 0.

chanhvinh.com

13.5 Usage Notes

13.5.1 Register Writing during Count Operation

The following registers cannot be written to during count operation (while the START bit = 1 in RCR2).

RSECCNT, RMINCNT, RHRCNT, RDAYCNT, RWKCNT, RMONCNT, RYRCONT

Count operation must be stopped before writing to any of the above registers.

13.5.2 Use of Realtime Clock Periodic Interrupts

The method of using the periodic interrupt function is shown in Figure 13.5.

A periodic interrupt can be generated periodically at the interval set by bits PES2 to PES0 in RCR2. When the time set by bits PES2 to PES0 has elapsed, the PEF is set to 1.

The PEF is cleared to 0 upon periodic interrupt generation or when bits PES2 to PES0 are set. Periodic interrupt generation can be confirmed by reading this bit, but normally the interrupt function is used.

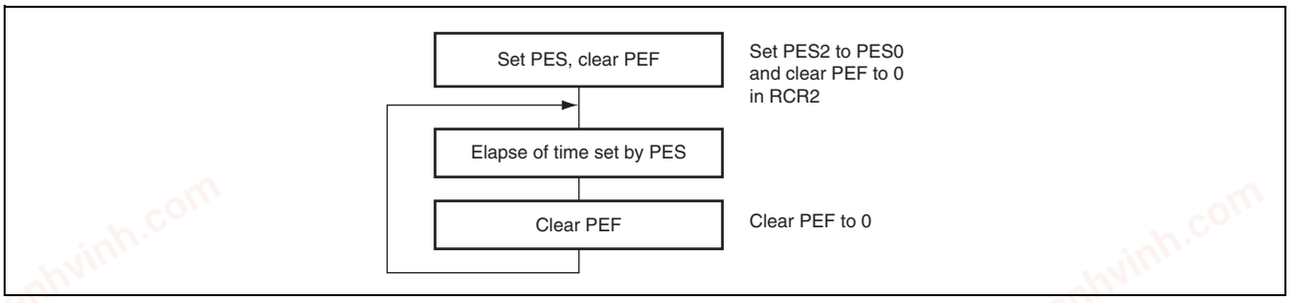


Figure 13.5 Using Periodic Interrupt Function

13.5.3 Transition to Standby Mode after Setting Register

When a transition to standby mode is made after registers in this module are set, sometimes counting is not performed correctly. After the registers are set, be sure to perform one dummy read of the registers before making a transition to standby mode.

13.5.4 Usage Notes when Writing to and Reading the Register

- After writing to the RCR2 register or any of the counters such as the second counter, dummy-read the register twice before reading the actual value. The register contents before the write are returned by the two dummy reads, and the third read returns the register contents reflecting the write.
- Registers other than the above can be read immediately after a write and the written value is reflected.

chanhvinh.com

14. Serial Communication Interface with FIFO

This LSI has a five-channel serial communication interface with FIFO that supports both asynchronous and clock synchronous serial communication. It also has 16-stage FIFO registers for both transmission and reception independently for each channel that enable this LSI to perform efficient high-speed continuous communication.

14.1 Features

- Asynchronous serial communication:
  - Serial data communication is performed by start-stop in character units. This module can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communication interface adapter (ACIA), or any other communications chip that employs a standard asynchronous serial system. There are eight selectable serial data communication formats.
  - Data length: 7 or 8 bits
  - Stop bit length: 1 or 2 bits
  - Parity: Even, odd, or none
  - Receive error detection: Parity, framing, and overrun errors
  - Break detection: Break is detected when a framing error is followed by at least one frame at the space 0 level (low level). It is also detected by reading the RxD pin level directly from the serial port register when a framing error occurs.
- Clock synchronous serial communication:
  - Serial data communication is synchronized with a clock signal. This module can communicate with other chips having a clock synchronous communication function. There is one serial data communication format.
  - Data length: 8 bits
  - Receive error detection: Overrun errors
- Full duplex communication: The transmitting and receiving sections are independent, so this module can transmit and receive simultaneously. Both sections use 16-stage FIFO buffering, so high-speed continuous data transfer is possible in both the transmit and receive directions.
- On-chip baud rate generator with selectable bit rates
- Internal or external transmit/receive clock source: From either baud rate generator (internal) or SCK pin (external)
- Four types of interrupts: Transmit-FIFO-data-empty interrupt, break interrupt, receive-FIFO-data-full interrupt, and receive-error interrupt are requested independently.
- When this module is not in use, it can be stopped by halting the clock supplied to it, saving power.
- In asynchronous mode, on-chip modem control functions ($\overline{\text{RTS}}$ and $\overline{\text{CTS}}$) (only channels 0, 1, and 2).
- The quantity of data in the transmit and receive FIFO data registers and the number of receive errors of the receive data in the receive FIFO data register can be ascertained.
- A time-out error (DR) can be detected when receiving in asynchronous mode.
- In asynchronous mode, the base clock frequency can be either 16 or 8 times the bit rate.
- When an internal clock is selected as a clock source and the SCK pin is used as an input pin in asynchronous mode, either normal mode or double-speed mode can be selected for the baud rate generator.

chanhvinh.com

Figure 14.1 shows a block diagram. However, certain channels do not have the $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$ pins.

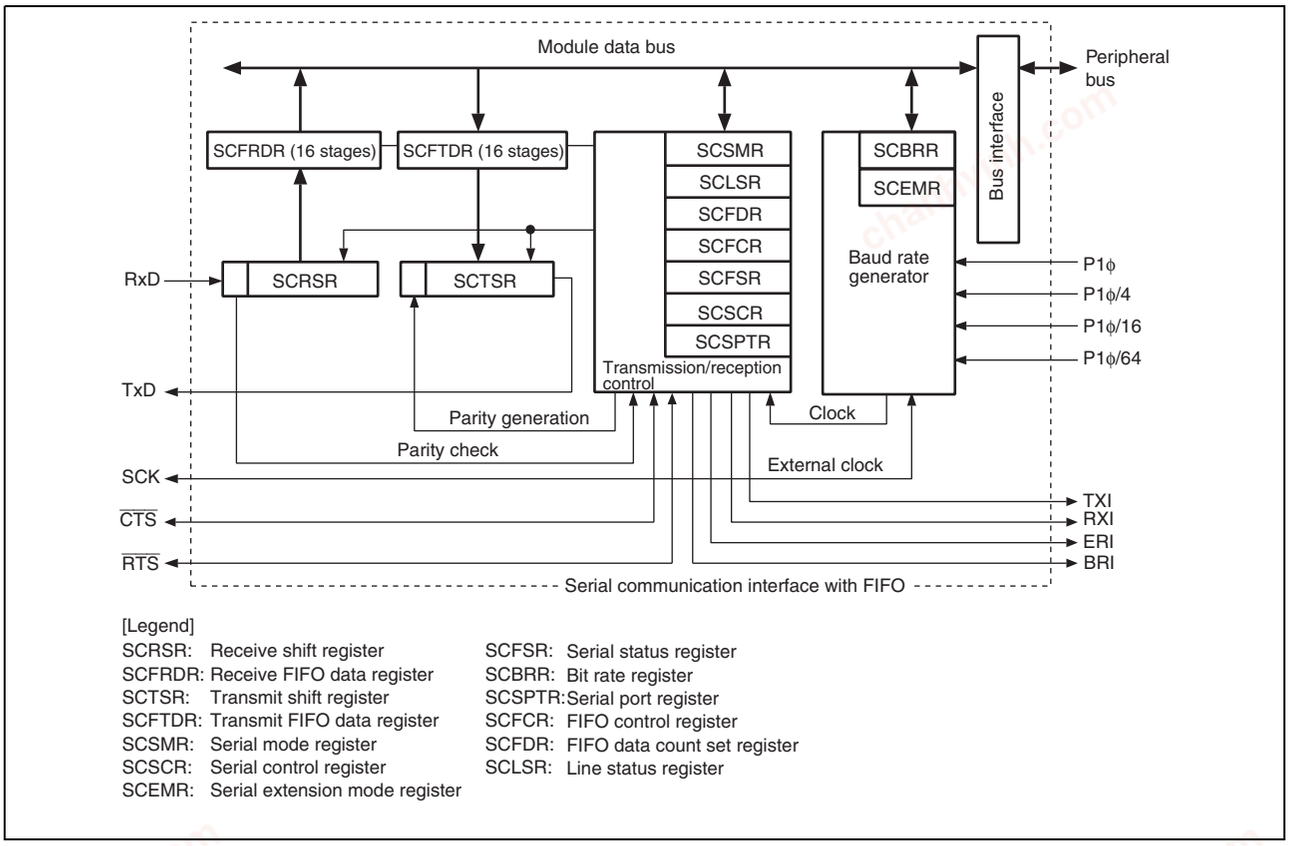


Figure 14.1 Block Diagram

chanhvinh.com

14.2 Input/Output Pins

Table 14.1 shows the pin configuration.

Table 14.1 Pin Configuration

Channel	Pin Name	Symbol	I/O	Function
0 to 4	Serial clock pins	SCK0 to SCK4	I/O	Clock I/O
	Receive data pins	RxD0 to RxD4	Input	Receive data input
	Transmit data pins	TxD0 to TxD4	Output	Transmit data output
0, 1, 2	Request to send pin	$\overline{\text{RTS0}}$, $\overline{\text{RTS1}}$, $\overline{\text{RTS2}}$	I/O	Request to send
	Clear to send pin	$\overline{\text{CTS0}}$, $\overline{\text{CTS1}}$, $\overline{\text{CTS2}}$	I/O	Clear to send

chanhvinh.com

14.3 Register Descriptions

This module has the following registers.

Table 14.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	Serial mode register_0	SCSMR_0	R/W	H'0000	H'E8007000	16
	Bit rate register_0	SCBRR_0	R/W	H'FF	H'E8007004	8
	Serial control register_0	SCSCR_0	R/W	H'0000	H'E8007008	16
	Transmit FIFO data register_0	SCFTDR_0	W	Undefined	H'E800700C	8
	Serial status register_0	SCFSR_0	R/(W)*1	H'0060	H'E8007010	16
	Receive FIFO data register_0	SCFRDR_0	R	Undefined	H'E8007014	8
	FIFO control register_0	SCFCR_0	R/W	H'0000	H'E8007018	16
	FIFO data count set register_0	SCFDR_0	R	H'0000	H'E800701C	16
	Serial port register_0	SCSPTR_0	R/W	H'0050	H'E8007020	16
	Line status register_0	SCLSR_0	R/(W)*2	H'0000	H'E8007024	16
	Serial extension mode register_0	SCEMR_0	R/W	H'0000	H'E8007028	16
1	Serial mode register_1	SCSMR_1	R/W	H'0000	H'E8007800	16
	Bit rate register_1	SCBRR_1	R/W	H'FF	H'E8007804	8
	Serial control register_1	SCSCR_1	R/W	H'0000	H'E8007808	16
	Transmit FIFO data register_1	SCFTDR_1	W	Undefined	H'E800780C	8
	Serial status register_1	SCFSR_1	R/(W)*1	H'0060	H'E8007810	16
	Receive FIFO data register_1	SCFRDR_1	R	Undefined	H'E8007814	8
	FIFO control register_1	SCFCR_1	R/W	H'0000	H'E8007818	16
	FIFO data count set register_1	SCFDR_1	R	H'0000	H'E800781C	16
	Serial port register_1	SCSPTR_1	R/W	H'0050	H'E8007820	16
	Line status register_1	SCLSR_1	R/(W)*2	H'0000	H'E8007824	16
	Serial extension mode register_1	SCEMR_1	R/W	H'0000	H'E8007828	16
2	Serial mode register_2	SCSMR_2	R/W	H'0000	H'E8008000	16
	Bit rate register_2	SCBRR_2	R/W	H'FF	H'E8008004	8
	Serial control register_2	SCSCR_2	R/W	H'0000	H'E8008008	16
	Transmit FIFO data register_2	SCFTDR_2	W	Undefined	H'E800800C	8
	Serial status register_2	SCFSR_2	R/(W)*1	H'0060	H'E8008010	16
	Receive FIFO data register_2	SCFRDR_2	R	Undefined	H'E8008014	8
	FIFO control register_2	SCFCR_2	R/W	H'0000	H'E8008018	16
	FIFO data count set register_2	SCFDR_2	R	H'0000	H'E800801C	16
	Serial port register_2	SCSPTR_2	R/W	H'0050	H'E8008020	16
	Line status register_2	SCLSR_2	R/(W)*2	H'0000	H'E8008024	16
	Serial extension mode register_2	SCEMR_2	R/W	H'0000	H'E8008028	16
3	Serial mode register_3	SCSMR_3	R/W	H'0000	H'E8008800	16
	Bit rate register_3	SCBRR_3	R/W	H'FF	H'E8008804	8
	Serial control register_3	SCSCR_3	R/W	H'0000	H'E8008808	16
	Transmit FIFO data register_3	SCFTDR_3	W	Undefined	H'E800880C	8
	Serial status register_3	SCFSR_3	R/(W)*1	H'0060	H'E8008810	16
	Receive FIFO data register_3	SCFRDR_3	R	Undefined	H'E8008814	8
	FIFO control register_3	SCFCR_3	R/W	H'0000	H'E8008818	16
	FIFO data count set register_3	SCFDR_3	R	H'0000	H'E800881C	16
	Serial port register_3	SCSPTR_3	R/W	H'0050	H'E8008820	16
	Line status register_3	SCLSR_3	R/(W)*2	H'0000	H'E8008824	16
	Serial extension mode register_3	SCEMR_3	R/W	H'0000	H'E8008828	16

chanhvinh.com

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
4	Serial mode register_4	SCSMR_4	R/W	H'0000	H'E8009000	16
	Bit rate register_4	SCBRR_4	R/W	H'FF	H'E8009004	8
	Serial control register_4	SCSCR_4	R/W	H'0000	H'E8009008	16
	Transmit FIFO data register_4	SCFTDR_4	W	Undefined	H'E800900C	8
	Serial status register_4	SCFSR_4	R/(W)*1	H'0060	H'E8009010	16
	Receive FIFO data register_4	SCFRDR_4	R	Undefined	H'E8009014	8
	FIFO control register_4	SCFCR_4	R/W	H'0000	H'E8009018	16
	FIFO data count set register_4	SCFDR_4	R	H'0000	H'E800901C	16
	Serial port register_4	SCSPTR_4	R/W	H'0050	H'E8009020	16
	Line status register_4	SCLSR_4	R/(W)*2	H'0000	H'E8009024	16
	Serial extension mode register_4	SCEMR_4	R/W	H'0000	H'E8009028	16

Note 1. Only 0 can be written to clear the flag. Bits 15 to 8, 3, and 2 are read-only bits that cannot be modified.
Note 2. Only 0 can be written to clear the flag. Bits 15 to 1 are read-only bits that cannot be modified.

14.3.1 Receive Shift Register (SCRSR)

SCRSR receives serial data. Data input at the RxD pin is loaded into SCRSR in the order received, LSB (bit 0) first, being converted to parallel form. When one byte has been received, it is automatically transferred to the receive FIFO data register (SCFRDR).

The CPU cannot read from or write to SCRSR directly.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

14.3.2 Receive FIFO Data Register (SCFRDR)

SCFRDR is a 16-stage FIFO register that stores serial receive data. The reception of one byte of serial data is complete when the received data is moved from the receive shift register (SCRSR) to SCFRDR for storage. Continuous reception is possible until 16 bytes are stored. The CPU can read but not write to SCFRDR. If data is read when there is no receive data in the SCFRDR, the value is undefined.

When SCFRDR is full of receive data, subsequent serial data is lost.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R

chanhvinh.com

14.3.3 Transmit Shift Register (SCTSR)

SCTSR transmits serial data. Transmit data is loaded from the transmit FIFO data register (SCFTDR) into SCTSR, then the data is transmitted serially from the TxD pin, LSB (bit 0) first. After one data byte has been transmitted, the next transmit data is automatically loaded from SCFTDR into SCTSR and transmission is started again.

The CPU cannot read from or write to SCTSR directly.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

14.3.4 Transmit FIFO Data Register (SCFTDR)

SCFTDR is a 16-stage FIFO register that stores data for serial transmission. When the transmit shift register (SCTSR) empty is detected, transmit data written in the SCFTDR is moved to SCTSR and serial transmission is started. Continuous serial transmission is performed until there is no transmit data left in SCFTDR. The CPU can write to SCFTDR at all times.

When SCFTDR is full of transmit data (16 bytes), no more data can be written. If writing of next data is attempted, the data is ignored.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W

chanhvinh.com

14.3.5 Serial Mode Register (SCSMR)

SCSMR specifies the serial communication format and selects the clock source for the baud rate generator.

The CPU can always read from and write to SCSMR.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	-	CKS[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	C/$\overline{A}$	0	R/W	Communication Mode Selects operating mode from asynchronous and clock synchronous modes. 0: Asynchronous mode 1: Clock synchronous mode
6	CHR	0	R/W	Character Length Selects 7-bit or 8-bit data length in asynchronous mode. In the clock synchronous mode, the data length is always 8 bits, regardless of the CHR setting. 0: 8-bit data 1: 7-bit data* Note: * When 7-bit data is selected, the MSB (bit 7) of the transmit FIFO data register is not transmitted.
5	PE	0	R/W	Parity Enable Selects whether to add a parity bit to transmit data and to check the parity of receive data, in asynchronous mode. In clock synchronous mode, a parity bit is neither added nor checked, regardless of the PE setting. 0: Parity bit not added or checked 1: Parity bit added and checked* Note: * When PE is set to 1, an even or odd parity bit is added to transmit data, depending on the parity mode (O/E) setting. Receive data parity is checked according to the even/odd (O/E) mode setting.
4	O/$\overline{E}$	0	R/W	Parity Mode Selects even or odd parity when parity bits are added and checked. The O/$\overline{E}$ setting is used only in asynchronous mode and only when the parity enable bit (PE) is set to 1 to enable parity addition and checking. The O/$\overline{E}$ setting is ignored in clock synchronous mode or in asynchronous mode when parity addition and checking is disabled. 0: Even parity*1 1: Odd parity*2 Note: 1. If even parity is selected, the parity bit is added to transmit data to make an even number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an even number of 1s in the received character and parity bit combined. 2. If odd parity is selected, the parity bit is added to transmit data to make an odd number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an odd number of 1s in the received character and parity bit combined.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
3	STOP	0	R/W	Stop Bit Length Selects one or two bits as the stop bit length in asynchronous mode. This setting is used only in asynchronous mode. It is ignored in clock synchronous mode because no stop bits are added. When receiving, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit, but if the second stop bit is 0, it is treated as the start bit of the next incoming character. 0: One stop bit When transmitting, a single 1-bit is added at the end of each transmitted character. 1: Two stop bits When transmitting, two 1 bits are added at the end of each transmitted character.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1, 0	CKS[1:0]	00	R/W	Clock Select Select the internal clock source of the on-chip baud rate generator. For further information on the clock source, bit rate register settings, and baud rate, see section 14.3.8, Bit Rate Register (SCBRR). 00: P1ϕ 01: P1ϕ/4 10: P1ϕ/16 11: P1ϕ/64 Note: P1ϕ: Peripheral clock

chanhvinh.com

14.3.6 Serial Control Register (SCSCR)

SCSCR enables/disables the transmitter/receiver operation and interrupt requests, and selects the transmit/receive clock source. The CPU can always read and write to SCSCR.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	TIE	RIE	TE	RE	REIE	-	CKE[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables the transmit-FIFO-data-empty interrupt (TXI) requested when the serial transmit data is transferred from the transmit FIFO data register (SCFTDR) to the transmit shift register (SCTSR), the quantity of data in the transmit FIFO register becomes less than the specified number of transmission triggers, and then the TDFE flag in the serial status register (SCFSR) is set to 1. 0: Transmit-FIFO-data-empty interrupt request (TXI) is disabled 1: Transmit-FIFO-data-empty interrupt request (TXI) is enabled* Note: * The TXI interrupt request can be cleared by writing a greater quantity of transmit data than the specified transmission trigger number to SCFTDR and clearing TDFE to 0 after reading 1 from TDFE, or by clearing TIE to 0.
6	RIE	0	R/W	Receive Interrupt Enable Enables or disables the receive FIFO data full (RXI) interrupts requested when the RDF flag or DR flag in serial status register (SCFSR) is set to 1, receive-error (ERI) interrupts requested when the ER flag in SCFSR is set to 1, and break (BRI) interrupts requested when the BRK flag in SCFSR or the ORER flag in line status register (SCLSR) is set to 1. 0: Receive FIFO data full interrupt (RXI), receive-error interrupt (ERI), and break interrupt (BRI) requests are disabled 1: Receive FIFO data full interrupt (RXI), receive-error interrupt (ERI), and break interrupt (BRI) requests are enabled* Note: * RXI interrupt requests can be cleared by reading the DR or RDF flag after it has been set to 1, then clearing the flag to 0, or by clearing RIE to 0. ERI or BRI interrupt requests can be cleared by reading the ER, BRK or ORER flag after it has been set to 1, then clearing the flag to 0, or by clearing RIE and REIE to 0.
5	TE	0	R/W	Transmit Enable Enables or disables serial transmission. 0: Serial transmission disabled 1: Serial transmission enabled* Note: * When this bit is set to 1, serial transmission starts after writing of transmit data into SCFTDR. Be sure to select the transmit format in SCSMR and SCFCR and reset the transmit FIFO before setting TE to 1.
4	RE	0	R/W	Receive Enable Enables or disables serial reception. 0: Serial reception disabled*1 1: Serial reception enabled*2 Notes: 1. Clearing RE to 0 does not affect the receive flags (DR, ER, BRK, RDF, FER, PER, and ORER). These flags retain their previous values. 2. When this bit is set to 1, serial reception starts when a start bit is detected in asynchronous mode, or synchronous clock is detected in clock synchronous mode. Be sure to select the receive format in SCSMR and SCFCR and reset the receive FIFO before setting RE to 1.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
3	REIE	0	R/W	Receive Error Interrupt Enable Enables or disables the receive-error (ERI) interrupts and break (BRI) interrupts. The setting of REIE bit is valid only when RIE bit is set to 0. 0: Receive-error interrupt (ERI) and break interrupt (BRI) requests are disabled 1: Receive-error interrupt (ERI) and break interrupt (BRI) requests are enabled* Note: * ERI or BRI interrupt requests can be cleared by reading the ER, BRK or ORER flag after it has been set to 1, then clearing the flag to 0, or by clearing RIE and REIE to 0. Even if RIE is set to 0, when REIE is set to 1, ERI or BRI interrupt requests are enabled.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1, 0	CKE[1:0]	00	R/W	Clock Enable Select the clock source and enable or disable clock output from the SCK pin. Depending on CKE[1:0], the SCK pin can be used for serial clock output or serial clock input. If synchronous clock output is set in clock synchronous mode, set the C/$\overline{A}$ bit in SCSMR to 1, and then set CKE[1:0]. • Asynchronous mode 00: Internal clock, SCK pin used for input (input signal is ignored) 01: Internal clock, SCK pin used for clock output (The output clock frequency is either 16 or 8 times the bit rate.) 10: External clock, SCK pin used for clock input (The input clock frequency is either 16 or 8 times the bit rate.) 11: Setting prohibited • Clock synchronous mode 00: Internal clock, SCK pin used for synchronous clock output 01: Internal clock, SCK pin used for synchronous clock output 10: External clock, SCK pin used for synchronous clock input 11: Setting prohibited

chanhvinh.com

14.3.7 Serial Status Register (SCFSR)

SCFSR is a 16-bit register. The upper 8 bits indicate the number of receive errors in the receive FIFO data register, and the lower 8 bits indicate the status flag indicating operating state.

The CPU can always read from and write to SCFSR, but cannot write 1 to the status flags (ER, TEND, TDFE, BRK, RDF, and DR). These flags can be cleared to 0 only if they have first been read (after being set to 1). The PER flag (bits 15 to 12 and bit 2) and the FER flag (bits 11 to 8 and bit 3) are read-only bits that cannot be written.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PER[3:0]				FER[3:0]				ER	TEND	TDFE	BRK	FER	PER	RDF	DR
Initial value:	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PER[3:0]	0000	R	Number of Parity Errors Indicate the quantity of data including a parity error in the receive data stored in the receive FIFO data register (SCFRDR). The value indicated by bits 15 to 12 after the ER bit in SCFSR is set, represents the number of parity errors in SCFRDR. When parity errors have occurred in all 16-byte receive data in SCFRDR, PER[3:0] shows 0000.
11 to 8	FER[3:0]	0000	R	Number of Framing Errors Indicate the quantity of data including a framing error in the receive data stored in SCFRDR. The value indicated by bits 11 to 8 after the ER bit in SCFSR is set, represents the number of framing errors in SCFRDR. When framing errors have occurred in all 16-byte receive data in SCFRDR, FER[3:0] shows 0000.
7	ER	0	R/(W)*	Receive Error Indicates the occurrence of a framing error, or of a parity error when receiving data that includes parity.*1 0: Receiving is in progress or has ended normally [Clearing conditions] • ER is cleared to 0 by a power-on reset • ER is cleared to 0 when 0 is written to after 1 is read from ER 1: A framing error or parity error has occurred. [Setting conditions] • ER is set to 1 when the stop bit is 0 after checking whether or not the last stop bit of the received data is 1 at the end of one data receive operation*2 • ER is set to 1 when the total number of 1s in the receive data plus parity bit does not match the even/odd parity specified by the O/E bit in SCSMR Note: 1. Clearing the RE bit to 0 in SCSCR does not affect the ER bit, which retains its previous value. Even if a receive error occurs, the receive data is transferred to SCFRDR and the receive operation is continued. Whether or not the data read from SCFRDR includes a receive error can be detected by the FER and PER bits in SCFSR. 2. In two stop bits mode, only the first stop bit is checked; the second stop bit is not checked.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
6	TEND	1	R/(W)*	Transmit End Indicates that when the last bit of a serial character was transmitted, SCFTDR did not contain valid data, so transmission has ended. 0: Transmission is in progress [Clearing condition] • TEND is cleared to 0 when 0 is written after 1 is read from TEND after transmit data is written in SCFTDR*1 1: End of transmission [Setting conditions] • TEND is set to 1 by a power-on reset • TEND is set to 1 when TE is cleared to 0 in the serial control register (SCSCR) • TEND is set to 1 when SCFTDR does not contain transmit data when the last bit of a one-byte serial character is transmitted Note: 1. Do not use this bit as a transmit end flag when the direct memory access controller writes data to SCFTDR due to a TXI interrupt request.
5	TDFE	1	R/(W)*	Transmit FIFO Data Empty Indicates that data has been transferred from the transmit FIFO data register (SCFTDR) to the transmit shift register (SCTSR), the quantity of data in SCFTDR has become less than the transmission trigger number specified by the TTRG[1:0] bits in the FIFO control register (SCFCR), and writing of transmit data to SCFTDR is enabled. 0: The quantity of transmit data written to SCFTDR is greater than the specified transmission trigger number [Clearing conditions] • TDFE is cleared to 0 when data exceeding the specified transmission trigger number is written to SCFTDR after 1 is read from TDFE and then 0 is written to TDFE • TDFE is cleared to 0 when direct memory access controller is activated by transmit FIFO data empty interrupt (TXI) and data exceeding the specified transmission trigger number is written to SCFTDR 1: The quantity of transmit data in SCFTDR is less than or equal to the specified transmission trigger number*1 [Setting conditions] • TDFE is set to 1 by a power-on reset • TDFE is set to 1 when the quantity of transmit data in SCFTDR becomes less than or equal to the specified transmission trigger number as a result of transmission Note: 1. Since SCFTDR is a 16-byte FIFO register, the maximum quantity of data that can be written when TDFE is 1 is "16 minus the specified transmission trigger number". If an attempt is made to write additional data, the data is ignored. The quantity of data in SCFTDR is indicated by the upper 8 bits of SCFDR.
4	BRK	0	R/(W)*	Break Detection Indicates that a break signal has been detected in receive data. 0: No break signal received [Clearing conditions] • BRK is cleared to 0 by a power-on reset • BRK is cleared to 0 when software reads BRK after it has been set to 1, then writes 0 to BRK 1: Break signal received*1 [Setting condition] • BRK is set to 1 when data including a framing error is received, followed by at least one frame at the space 0 level (low level) Note: 1. When a break is detected, transfer of the receive data (H'00) to SCFRDR stops after detection. When the break ends and the receive signal becomes mark 1, the transfer of receive data resumes.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
3	FER	0	R	Framing Error Indication Indicates a framing error in the data read from the receive FIFO data register (SCFRDR) in asynchronous mode. 0: No receive framing error occurred in the next data read from SCFRDR [Clearing conditions] • FER is cleared to 0 by a power-on reset • FER is cleared to 0 when no framing error is present in the next data read from SCFRDR 1: A receive framing error occurred in the next data read from SCFRDR [Setting condition] • FER is set to 1 when a framing error is present in the next data read from SCFRDR
2	PER	0	R	Parity Error Indication Indicates a parity error in the data read from the receive FIFO data register (SCFRDR) in asynchronous mode. 0: No receive parity error occurred in the next data read from SCFRDR [Clearing conditions] • PER is cleared to 0 by a power-on reset • PER is cleared to 0 when no parity error is present in the next data read from SCFRDR 1: A receive parity error occurred in the next data read from SCFRDR [Setting condition] • PER is set to 1 when a parity error is present in the next data read from SCFRDR
1	RDF	0	R/(W)*	Receive FIFO Data Full Indicates that receive data has been transferred to the receive FIFO data register (SCFRDR), and the quantity of data in SCFRDR has become more than the receive trigger number specified by the RTRG[1:0] bits in the FIFO control register (SCFCR). 0: The quantity of transmit data written to SCFRDR is less than the specified receive trigger number [Clearing conditions] • RDF is cleared to 0 by a power-on reset • RDF is cleared to 0 when the SCFRDR is read until the quantity of receive data in SCFRDR becomes less than the specified receive trigger number after 1 is read from RDF, and then 0 is written • RDF is cleared to 0 when the direct memory access controller is activated by receive FIFO data full interrupt (RXI) and SCFRDR is read until the quantity of receive data in it becomes less than the specified receive trigger number 1: The quantity of receive data in SCFRDR is equal to or greater than the specified receive trigger number [Setting condition] • RDF is set to 1 when a quantity of receive data equal to or greater than the specified receive trigger number is stored in SCFRDR*[1] Note: 1. As SCFRDR is a 16-byte FIFO register, the maximum quantity of data that can be read when RDF is 1 is the specified receive trigger number. If an attempt is made to read after all the data in SCFRDR has been read, the data is undefined. The quantity of receive data in SCFRDR is indicated by the lower 8 bits of SCFDR.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
0	DR	0	R/(W)*	Receive Data Ready Indicates that the quantity of data in the receive FIFO data register (SCFRDR) is less than the specified receive trigger number, and that the next data has not yet been received after the elapse of 15 ETU from the last stop bit in asynchronous mode. In clock synchronous mode, this bit is not set to 1. 0: Receiving is in progress, or no receive data remains in SCFRDR after receiving ended normally [Clearing conditions] • DR is cleared to 0 by a power-on reset • DR is cleared to 0 when all receive data are read from SCFRDR after 1 is read from DR, and then 0 is written. • DR is cleared to 0 when all receive data are read from SCFRDR after the direct memory access controller is activated by receive FIFO data full interrupt (RXI). 1: Next receive data has not been received [Setting condition] • DR is set to 1 when SCFRDR contains less data than the specified receive trigger number, and the next data has not yet been received after the elapse of 15 ETU from the last stop bit.*1 Note: 1. This is equivalent to 1.5 frames with the 8-bit, 1-stop-bit format. (ETU: elementary time unit)

Note: * Only 0 can be written to clear the flag after 1 is read.

chanhvinh.com

14.3.8 Bit Rate Register (SCBRR)

SCBRR is an 8-bit register that is used with the CKS[1:0] bits in the serial mode register (SCSMR) and the BGDM and ABCS bits in the serial extension mode register (SCEMR) to determine the serial transmit/receive bit rate.

The CPU can always read from and write to SCBRR. SCBRR is initialized to H'FF by a power-on reset. Each channel has independent baud rate generator control, so different values can be set in eight channels.

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The SCBRR setting is calculated as follows:

- Asynchronous mode:

When baud rate generator operates in normal mode (when the BGDM bit of SCEMR is 0):

$$N = \frac{P1\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$ (Operation on a base clock with a frequency of 16 times the bit rate)

$$N = \frac{P1\phi}{32 \times 2^{2n-1} \times B} \times 10^6 - 1$$ (Operation on a base clock with a frequency of 8 times the bit rate)

When baud rate generator operates in double speed mode (when the BGDM bit of SCEMR is 1):

$$N = \frac{P1\phi}{32 \times 2^{2n-1} \times B} \times 10^6 - 1$$ (Operation on a base clock with a frequency of 16 times the bit rate)

$$N = \frac{P1\phi}{16 \times 2^{2n-1} \times B} \times 10^6 - 1$$ (Operation on a base clock with a frequency of 8 times the bit rate)

chanhvinh.com

- Clock synchronous mode:

$$N = \frac{P1\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bits/s)
N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)
(The setting must satisfy the electrical characteristics.)
P1ϕ: Operating frequency for peripheral modules (MHz)
n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources and values of n, see Table 14.3.)

Table 14.3 SCSMR Settings

		SCSMR Settings	
n	**Clock Source**	**CKS[1]**	**CKS[0]**
0	P1ϕ	0	0
1	P1ϕ/4	0	1
2	P1ϕ/16	1	0
3	P1ϕ/64	1	1

The bit rate error in asynchronous mode is given by the following formula:

When baud rate generator operates in normal mode (the BGDM bit of SCEMR is 0):

$$\text{Error (\%)} = \left\{ \frac{P1\phi \times 10^6}{(N+1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$ (Operation on a base clock with a frequency of 16 times the bit rate)

$$\text{Error (\%)} = \left\{ \frac{P1\phi \times 10^6}{(N+1) \times B \times 32 \times 2^{2n-1}} - 1 \right\} \times 100$$ (Operation on a base clock with a frequency of 8 times the bit rate)

When baud rate generator operates in double speed mode (the BGDM bit of SCEMR is 1):

$$\text{Error (\%)} = \left\{ \frac{P1\phi \times 10^6}{(N+1) \times B \times 32 \times 2^{2n-1}} - 1 \right\} \times 100$$ (Operation on a base clock with a frequency of 16 times the bit rate)

$$\text{Error (\%)} = \left\{ \frac{P1\phi \times 10^6}{(N+1) \times B \times 16 \times 2^{2n-1}} - 1 \right\} \times 100$$ (Operation on a base clock with a frequency of 8 times the bit rate)

chanhvinh.com

Table 14.4 lists the sample SCBRR settings in asynchronous mode in which a base clock frequency is 16 times the bit rate (the ABCS bit in SCEMR is 0) and the baud rate generator operates in normal mode (the BGDM bit in SCEMR is 0), and Table 14.5 lists the sample SCBRR settings in clock synchronous mode.

Table 14.4 Bit Rates and SCBRR Settings (Asynchronous Mode, BGDM = 0, ABCS = 0)

	P1ϕ (MHz)					
	50			66.67		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)
110	3	221	-0.02			
150	3	162	-0.15	3	216	0.01
300	3	80	0.47	3	108	-0.45
600	2	162	-0.15	2	216	0.01
1200	2	80	0.47	2	108	-0.45
2400	1	162	-0.15	1	216	0.01
4800	1	80	0.47	1	108	-0.45
9600	0	162	-0.15	0	216	0.01
19200	0	80	0.47	0	108	-0.45
31250	0	49	0.00	0	66	-0.50
38400	0	40	-0.76	0	53	0.47

Note: The error rate should be ≤ 1 %.
[Legend]
Blank space: Setting impossible

Table 14.5 Bit Rates and SCBRR Settings (Clock Synchronous Mode)

	P1ϕ (MHz)			
	50		66.67	
Bit Rate (bits/s)	n	N	n	N
500	—	—		
1000	3	194	—	—
2500	3	77	3	103
5000	2	155	2	207
10000	2	77	2	103
25000	1	124	1	166
50000	1	62	1	82
100000	0	124	0	166
250000	0	49	0	66
500000	0	24	—	—
1000000	—	—	—	—
2000000	—	—	—	—

[Legend]
Blank space: Setting impossible
—: Setting possible, but error occurs

chanhvinh.com

Table 14.6 indicates the maximum bit rates in asynchronous mode when the baud rate generator is used. Table 14.7 lists the maximum bit rates in asynchronous mode when the external clock input is used. Table 14.8 lists the maximum bit rates in clock synchronous mode when the external clock input is used (when $t_{Scyc} = 12t_{pcyc}$*).

Note: * Make sure that the electrical characteristics of this LSI and that of a connected LSI are satisfied.

Table 14.6 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

	Settings				
P1ϕ (MHz)	**BGDM**	**ABCS**	**n**	**N**	**Maximum Bit Rate (bits/s)**
50	0	0	0	0	1562500
		1	0	0	3125000
	1	0	0	0	3125000
		1	0	0	6250000
66.67	0	0	0	0	2083333
		1	0	0	4166667
	1	0	0	0	4166667
		1	0	0	8333333

Table 14.7 Maximum Bit Rates with External Clock Input (Asynchronous Mode)

		Settings	
P1ϕ (MHz)	**External Input Clock (MHz)**	**ABCS**	**Maximum Bit Rate (bits/s)**
50	12.5000	0	781250
		1	1562500
66.67	16.6667	0	1041667
		1	2083333

Table 14.8 Maximum Bit Rates with External Clock Input (Clock Synchronous Mode, $t_{Scyc} = 12\ t_{pcyc}$)

P1ϕ (MHz)	**External Input Clock (MHz)**	**Maximum Bit Rate (bits/s)**
50	4.1667	4166666.7
66.67	5.5556	5555555.5

chanhvinh.com

14.3.9 FIFO Control Register (SCFCR)

SCFCR resets the quantity of data in the transmit and receive FIFO data registers, sets the trigger data quantity, and contains an enable bit for loop-back testing. SCFCR can always be read and written to by the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	RSTRG[2:0]			RTRG[1:0]		TTRG[1:0]		MCE	TFRST	RFRST	LOOP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 8	RSTRG[2:0]	000	R/W	$\overline{\text{RTS}}$ Output Active Trigger When the quantity of receive data in receive FIFO data register (SCFRDR) becomes equal to or greater than the trigger set number shown below, $\overline{\text{RTS}}$ signal is set to high. 000: 15 001: 1 010: 4 011: 6 100: 8 101: 10 110: 12 111: 14
7, 6	RTRG[1:0]	00	R/W	Receive FIFO Data Trigger Set the quantity of receive data which sets the receive data full (RDF) flag in the serial status register (SCFSR). The RDF flag is set to 1 when the quantity of receive data stored in the receive FIFO data register (SCFRDR) becomes equal to or greater than the set trigger number shown below. • Asynchronous mode 00: 1 01: 4 10: 8 11: 14 • Clock synchronous mode 00: 1 01: 2 10: 8 11: 14 Note: In clock synchronous mode, to transfer the receive data using the direct memory access controller, set the receive trigger number to 1. If set to other than 1, CPU must read the receive data left in SCFRDR.
5, 4	TTRG[1:0]	00	R/W	Transmit FIFO Data Trigger Set the quantity of remaining transmit data which sets the transmit FIFO data register empty (TDFE) flag in the serial status register (SCFSR). The TDFE flag is set to 1 when the quantity of transmit data in the transmit FIFO data register (SCFTDR) becomes equal to or less than the set trigger number shown below. 00: 8 (8)* 01: 4 (12)* 10: 2 (14)* 11: 0 (16)* Note: * Values in parentheses mean the number of empty bytes in SCFTDR when the TDFE flag is set to 1.
3	MCE	0	R/W	Modem Control Enable Enables modem control signals $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$. For channels 3 and 4 in clock synchronous mode, MCE bit should always be 0. 0: Modem signal disabled* 1: Modem signal enabled Note: * $\overline{\text{CTS}}$ is fixed at active 0 regardless of the input value, and $\overline{\text{RTS}}$ is also fixed at 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
2	TFRST	0	R/W	Transmit FIFO Data Register Reset Disables the transmit data in the transmit FIFO data register and resets the register to the empty state. 0: Reset operation disabled* 1: Reset operation enabled Note: * Reset operation is executed by a power-on reset.
1	RFRST	0	R/W	Receive FIFO Data Register Reset Disables the receive data in the receive FIFO data register and resets the register to the empty state. 0: Reset operation disabled* 1: Reset operation enabled Note: * Reset operation is executed by a power-on reset.
0	LOOP	0	R/W	Loop-Back Test Internally connects the transmit output pin (TxD) and receive input pin (RxD) and internally connects the $\overline{\text{RTS}}$ pin and $\overline{\text{CTS}}$ pin and enables loop-back testing. 0: Loop back test disabled 1: Loop back test enabled

chanhvinh.com

14.3.10 FIFO Data Count Set Register (SCFDR)

SCFDR is a 16-bit register which indicates the quantity of data stored in the transmit FIFO data register (SCFTDR) and the receive FIFO data register (SCFRDR).

It indicates the quantity of transmit data in SCFTDR with the upper 8 bits, and the quantity of receive data in SCFRDR with the lower 8 bits. SCFDR can always be read by the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	T[4:0]					-	-	-	R[4:0]				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12 to 8	T[4:0]	00000	R	T4 to T0 bits indicate the quantity of non-transmitted data stored in SCFTDR. H'00 means no transmit data, and H'10 means that all transmit data is stored in SCFTDR.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4 to 0	R[4:0]	00000	R	R4 to R0 bits indicate the quantity of receive data stored in SCFRDR. H'00 means no receive data, and H'10 means that all receive data is stored in SCFRDR.

chanhvinh.com

14.3.11 Serial Port Register (SCSPTR)

SCSPTR controls input/output and data of pins multiplexed to the functions of this module. Bits 7 and 6 can control input/output data of $\overline{\text{RTS}}$ pin. Bits 5 and 4 can control input/output data of $\overline{\text{CTS}}$ pin. Bits 3 and 2 can control input/output data of SCK pin. Bits 1 and 0 can input data from RxD pin and output data to TxD pin, so they control break of serial transmitting/receiving.

The CPU can always read and write to SCSPTR.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT
Initial value:	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	RTSIO	0	R/W	$\overline{\text{RTS}}$ Port Input/Output Specifies input or output for the serial port $\overline{\text{RTS}}$ pin. When the $\overline{\text{RTS}}$ pin is actually used as a port outputting the RTSDT bit value, the MCE bit in SCFCR should be cleared to 0. 0: RTSDT bit value not output to $\overline{\text{RTS}}$ pin 1: RTSDT bit value output to $\overline{\text{RTS}}$ pin
6	RTSDT	1	R/W	$\overline{\text{RTS}}$ Port Data Specifies the input/output data of the serial port $\overline{\text{RTS}}$ pin. Input/output is specified by the RTSIO bit. For output, the RTSDT bit value is output to the $\overline{\text{RTS}}$ pin. The $\overline{\text{RTS}}$ pin status is read from the RTSDT bit regardless of the RTSIO bit setting. However, $\overline{\text{RTS}}$ input/output must be set in the general purpose I/O ports. 0: Input/output data is low level 1: Input/output data is high level
5	CTSIO	0	R/W	$\overline{\text{CTS}}$ Port Input/Output Specifies input or output for the serial port $\overline{\text{CTS}}$ pin. When the $\overline{\text{CTS}}$ pin is actually used as a port outputting the CTSDT bit value, the MCE bit in SCFCR should be cleared to 0. 0: CTSDT bit value not output to $\overline{\text{CTS}}$ pin 1: CTSDT bit value output to $\overline{\text{CTS}}$ pin
4	CTSDT	1	R/W	$\overline{\text{CTS}}$ Port Data Specifies the input/output data of the serial port $\overline{\text{CTS}}$ pin. Input/output is specified by the CTSIO bit. For output, the CTSDT bit value is output to the $\overline{\text{CTS}}$ pin. The $\overline{\text{CTS}}$ pin status is read from the CTSDT bit regardless of the CTSIO bit setting. However, $\overline{\text{CTS}}$ input/output must be set in the general purpose I/O ports. 0: Input/output data is low level 1: Input/output data is high level
3	SCKIO	0	R/W	SCK Port Input/Output Specifies input or output for the serial port SCK pin. When the SCK pin is actually used as a port outputting the SCKDT bit value, the CKE[1:0] bits in SCSCR should be cleared to 0. 0: SCKDT bit value not output to SCK pin 1: SCKDT bit value output to SCK pin
2	SCKDT	0	R/W	SCK Port Data Specifies the input/output data of the serial port SCK pin. Input/output is specified by the SCKIO bit. For output, the SCKDT bit value is output to the SCK pin. The SCK pin status is read from the SCKDT bit regardless of the SCKIO bit setting. However, SCK input/output must be set in the general purpose I/O ports. 0: Input/output data is low level 1: Input/output data is high level

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
1	SPB2IO	0	R/W	Serial Port Break Input/Output Specifies input or output for the serial port TxD pin. When the TxD pin is actually used as a port outputting the SPB2DT bit value, the TE bit in SCSCR should be cleared to 0. 0: SPB2DT bit value not output to TxD pin 1: SPB2DT bit value output to TxD pin
0	SPB2DT	0	R/W	Serial Port Break Data Specifies the input data of the RxD pin and the output data of the TxD pin used as serial ports. Input/output is specified by the SPB2IO bit. When the TxD pin is set to output, the SPB2DT bit value is output to the TxD pin. The RxD pin status is read from the SPB2DT bit regardless of the SPB2IO bit setting. However, RxD input and TxD output must be set in the general purpose I/O ports. 0: Input/output data is low level 1: Input/output data is high level

chanhvinh.com

14.3.12 Line Status Register (SCLSR)

The CPU can always read or write to SCLSR, but cannot write 1 to the ORER flag. This flag can be cleared to 0 only if it has first been read (after being set to 1).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	ORER
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/(W)*

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ORER	0	R/(W)*	Overrun Error Indicates the occurrence of an overrun error during reception. 0: Receiving is in progress or has ended normally*1 [Clearing conditions] • ORER is cleared to 0 by a power-on reset • ORER is cleared to 0 when 0 is written after 1 is read from ORER. 1: An overrun error has occurred during reception*2 [Setting condition] • ORER is set to 1 when the next serial receiving is finished while the receive FIFO is full of 16-byte receive data. Note: 1. Clearing the RE bit to 0 in SCSCR does not affect the ORER bit, which retains its previous value. 2. The receive FIFO data register (SCFRDR) retains the data before an overrun error has occurred, and the next received data is discarded. When the ORER bit is set to 1, the next serial reception cannot be continued.

chanhvinh.com

14.3.13 Serial Extension Mode Register (SCEMR)

The CPU can always read from or write to SCEMR. Setting the BGDM bit in this register to 1 allows the baud rate generator in this module to operate in double-speed mode when asynchronous mode is selected (by setting the C/$\overline{A}$ bit in SCSMR to 0) and an internal clock is selected as a clock source and the SCK pin is set as an input pin (by setting the CKE[1:0] bits in SCSCR to 00).
The base clock frequency within a bit period in asynchronous mode can be switched by the setting of the ABCS bit.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	BGDM	-	-	-	-	-	-	ABCS
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	BGDM	0	R/W	Baud Rate Generator Double-Speed Mode When the BGDM bit is set to 1, the baud rate generator in this module operates in double-speed mode. This bit is valid only when asynchronous mode is selected by setting the C/$\overline{A}$ bit in SCSMR to 0 and an internal clock is selected as a clock source and the SCK pin is set as an input pin by setting the CKE[1:0] bits in SCSCR to 00. In other settings, use normal mode. 0: Normal mode 1: Double-speed mode
6 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ABCS	0	R/W	Base Clock Select in Asynchronous Mode This bit selects the base clock frequency within a bit period in asynchronous mode. This bit is valid only in asynchronous mode (when the C/$\overline{A}$ bit in SCSMR is 0). 0: Base clock frequency is 16 times the bit rate 1: Base clock frequency is 8 times the bit rate

chanhvinh.com

14.4 Operation

14.4.1 Overview

For serial communication, this module has an asynchronous mode in which characters are synchronized individually, and a clock synchronous mode in which communication is synchronized with clock pulses.

This module has a 16-stage FIFO buffer for both transmission and reception, reducing the overhead of the CPU, and enabling continuous high-speed communication. Furthermore, channels 0, 1, and 2 have $\overline{\text{RTS}}$ and $\overline{\text{CTS}}$ signals to be used as modem control signals.

The transmission/reception format is selected in the serial mode register (SCSMR), as shown in Table 14.9. The clock source is selected by the combination of the C/$\overline{\text{A}}$ bit in SCSMR and the CKE[1:0] bits in the serial control register (SCSCR), as shown in Table 14.10.

(1) Asynchronous Mode

- Data length is selectable: 7 or 8 bits
- Parity bit is selectable. So is the stop bit length (1 or 2 bits). The combination of the preceding selections constitutes the communication format and character length.
- In receiving, it is possible to detect framing errors, parity errors, receive FIFO data full, overrun errors, receive data ready, and breaks.
- The number of stored data bytes is indicated for both the transmit and receive FIFO registers.
- An internal or external clock can be selected as the clock source.
  - When an internal clock is selected, this module operates using the clock of on-chip baud rate generator.
  - When an external clock is selected, the external clock input must have a frequency 16 or 8 times the bit rate. (The on-chip baud rate generator is not used.)

chanhvinh.com

(2) Clock Synchronous Mode

- The transmission/reception format has a fixed 8-bit data length.
- In receiving, it is possible to detect overrun errors (ORER).
- An internal or external clock can be selected as the clock source.
  - When an internal clock is selected, this module operates using the clock of the on-chip baud rate generator, and outputs this clock to external devices as the synchronous clock.
  - When an external clock is selected, this module operates on the input external synchronous clock not using the on-chip baud rate generator.

Table 14.9 SCSMR Settings and Communication Formats

SCSMR Settings					Communication Format		
Bit 7 C/$\overline{A}$	Bit 6 CHR	Bit 5 PE	Bit 3 STOP	Mode	Data Length	Parity Bit	Stop Bit Length
0	0	0	0	Asynchronous	8 bits	Not set	1 bit
			1				2 bits
		1	0			Set	1 bit
			1				2 bits
	1	0	0		7 bits	Not set	1 bit
			1				2 bits
		1	0			Set	1 bit
			1				2 bits
1	x	x	x	Clock synchronous	8 bits	Not set	None

[Legend]
x: Don't care

Table 14.10 SCSMR and SCSCR Settings and Clock Source Selection

SCSMR	SCSCR			
Bit 7 C/$\overline{A}$	Bit 1, 0 CKE[1:0]	Mode	Clock Source	SCK Pin Function
0	00	Asynchronous	Internal	This module does not use the SCK pin.
	01			Outputs a clock with a frequency 16/8 times the bit rate
	10		External	Inputs a clock with frequency 16/8 times the bit rate
	11		Setting prohibited	
1	0x	Clock synchronous	Internal	Outputs the synchronous clock
	10		External	Inputs the synchronous clock
	11		Setting prohibited	

[Legend]
x: Don't care
Note: When using the baud rate generator in double-speed mode (BGMD = 1), select asynchronous mode by setting the C/$\overline{A}$ bit to 0, and select an internal clock as a clock source and the SCK pin is not used (the CKE[1:0] bits set to 00).

chanhvinh.com

14.4.2 Operation in Asynchronous Mode

In asynchronous mode, each transmitted or received character begins with a start bit and ends with a stop bit. Serial communication is synchronized one character at a time.

The transmitting and receiving sections in this module are independent, so full duplex communication is possible. The transmitter and receiver are 16-stage FIFO buffered, so data can be written and read while transmitting and receiving are in progress, enabling continuous transmitting and receiving.

Figure 14.2 shows the general format of asynchronous serial communication.

In asynchronous serial communication, the communication line is normally held in the mark (high) state. This module monitors the line and starts serial communication when the line goes to the space (low) state, indicating a start bit.

One serial character consists of a start bit (low), data (LSB first), parity bit (high or low), and stop bit (high), in that order. When receiving in asynchronous mode, this module synchronizes at the falling edge of the start bit. This module samples each data bit on the eighth or fourth pulse of a clock with a frequency 16 or 8 times the bit rate; receive data is latched at the center of each bit.

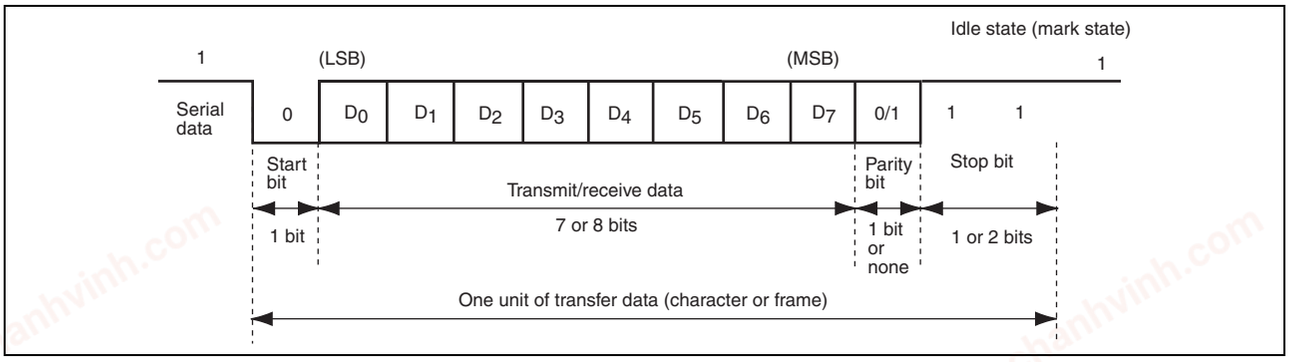


Figure 14.2 Example of Data Format in Asynchronous Communication (8-Bit Data with Parity and Two Stop Bits)

chanhvinh.com

(1) Transmit/Receive Formats

Table 14.11 lists the eight communication formats that can be selected in asynchronous mode. The format is selected by settings in the serial mode register (SCSMR).

Table 14.11 Serial Communication Formats (Asynchronous Mode)

SCSMR Bits			Serial Transmit/Receive Format and Frame Length											
CHR	PE	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	START	8-bit data								STOP		
0	0	1	START	8-bit data								STOP	STOP	
0	1	0	START	8-bit data								P	STOP	
0	1	1	START	8-bit data								P	STOP	STOP
1	0	0	START	7-bit data							STOP			
1	0	1	START	7-bit data							STOP	STOP		
1	1	0	START	7-bit data							P	STOP		
1	1	1	START	7-bit data							P	STOP	STOP	

[Legend]
START: Start bit
STOP: Stop bit
P: Parity bit

chanhvinh.com

(2) Clock

An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the transmit/receive clock. The clock source is selected by the C/$\overline{A}$ bit in the serial mode register (SCSMR) and the CKE[1:0] bits in the serial control register (SCSCR). For clock source selection, refer to Table 14.10, SCSMR and SCSCR Settings and Clock Source Selection.

When an external clock is input at the SCK pin, it must have a frequency equal to 16 or 8 times the desired bit rate. When this module operates on an internal clock, it can output a clock signal on the SCK pin. The frequency of this output clock is 16 or 8 times the desired bit rate.

(3) Transmitting and Receiving Data

- Initialization (Asynchronous Mode)
  Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize this module as follows.
  When changing the operation mode or the communication format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 initializes the transmit shift register (SCTSR). Clearing TE and RE to 0, however, does not initialize the serial status register (SCFSR), transmit FIFO data register (SCFTDR), or receive FIFO data register (SCFRDR), which retain their previous contents. Clear TE to 0 after all transmit data has been transmitted and the TEND flag in the SCFSR is set. The TE bit can be cleared to 0 during transmission, but the transmit data goes to the Mark state after the bit is cleared to 0. Set the TFRST bit in SCFCR to 1 and reset SCFTDR before TE is set again to start transmission.
  When an external clock is used, the clock should not be stopped during initialization or subsequent operation. The operation becomes unreliable if the clock is stopped.

chanhvinh.com

Figure 14.3 shows a sample flowchart for initialization.

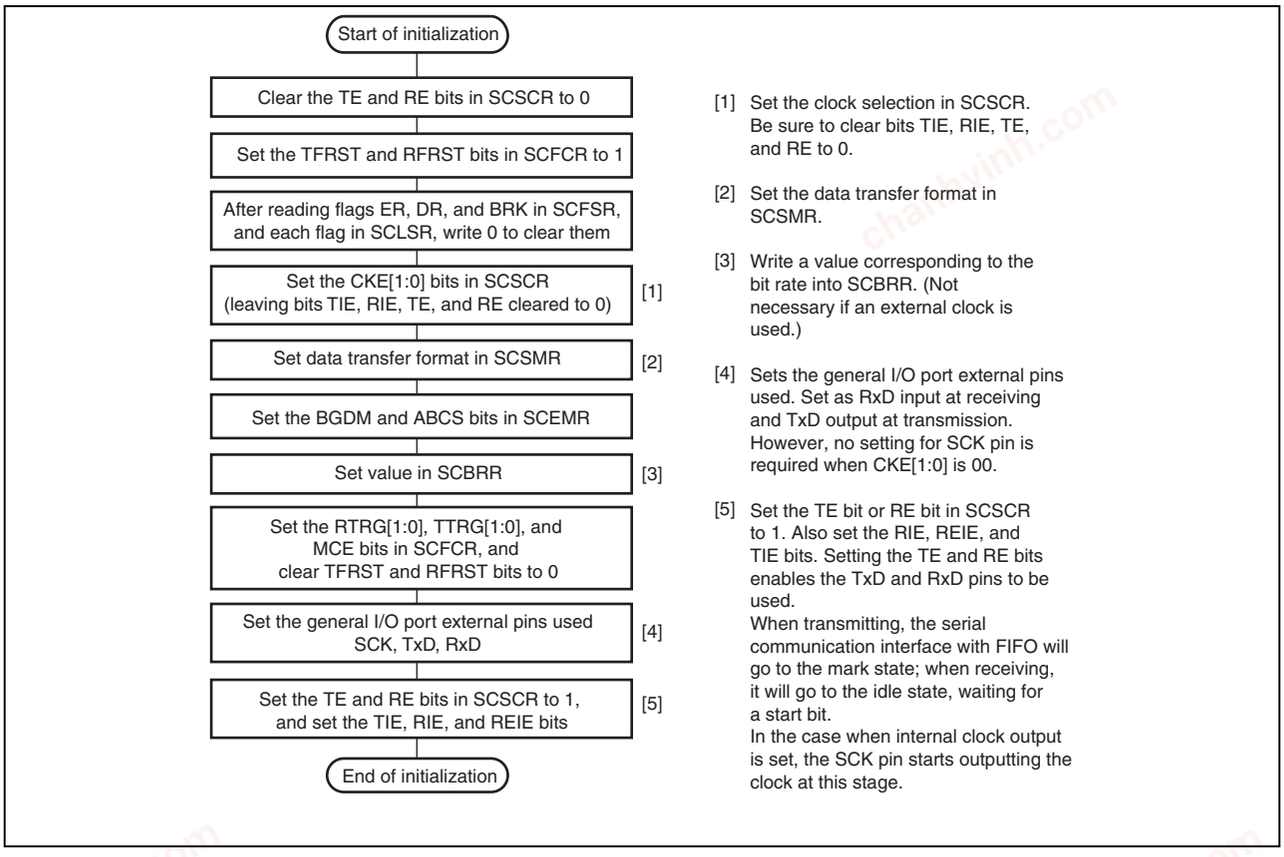


Figure 14.3 Sample Flowchart for Initialization

chanhvinh.com

- Transmitting Serial Data (Asynchronous Mode)

  Figure 14.4 shows a sample flowchart for serial transmission.

  Use the following procedure for serial data transmission after enabling transmission.

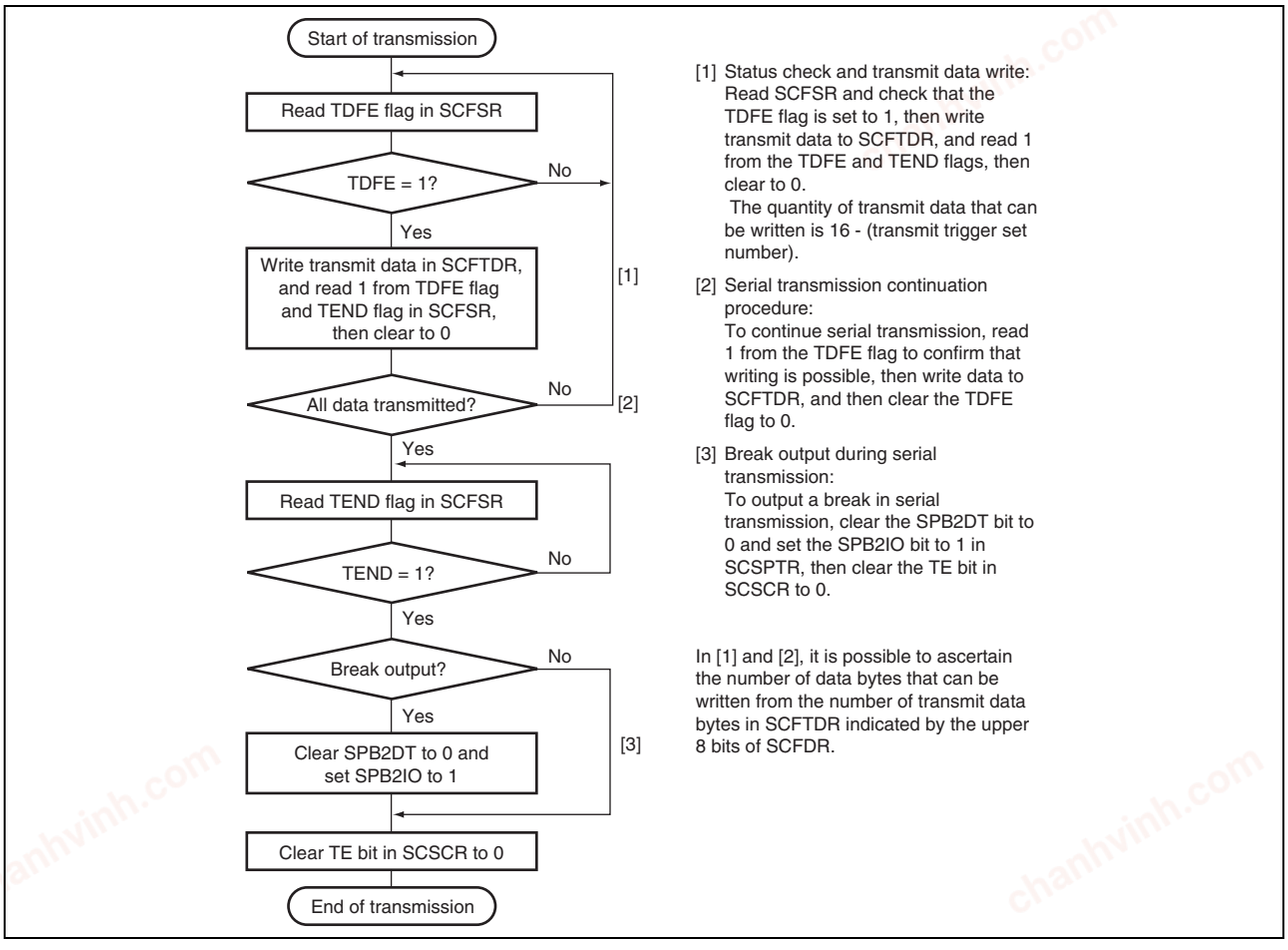


Figure 14.4 Sample Flowchart for Transmitting Serial Data

chanhvinh.com

In serial transmission, this module operates as described below.

1. When data is written into the transmit FIFO data register (SCFTDR), the data is transferred from SCFTDR to the transmit shift register (SCTSR). Confirm that the TDFE flag in the serial status register (SCFSR) is set to 1 before writing transmit data to SCFTDR. The number of data bytes that can be written is (16 – transmit trigger setting).

2. When data is transferred from SCFTDR to SCTSR and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCFTDR. When the number of transmit data bytes in SCFTDR becomes equal to or less than the transmit trigger number set in the FIFO control register (SCFCR), the TDFE flag is set. If the TIE bit in the serial control register (SCSR) is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.
   The serial transmit data is sent from the TxD pin in the following order.

   A. Start bit: One-bit 0 is output.
   B. Transmit data: 8-bit or 7-bit data is output in LSB-first order.
   C. Parity bit: One parity bit (even or odd parity) is output. (A format in which a parity bit is not output can also be selected.)
   D. Stop bit(s): One or two 1 bits (stop bits) are output.
   E. Mark state: 1 is output continuously until the start bit that starts the next transmission is sent.

3. The SCFTDR transmit data is checked at the timing for sending the stop bit. If data is present, the data is transferred from SCFTDR to SCTSR, the stop bit is sent, and then serial transmission of the next frame is started.

chanhvinh.com

Figure 14.5 shows an example of the operation for transmission.

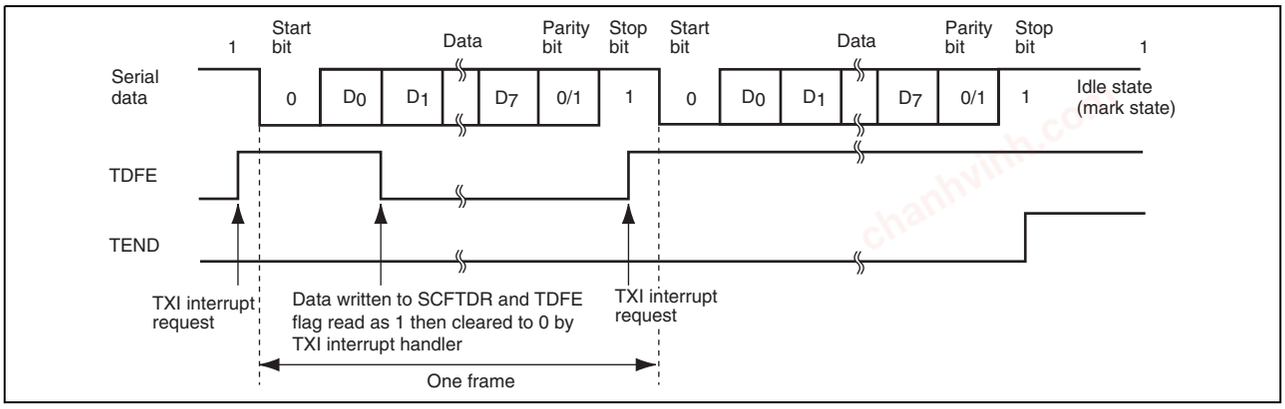


Figure 14.5 Example of Transmit Operation (8-Bit Data, Parity, 1 Stop Bit)

4. When modem control is enabled in channels 0, 1, and 2, transmission can be stopped and restarted in accordance with the $\overline{\text{CTS}}$ input value. When $\overline{\text{CTS}}$ is set to 1, if transmission is in progress, the line goes to the mark state after transmission of one frame. When $\overline{\text{CTS}}$ is set to 0, the next transmit data is output starting from the start bit.

Figure 14.6 shows an example of the operation when modem control is used.

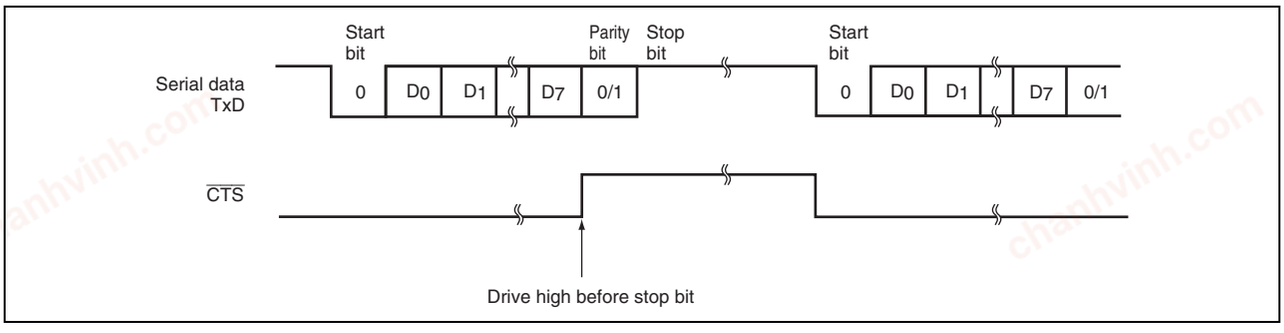


Figure 14.6 Example of Operation Using Modem Control ($\overline{\text{CTS}}$)

chanhvinh.com

- Receiving Serial Data (Asynchronous Mode)

  Figure 14.7 and Figure 14.8 show sample flowcharts for serial reception.
  Use the following procedure for serial data reception after enabling reception.

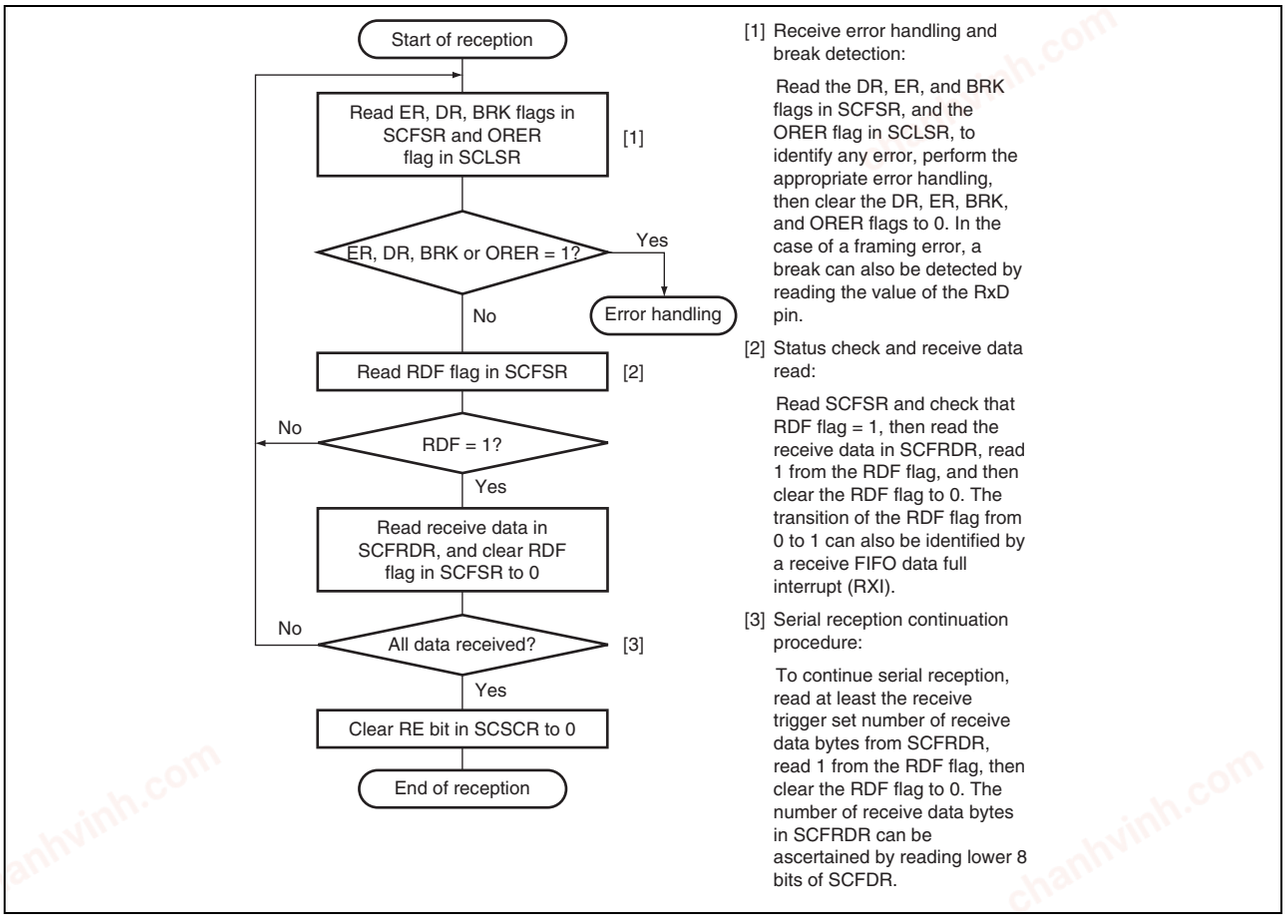


Figure 14.7 Sample Flowchart for Receiving Serial Data (1)

chanhvinh.com

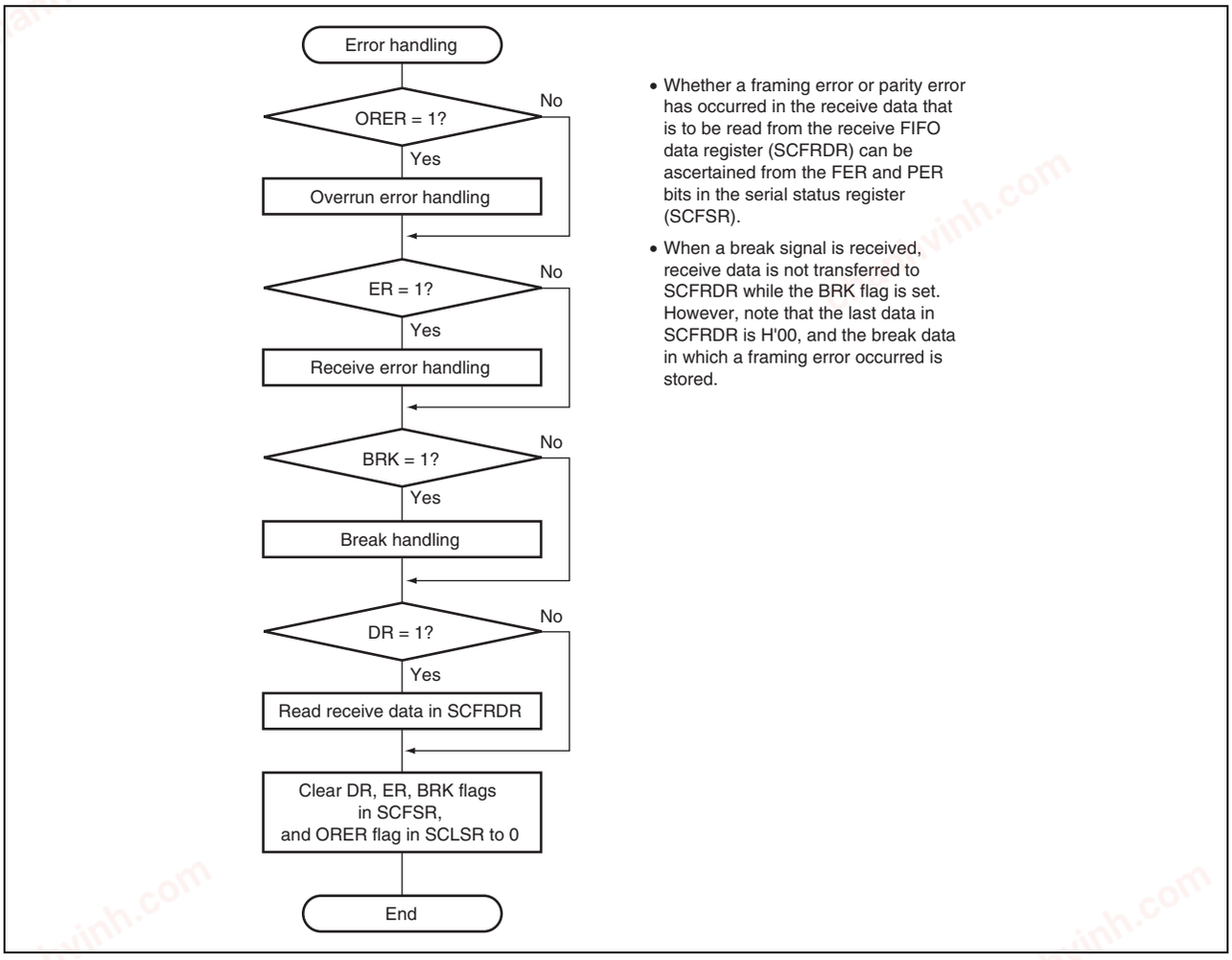


Figure 14.8 Sample Flowchart for Receiving Serial Data (2)

chanhvinh.com

In serial reception, this module operates as described below.

1. The transmission line is monitored, and if a 0 start bit is detected, internal synchronization is performed and reception is started.

2. The received data is stored in SCRSR in LSB-to-MSB order.

3. The parity bit and stop bit are received.
   After receiving these bits, this module carries out the following checks.

   A. Stop bit check: Checks whether the stop bit is 1. If there are two stop bits, only the first is checked.
   B. Checks whether receive data can be transferred from the receive shift register (SCRSR) to SCFRDR.
   C. Overrun check: Checks that the ORER flag is 0, indicating that the overrun error has not occurred.
   D. Break check: Checks that the BRK flag is 0, indicating that the break state is not set.

   If all the above checks are passed, the receive data is stored in SCFRDR.

   Note: When a parity error or a framing error occurs, reception is not suspended.

4. If the RIE bit in SCSCR is set to 1 when the RDF or DR flag changes to 1, a receive-FIFO-data-full interrupt (RXI) request is generated. If the RIE bit or the REIE bit in SCSCR is set to 1 when the ER flag changes to 1, a receive-error interrupt (ERI) request is generated. If the RIE bit or the REIE bit in SCSCR is set to 1 when the BRK or ORER flag changes to 1, a break reception interrupt (BRI) request is generated.

chanhvinh.com

Figure 14.9 shows an example of the operation for reception in asynchronous mode.

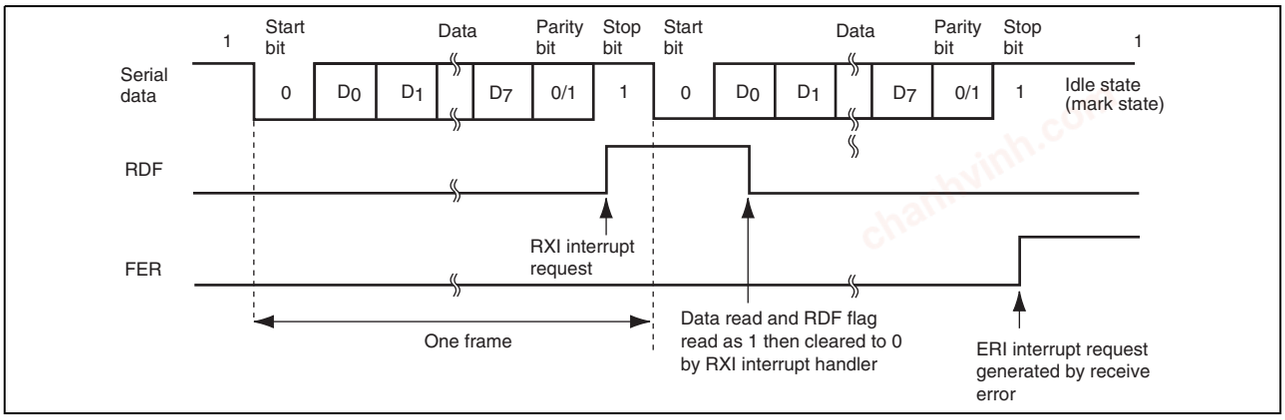


Figure 14.9 Example of Receive Operation(8-Bit Data, Parity, 1 Stop Bit)

5. When modem control is enabled in channels 0, 1, and 2, the $\overline{\text{RTS}}$ signal is output when SCFRDR is empty. When $\overline{\text{RTS}}$ is 0, reception is possible. When $\overline{\text{RTS}}$ is 1, this indicates that the quantity of data stored in SCFRDR has become equal to or greater than the number set for the RTS output active trigger.

Figure 14.10 shows an example of the operation when modem control is used.

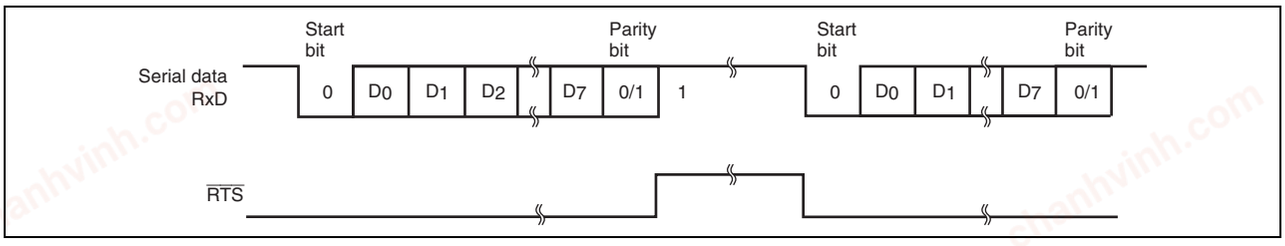


Figure 14.10 Example of Operation Using Modem Control ($\overline{\text{RTS}}$)

chanhvinh.com

14.4.3 Operation in Clock Synchronous Mode

In clock synchronous mode, data is transmitted and received in synchronization with clock pulses. This mode is suitable for high-speed serial communication.

The transmitter and receiver in this module are independent, so full-duplex communication is possible while sharing the same clock. The transmitter and receiver are also 16-stage FIFO buffered, so continuous transmitting or receiving is possible by reading or writing data while transmitting or receiving is in progress.

Figure 14.11 shows the general format in clock synchronous serial communication.

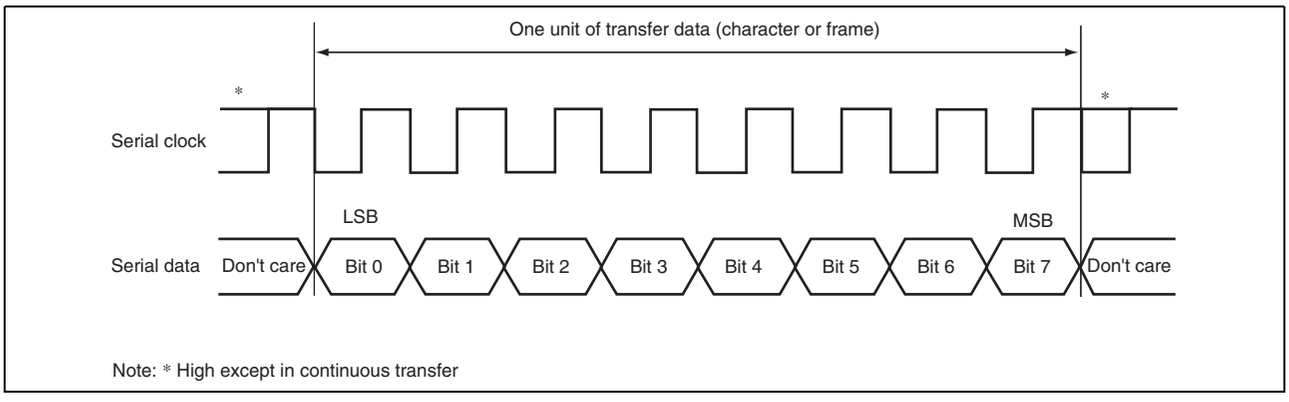


Figure 14.11 Data Format in Clock Synchronous Communication

In clock synchronous serial communication, each data bit is output on the communication line from one falling edge of the serial clock to the next. Data is guaranteed valid at the rising edge of the serial clock.

In each character, the serial data bits are transmitted in order from the LSB (first) to the MSB (last). After output of the MSB, the communication line remains in the state of the MSB.

In clock synchronous mode, data is received in synchronization with the rising edge of the synchronous clock.

chanhvinh.com

(1) Transmit/Receive Formats

The data length is fixed at eight bits. No parity bit can be added.

(2) Clock

An internal clock generated by the on-chip baud rate generator or an external synchronous clock input from the SCK pin can be selected by setting the C/$\overline{A}$ bit in SCSMR and the CKE[1:0] bits in SCSCR,.

When this module operates on an internal clock, it outputs the synchronous clock signal at the SCK pin. Eight clock pulses are output per one character transmission or reception. When transmission or reception is not performed, the clock signal remains in the high state. When only receiving is performed with the internal clock selected, the clock signal pulses are output while the RE bit of SCSCR is 1 and the number of data in receive FIFO is more than the receive FIFO data trigger number.

(3) Transmitting and Receiving Data

- Initialization (Clock Synchronous Mode)
  Before transmitting, receiving, or changing the mode or communication format, the software must clear the TE and RE bits to 0 in the serial control register (SCSCR), and then initialize this module following the procedure described below.

  Clearing TE to 0 initializes the transmit shift register (SCTSR). Clearing RE to 0, however, does not initialize the RDF, PER, FER, and ORER flags and receive data register (SCRDR), which retain their previous contents.

chanhvinh.com

Figure 14.12 shows a sample flowchart for initialization.

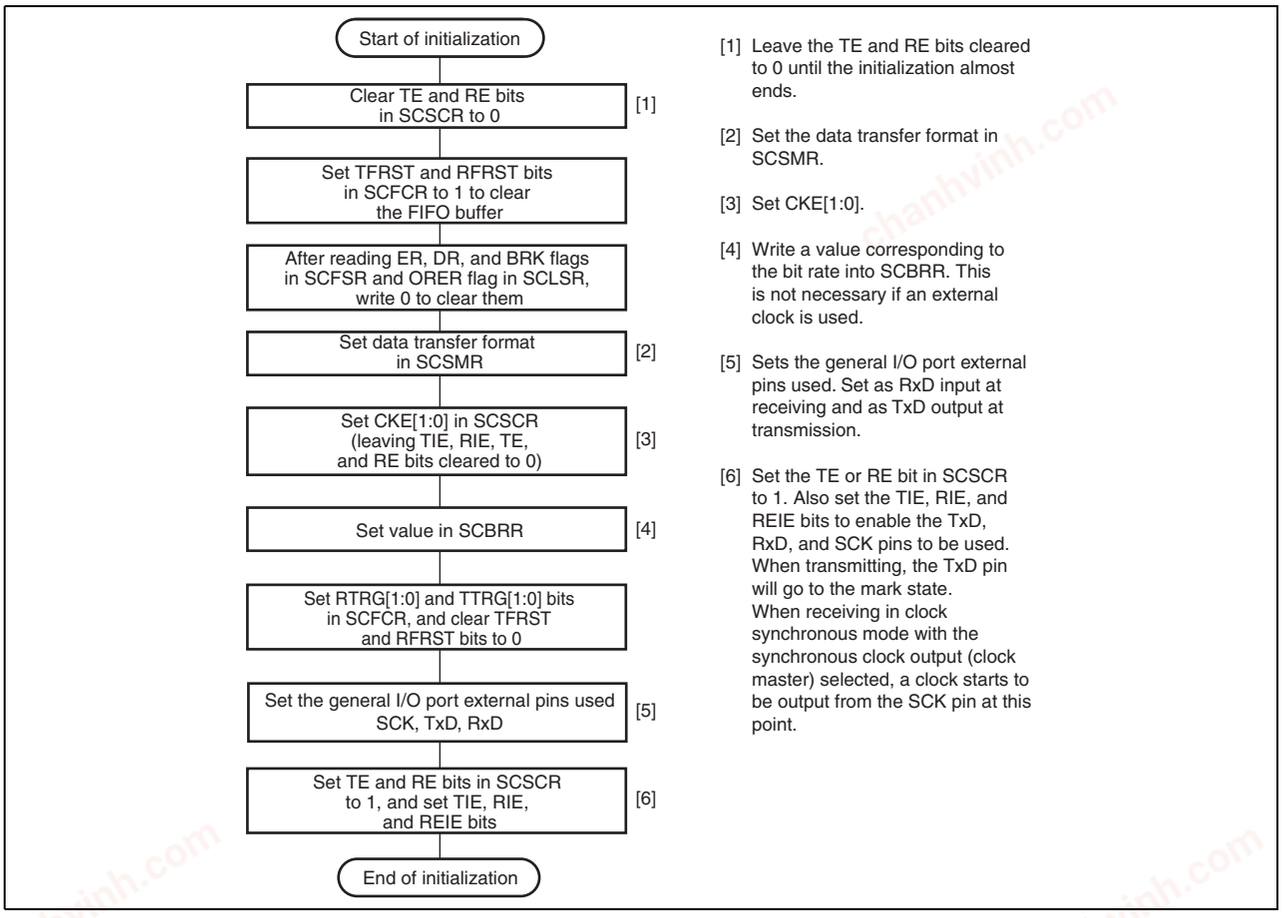


Figure 14.12 Sample Flowchart for Initialization

chanhvinh.com

- Transmitting Serial Data (Clock Synchronous Mode)

  Figure 14.13 shows a sample flowchart for transmitting serial data.

  Use the following procedure for serial data transmission after enabling transmit operation.

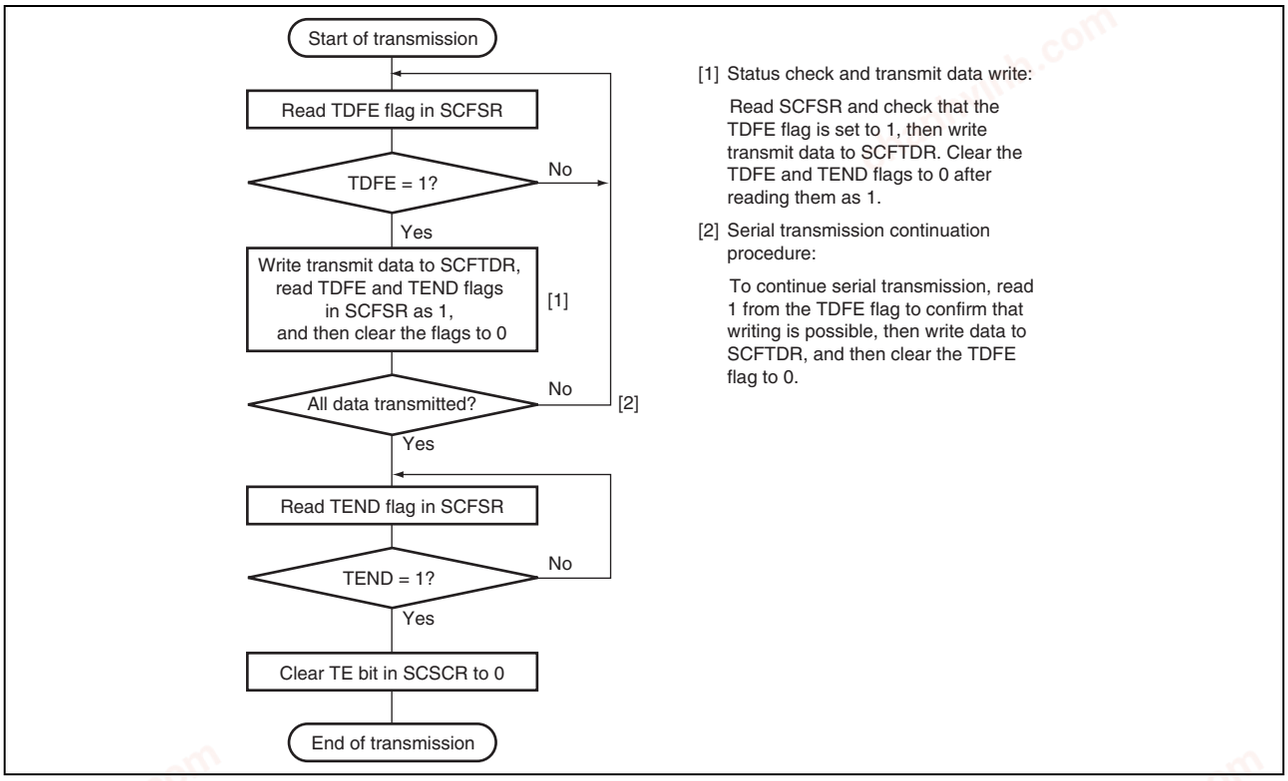


Figure 14.13 Sample Flowchart for Transmitting Serial Data

chanhvinh.com

In serial transmission, this module operates as described below.

1. When data is written into the transmit FIFO data register (SCFTDR), the data is transferred from SCFTDR to the transmit shift register (SCTSR). Confirm that the TDFE flag in the serial status register (SCFSR) is set to 1 before writing transmit data to SCFTDR. The number of transmit data bytes that can be written is (16 – transmit trigger setting).

2. When data is transferred from SCFTDR to SCTSR and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCFTDR. When the number of transmit data bytes in SCFTDR falls below the transmit trigger number set in the FIFO control register (SCFCR), the TDFE flag is set. If the TIE bit in the serial control register (SCSCR) is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.
   If clock output mode is selected, eight synchronous clock pulses are output. If an external clock source is selected, data is output in synchronization with the input clock. Data is output from the TxD pin in order from the LSB (bit 0) to the MSB (bit 7).

3. The SCFTDR transmit data is checked at the timing for sending the MSB (bit 7). If data is present, the data is transferred from SCFTDR to SCTSR, and then serial transmission of the next frame is started. If there is no data, the TxD pin holds the state after the TEND flag in SCFSR is set to 1 and the MSB (bit 7) is sent.

4. After the end of serial transmission, the SCK pin is held in the high state.

Figure 14.14 shows an example of transmit operation.

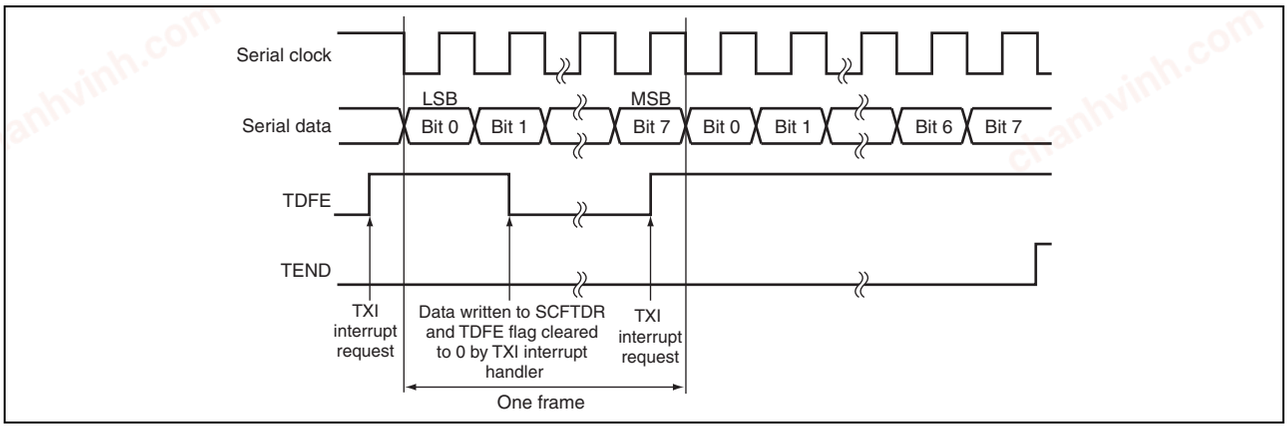


Figure 14.14 Example of Transmit Operation

chanhvinh.com

- Receiving Serial Data (Clock Synchronous Mode)

  Figure 14.15 and Figure 14.16 show sample flowcharts for receiving serial data. Use the following procedure for serial data reception after enabling receive operation. When switching from asynchronous mode to clock synchronous mode without initialization, make sure that ORER, PER, and FER are cleared to 0.

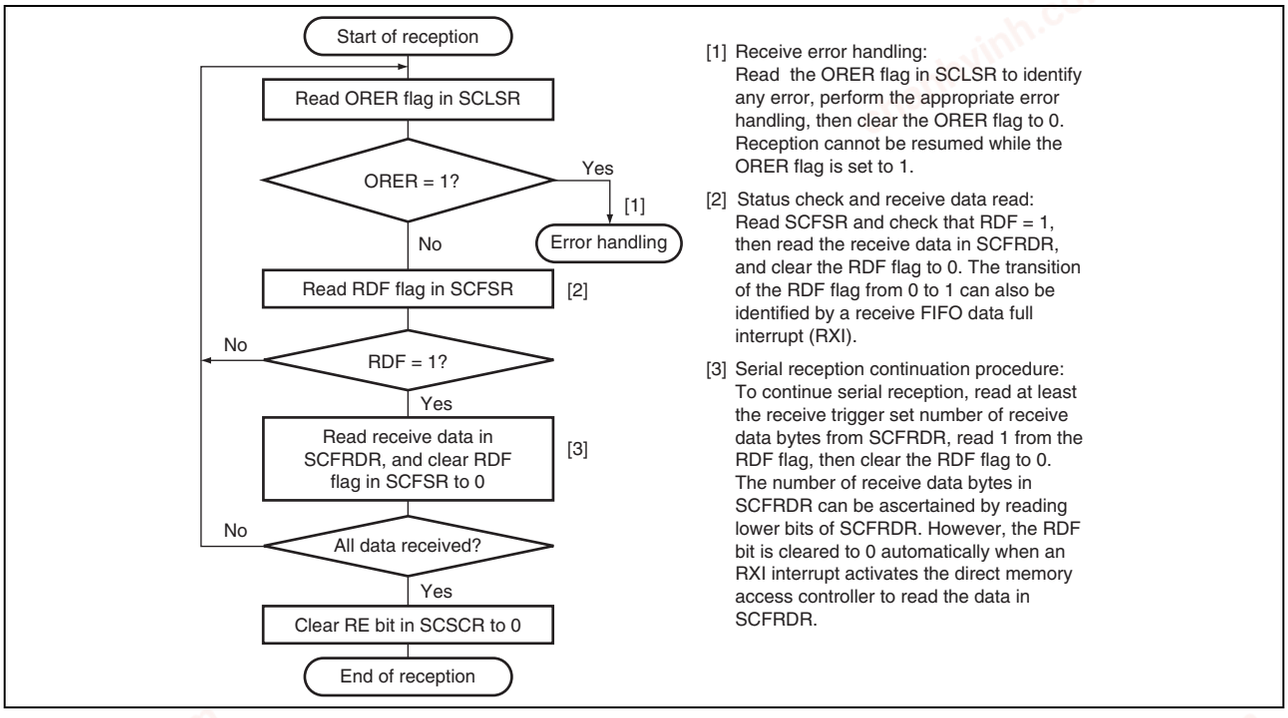


Figure 14.15 Sample Flowchart for Receiving Serial Data (1)

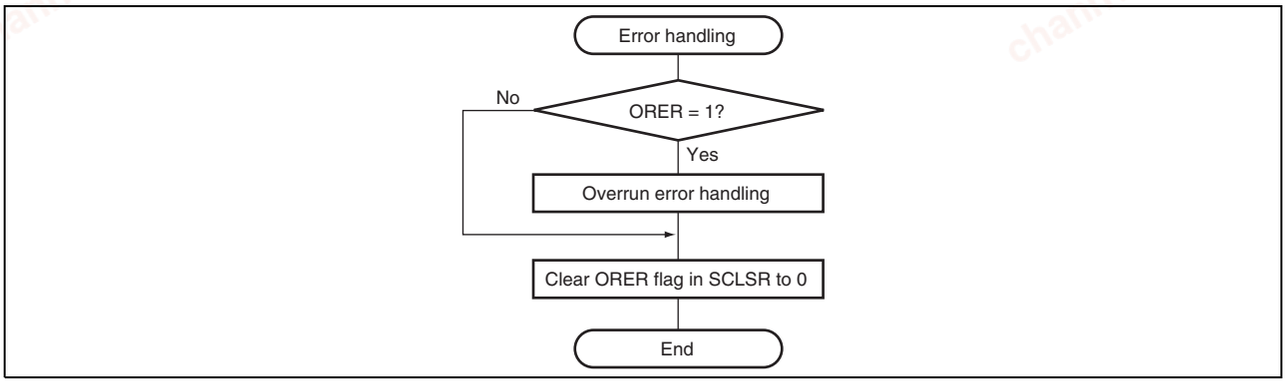


Figure 14.16 Sample Flowchart for Receiving Serial Data (2)

chanhvinh.com

In serial reception, this module operates as described below.

1. Reception is started in synchronization with synchronous clock input or output.

2. Receive data is shifted into SCRSR in order from the LSB to the MSB. After the data reception, whether the receive data can be loaded from SCRSR into SCFRDR or not is checked. If this check is passed, the RDF flag is set to 1 and the received data is stored in SCFRDR. If the check is not passed (overrun error is detected), further reception is prevented.

3. After setting RDF to 1, if the receive interrupt enable bit (RIE) is set to 1 in SCSCR, a receive FIFO data full interrupt (RXI) request is generated. If the ORER bit is set to 1 and the receive interrupt enable bit (RIE) or the receive error interrupt enable bit (REIE) in SCSCR is also set to 1, a break interrupt (BRI) request is generated.

Figure 14.17 shows an example of receive operation.

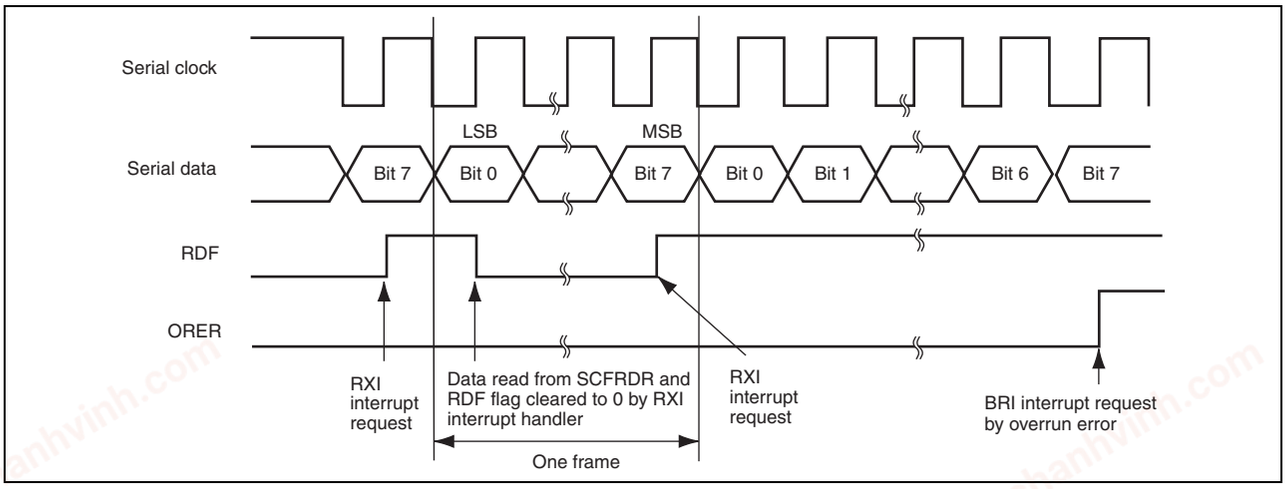


Figure 14.17 Example of Receive Operation

chanhvinh.com

- Transmitting and Receiving Serial Data Simultaneously (Clock Synchronous Mode)

  Figure 14.18 shows a sample flowchart for transmitting and receiving serial data simultaneously.
  Use the following procedure for the simultaneous transmission/reception of serial data, after enabling transmit/receive operation.

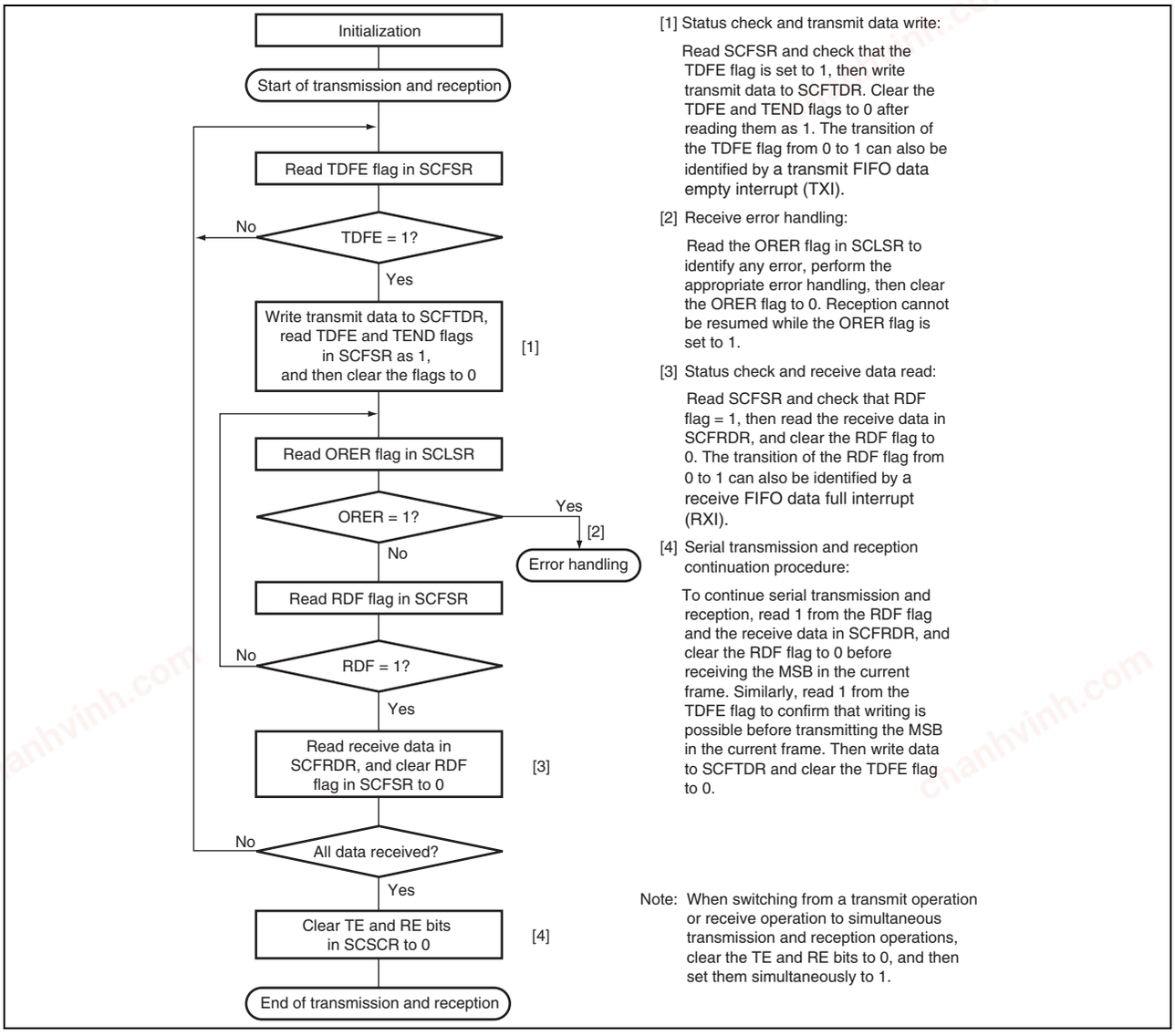


Figure 14.18 Sample Flowchart for Transmitting/Receiving Serial Data

chanhvinh.com

14.5 Interrupts

This module has four interrupt sources: transmit-FIFO-data-empty (TXI), receive-error (ERI), receive FIFO data full (RXI), and break (BRI).

Table 14.12 shows the interrupt sources and their order of priority. The interrupt sources are enabled or disabled by means of the TIE, RIE, and REIE bits in SCSCR. A separate interrupt request is sent to the interrupt controller for each of these interrupt sources.

When a TXI request is enabled by the TIE bit and the TDFE flag in the serial status register (SCFSR) is set to 1, a TXI interrupt request is generated. The direct memory access controller can be activated and data transfer performed by this TXI interrupt request. At this time, an interrupt request is not sent to the CPU.

When an RXI request is enabled by the RIE bit and the RDF flag or the DR flag in SCFSR is set to 1, an RXI interrupt request is generated. The direct memory access controller can be activated and data transfer performed by this RXI interrupt request. At this time, an interrupt request is not sent to the CPU. The RXI interrupt request caused by the DR flag is generated only in asynchronous mode.

When the RIE bit is set to 0 and the REIE bit is set to 1, this module requests only an ERI or a BRI interrupt without requesting an RXI interrupt.

The TXI indicates that transmit data can be written, and the RXI indicates that there is receive data in SCFRDR.

Table 14.12 Interrupt Sources

Interrupt Source	Description	Direct Memory Access Controller Activation	Priority on Reset Release
BRI	Interrupt initiated by break (BRK) or overrun error (ORER)	Not possible	High
ERI	Interrupt initiated by receive error (ER)	Not possible	↕
RXI	Interrupt initiated by receive FIFO data full (RDF) or data ready (DR)	Possible	↕
TXI	Interrupt initiated by transmit FIFO data empty (TDFE)	Possible	Low

chanhvinh.com

14.6 Usage Notes

Note the following when using this module.

14.6.1 SCFTDR Writing and TDFE Flag

The TDFE flag in the serial status register (SCFSR) is set when the number of transmit data bytes written in the transmit FIFO data register (SCFTDR) has fallen below the transmit trigger number set by bits TTRG[1:0] in the FIFO control register (SCFCR). After the TDFE flag is set, transmit data up to the number of empty bytes in SCFTDR can be written, allowing efficient continuous transmission.

However, if the number of data bytes written in SCFTDR is equal to or less than the transmit trigger number, the TDFE flag will be set to 1 again after being read as 1 and cleared to 0. The TDFE flag should therefore be cleared to 0 after being read as 1 when SCFTDR contains more than the transmit trigger number of transmit data bytes.

The number of transmit data bytes in SCFTDR can be found from the upper 8 bits of the FIFO data count register (SCFDR).

14.6.2 SCFRDR Reading and RDF Flag

The RDF flag in the serial status register (SCFSR) is set when the number of receive data bytes in the receive FIFO data register (SCFRDR) has become equal to or greater than the receive trigger number set by bits RTRG[1:0] in the FIFO control register (SCFCR). After RDF flag is set, receive data equivalent to the trigger number can be read from SCFRDR, allowing efficient continuous reception.

However, if the number of data bytes in SCFRDR exceeds the trigger number, the RDF flag will be set to 1 again after being read as 1 and then cleared to 0. The RDF flag should therefore be cleared to 0 after being read as 1 after reading the number of the received data in the receive FIFO data register (SCFRDR) which is less than the trigger number.

The number of receive data bytes in SCFRDR can be found from the lower 8 bits of the FIFO data count register (SCFDR).

14.6.3 Restriction on Direct Memory Controller Usage

When the direct memory access controller writes data to SCFTDR due to a TXI interrupt request, the state of the TEND flag becomes undefined. Therefore, the TEND flag should not be used as the transfer end flag in such a case.

14.6.4 Break Detection and Processing

Break signals can be detected by reading the RxD pin directly when a framing error (FER) is detected. In the break state the input from the RxD pin consists of all 0s, so the FER flag is set and the parity error flag (PER) may also be set.

Note that, although transfer of receive data to SCFRDR is halted in the break state, the receive operation is continued.

chanhvinh.com

14.6.5 Sending a Break Signal

The I/O condition and level of the TxD pin are determined by the SPB2IO and SPB2DT bits in the serial port register (SCSPTR). This feature can be used to send a break signal.

Until TE bit is set to 1 (enabling transmission) after initializing, the TxD pin does not work. During the period, mark status is performed by the SPB2DT bit. Therefore, the SPB2IO and SPB2DT bits should be set to 1 (high level output).

To send a break signal during serial transmission, clear the SPB2DT bit to 0 (designating low level), then clear the TE bit to 0 (halting transmission). When the TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission state, and 0 is output from the TxD pin.

14.6.6 Receive Data Sampling Timing and Receive Margin (Asynchronous Mode)

This module operates on a base clock with a frequency 16 or 8 times the bit rate. In reception, the falling edge of the start bit is sampled at the base clock to perform synchronization internally. Receive data is latched at the rising edge of the eighth or fourth base clock pulse. When this module operates on a base clock with a frequency 16 times the bit rate, the receive data is sampled at the timing shown in Figure 14.19.

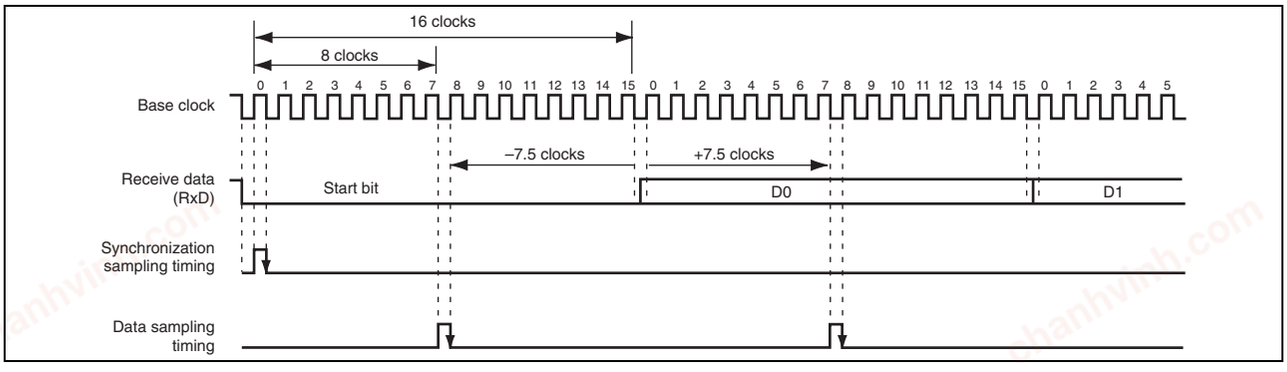


Figure 14.19 Receive Data Sampling Timing in Asynchronous Mode (Operation on a Base Clock with a Frequency 16 Times the Bit Rate)

The receive margin in asynchronous mode can therefore be expressed as shown in equation 1.

Equation 1:

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\ \%$$

Where: M: Receive margin (%)
N: Ratio of clock frequency to bit rate (N = 16 or 8)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

From equation 1, if F = 0, D = 0.5 and N = 16, the receive margin is 46.875%, as given by equation 2.

chanhvinh.com

Equation 2:

When D = 0.5 and F = 0:

$M = (0.5 - 1/(2 \times 16)) \times 100\%$

$= 46.875\%$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

14.6.7 Selection of Base Clock in Asynchronous Mode

In this LSI, when asynchronous mode is selected, the base clock frequency within a bit period can be set to the frequency 16 or 8 times the bit rate by setting the ABCS bit in SCEMR.

Note that, however, if the base clock frequency 8 times the bit rate is used, receive margin is decreased as calculated using equation 1 in section 14.6.6, Receive Data Sampling Timing and Receive Margin (Asynchronous Mode).

If the desired bit rate can be set simply by setting SCBRR and the CKS[1:0] bits in SCSMR, it is recommended to use the base clock frequency within a bit period 16 times the bit rate (by setting the ABCS bit in SCEMR to 0). If an internal clock is selected as a clock source and the SCK pin is not used, the bit rate can be increased without decreasing receive margin by selecting double-speed mode for the baud rate generator (setting the BGDM bit in SCEMR to 1).

chanhvinh.com

15. Serial Communications Interface

This LSI has two independent serial communications interface (SCI) channels.
The SCI can handle both asynchronous and clock synchronous serial communications.
Asynchronous serial data communications can be carried out with standard asynchronous communications chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communications Interface Adapter (ACIA).
As an extended function in asynchronous communications mode, the SCI also supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards).

15.1 Overview

Table 15.1 lists the specifications of the SCI module.
Figure 15.1 is a block diagram of the SCI module.

Table 15.1 Specifications of SCI

Item		Specifications
Serial communications mode		• Asynchronous operation • Clock synchronous operation • Smart card interface
Transfer speed		Bit rate specifiable with on-chip baud rate generator.
Full-duplex communications		Transmitter: Enables continuous transmission by double-buffering. Receiver: Enables continuous reception by double-buffering.
I/O pins		See Table 15.2.
Data transfer		Selectable as LSB-first or MSB-first transfer
Interrupt sources		Transmit-end, transmit-data-empty, receive-data-full, and receive error
Power consumption reduction function		Module-standby state can be set for each channel.
Asynchronous mode	Data length	7 or 8 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even, odd, or none
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	CTSn and RTSn pins can be used in transfer control.
	Break detection	Break can be detected by reading RXDn pin level directly in case of a framing error
	Clock source	Selectable from internal or external clock
	Multi-processor communications function	Serial communication among multiple processors
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters.
Clock synchronous mode	Data length	8 bits
	Receive error detection	Overrun errors
	Hardware flow control	CTSn and RTSn pins can be used in transfer control.
Smart card interface mode	Error processing	An error signal can be automatically transmitted on detection of a parity error during reception
		Data can be automatically re-transmitted on receiving an error signal during transmission
	Data type	Both direct convention and inverse convention are supported.

chanhvinh.com

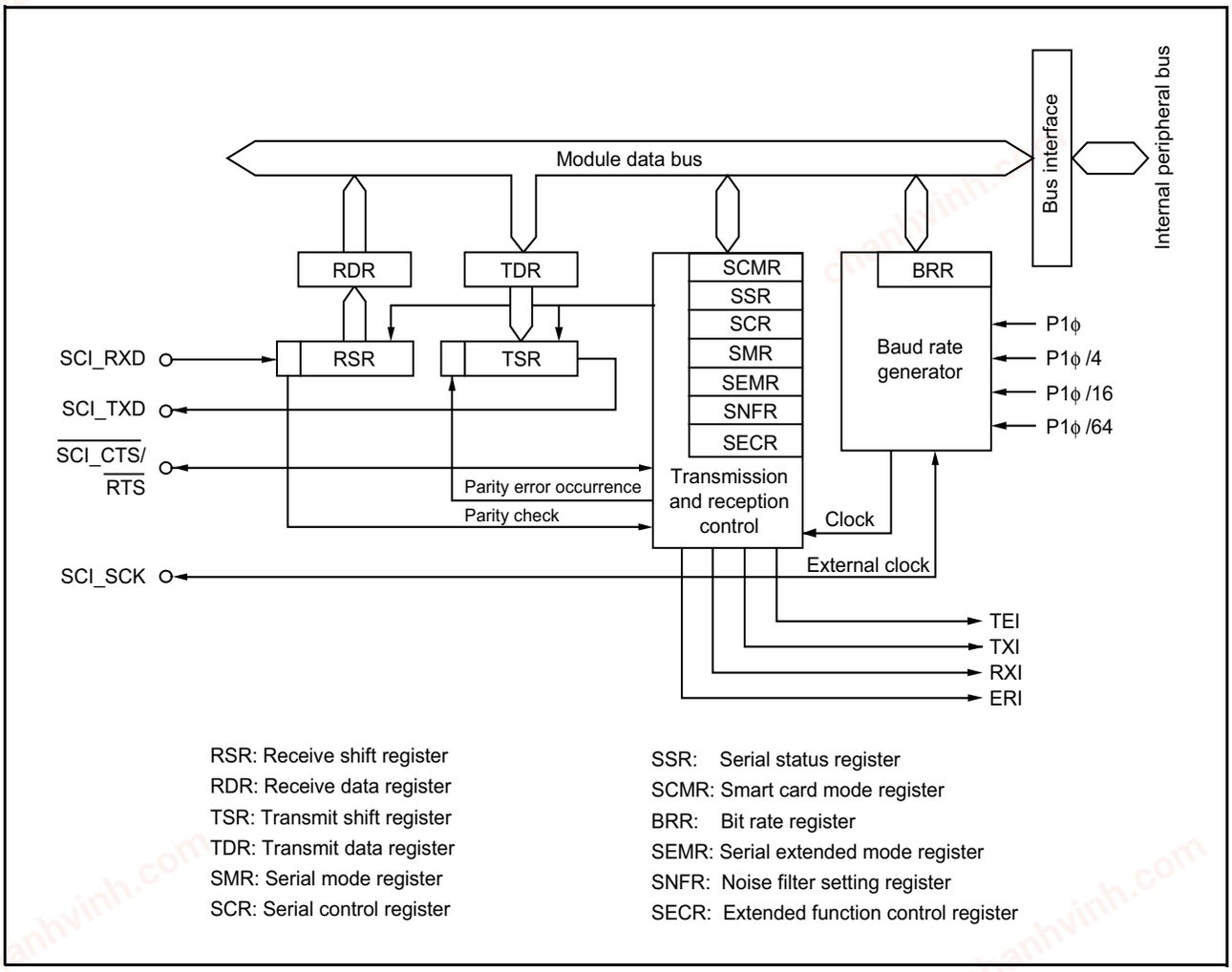


Figure 15.1 Block Diagram of SCI

Table 15.2 lists the pin configuration of the SCIs.

Table 15.2 Input and Output Pins of the SCIs

Channel	Pin Name	I/O	Function
SCI0	SCI_SCK0	I/O	SCI0 clock input/output
	SCI_RXD0	Input	SCI0 receive data input
	SCI_TXD0	Output	SCI0 transmit data output
	$\overline{\text{SCI_CTS0}}$/$\overline{\text{RTS0}}$	I/O	SCI0 transfer start control input/output
SCI1	SCI_SCK1	I/O	SCI1 clock input/output
	SCI_RXD1	Input	SCI1 receive data input
	SCI_TXD1	Output	SCI1 transmit data output
	$\overline{\text{SCI_CTS1}}$/$\overline{\text{RTS1}}$	I/O	SCI1 transfer start control input/output

Note: • These pins are referred to as SCKn, RXDn, TXDn, RTSn#, and CTSn# in the text in this section.

chanhvinh.com

15.2 Register Descriptions

Table 15.3 is a list of registers.

Table 15.3 List of Registers

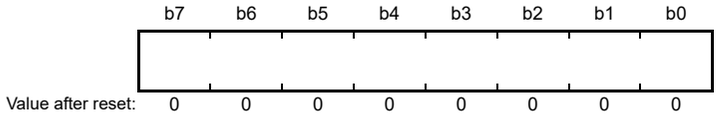
Channel	Register Name	Symbol	Value after Reset	Address	Access size
0	Serial mode register 0	SMR0	H'00	H'E800B000	8
	Bit rate register 0	BRR0	H'FF	H'E800B001	8
	Serial control register 0	SCR0	H'00	H'E800B002	8
	Transmit data register 0	TDR0	H'FF	H'E800B003	8
	Serial status register 0	SSR0	H'84	H'E800B004	8
	Receive data register 0	RDR0	H'00	H'E800B005	8
	Smart card mode register 0	SCMR0	H'F2	H'E800B006	8
	Serial extended mode register 0	SEMR0	H'00	H'E800B007	8
	Noise filter setting register 0	SNFR0	H'00	H'E800B008	8
	Extended function control register 0	SECR0	H'00	H'E800B00D	8
1	Serial mode register 1	SMR1	H'00	H'E800B800	8
	Bit rate register 1	BRR1	H'FF	H'E800B801	8
	Serial control register 1	SCR1	H'00	H'E800B802	8
	Transmit data register 1	TDR1	H'FF	H'E800B803	8
	Serial status register 1	SSR1	H'84	H'E800B804	8
	Receive data register 1	RDR1	H'00	H'E800B805	8
	Smart card mode register 1	SCMR1	H'F2	H'E800B806	8
	Serial extended mode register 1	SEMR1	H'00	H'E800B807	8
	Noise filter setting register 1	SNFR1	H'00	H'E800B808	8
	Extended function control register 1	SECR1	H'00	H'E800B80D	8

chanhvinh.com

15.2.1 Receive Shift Register (RSR)

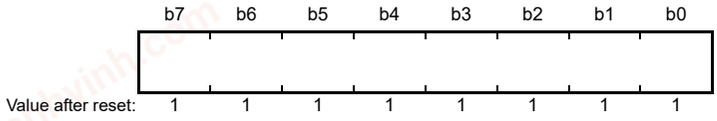
RSR is a shift register which is used to receive serial data input from the RXDn pin and converts it into parallel data. When one frame of data has been received, it is transferred to RDR automatically.
RSR cannot be directly accessed by the CPU.

15.2.2 Receive Data Register (RDR)



RDR is an 8-bit register that stores receive data.
When the SCI has received one frame of serial data, it transfers the received serial data from RSR to RDR where it is stored. This allows RSR to receive the next data.
Since RSR and RDR function as a double buffer in this way, continuous receive operations can be performed.
Only read RDR once after each instance of the receive data full interrupt (RXI). Note that if next one frame of data is received before reading receive data from RDR, an overrun error occurs.
RDR cannot be written to by the CPU.

15.2.3 Transmit Data Register (TDR)



TDR is an 8-bit register that stores transmit data.
When the SCI detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts transmission.
The double-buffered structures of TDR and TSR enable continuous serial transmission. If the next transmit data has already been written to TDR when one frame of data is transmitted, the SCI transfers the written data to TSR to continue transmission.
The CPU is able to read from or write to TDR at any time. Only write data for transmission to TDR once after each instance of the transmit data empty interrupt (TXI).

15.2.4 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data.
To perform serial data transmission, the SCI first automatically transfers transmit data from TDR to TSR, and then sends the data to the TXDn pin.
TSR cannot be directly accessed by the CPU.

chanhvinh.com

15.2.5 Serial Mode Register (SMR)

Note: • Some bits in SMR have different functions in serial communications interface mode and smart card interface mode.

(1) Serial Communications Interface Mode (SMIF in SCMR = 0)

	b7	b6	b5	b4	b3	b2	b1	b0
	CM	CHR	PE	PM	STOP	MP	CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: P1ϕ clock (n = 0)*1 0 1: P1ϕ/4 clock (n = 1)*1 1 0: P1ϕ/16 clock (n = 2)*1 1 1: P1ϕ/64 clock (n = 3)*1	R/W*4
b2	MP	Multi-Processor Mode	(Valid only in asynchronous mode) 0: Multi-processor communications function is disabled 1: Multi-processor communications function is enabled	R/W*4
b3	STOP	Stop Bit Length	(Valid only in asynchronous mode) 0: 1 stop bit 1: 2 stop bits	R/W*4
b4	PM	Parity Mode	(Valid only when the PE bit is 1 in asynchronous mode) 0: Selects even parity 1: Selects odd parity	R/W*4
b5	PE	Parity Enable	(Valid only in asynchronous mode) • When transmitting 0: Parity bit addition is not performed 1: The parity bit is added • When receiving 0: Parity bit checking is not performed 1: The parity bit is checked	R/W*4
b6	CHR	Character Length	(Valid only in asynchronous mode) 0: Selects 8 bits as the data length*2 1: Selects 7 bits as the data length*3	R/W*4
b7	CM	Communications Mode	0: Asynchronous mode 1: Clock synchronous mode	R/W*4

Note 1. n is the decimal notation of the value of n in BRR (see section 15.2.9, Bit Rate Register (BRR)).
Note 2. In clock synchronous mode, this bit setting is invalid and a fixed data length of 8 bits is used.
Note 3. LSB-first is fixed and the MSB (bit 7) in TDR is not transmitted in transmission.
Note 4. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).

CKS[1:0] Bits (Clock Select)

These bits select the clock source for the on-chip baud rate generator.
For the relation between the settings of these bits and the baud rate, see section 15.2.9, Bit Rate Register (BRR).

MP Bit (Multi-Processor Mode)

Disables/enables the multi-processor communications function. The settings of the PE bit and PM bit are invalid in multi-processor mode.

STOP Bit (Stop Bit Length)

Selects the stop bit length in transmission.
In reception, only the first stop bit is checked regardless of this bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit frame.

chanhvinh.com

PM Bit (Parity Mode)

Selects the parity mode (even or odd) for transmission and reception.
The setting of the PM bit is invalid in multi-processor mode.

PE Bit (Parity Enable)

When this bit is set to 1, the parity bit is added to transmit data, and the parity bit is checked in reception.
Irrespective of the setting of the PE bit, the parity bit is not added or checked in multi-processor format.

CHR Bit (Character Length)

Selects the data length for transmission and reception.
In clock synchronous mode, a fixed data length of 8 bits is used.

(2) Smart Card Interface Mode (SMIF in SCMR = 1)

	b7	b6	b5	b4	b3	b2	b1	b0
	GM	BLK	PE	PM	BCP[1:0]		CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: P1ϕ clock (n = 0)*1 0 1: P1ϕ/4 clock (n = 1)*1 1 0: P1ϕ/16 clock (n = 2)*1 1 1: P1ϕ/64 clock (n = 3)*1	R/W*3
b3, b2	BCP[1:0]	Base Clock Pulse	Selects the number of base clock cycles in combination with the BCP2 bit in SCMR. Setting values in BCP2 bit in SCMR and BCP[1:0] bits in SMR: BCP2 b3 b2 0 0 0: 93 clock cycles (S = 93)*2 0 0 1: 128 clock cycles (S = 128)*2 0 1 0: 186 clock cycles (S = 186)*2 0 1 1: 512 clock cycles (S = 512)*2 1 0 0: 32 clock cycles (S = 32)*2 (Initial value) 1 0 1: 64 clock cycles (S = 64)*2 1 1 0: 372 clock cycles (S = 372)*2 1 1 1: 256 clock cycles (S = 256)*2	R/W*3
b4	PM	Parity Mode	(Valid only when the PE bit is 1 in asynchronous mode) 0: Selects even parity 1: Selects odd parity	R/W*3
b5	PE	Parity Enable	When this bit is set to 1, a parity bit is added to data for transmission, and the parity of received data is checked. Set this bit to 1 in smart card interface mode.	R/W*3
b6	BLK	Block Transfer Mode	0: Normal mode operation 1: Block transfer mode operation	R/W*3
b7	GM	GSM Mode	0: Normal mode operation 1: GSM mode operation	R/W*3

Note 1. n is the decimal notation of the value of n in BRR (see section 15.2.9, Bit Rate Register (BRR)).
Note 2. S is the value of S in BRR (see section 15.2.9, Bit Rate Register (BRR)).
Note 3. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).

CKS[1:0] Bits (Clock Select)

These bits select the clock source for the on-chip baud rate generator.
For the relationship between the settings of these bits and the baud rate, see section 15.2.9, Bit Rate Register (BRR).

chanhvinh.com

BCP[1:0] Bits (Base Clock Pulse)

These bits select the number of base clock cycles in a 1-bit data transfer time in smart card interface mode.
Set these bits in combination with the BCP2 bit in SCMR.
For details, see section 15.6.4, Receive Data Sampling Timing and Reception Margin.

PM Bit (Parity Mode)

Selects the parity mode for transmission and reception (even or odd).
For details on the usage of this bit in smart card interface mode, see section 15.6.2, Data Format (Except in Block Transfer Mode).

PE Bit (Parity Enable)

Set the PE bit to 1.
The parity bit is added to transmit data before transmission, and the parity bit is checked in reception.

BLK Bit (Block Transfer Mode)

Setting this bit to 1 allows block transfer mode operation.
For details, see section 15.6.3, Block Transfer Mode.

GM Bit (GSM Mode)

Setting this bit to 1 allows GSM mode operation.
In GSM mode, the SSR.TEND flag set timing is put forward to 11.0 etu (elementary time unit = 1-bit transfer time) from the start and the clock output control function is appended. For details, see section 15.6.6, Serial Data Transmission (Except in Block Transfer Mode) and section 15.6.8, Clock Output Control.

chanhvinh.com

15.2.6 Serial Control Register (SCR)

Note: • Some bits in SCR have different functions in serial communications interface mode and smart card interface mode.

(1) Serial Communications Interface Mode (SMIF in SCMR = 0)

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKE[1:0]	Clock Enable	(Asynchronous mode) b1 b0 0 0: On-chip baud rate generator The SCKn pin functions as general-purpose I/O port. 0 1: On-chip baud rate generator The clock with the same frequency as the bit rate is output from the SCKn pin. 1 x: External clock SEMR.ABCS bit is 0: The clock with a frequency 16 times the bit rate should be input from the SCKn pin. SEMR.ABCS bit is 1: The clock with a frequency eight times the bit rate should be input from the SCKn pin. (Clock synchronous mode) b1 b0 0 x: Internal clock The SCKn pin functions as the clock output pin. 1 x: External clock The SCKn pin functions as the clock input pin.	R/W*1
b2	TEIE	Transmit End Interrupt Enable	0: A TEI interrupt request is disabled 1: A TEI interrupt request is enabled	R/W
b3	MPIE	Multi-Processor Interrupt Enable	(Valid in asynchronous mode when SMR.MP = 1) 0: Normal reception 1: When the data with the multi-processor bit set to 0 is received, the data is not read, and setting the status flags ORER and FER in SSR to 1 is disabled. When the data with the multi-processor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and normal reception is resumed.	R/W
b4	RE	Receive Enable	0: Serial reception is disabled 1: Serial reception is enabled	R/W*2
b5	TE	Transmit Enable	0: Serial transmission is disabled 1: Serial transmission is enabled	R/W*2
b6	RIE	Receive Interrupt Enable	0: RXI and ERI interrupt requests are disabled 1: RXI and ERI interrupt requests are enabled	R/W
b7	TIE	Transmit Interrupt Enable	0: A TXI interrupt request is disabled 1: A TXI interrupt request is enabled	R/W

x: Don't care

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. A 1 can be written only when TE = 0 and RE = 0, while the SMR.CM bit is 1. After setting TE or RE to 1, only 0 can be written in TE and RE. While the SMR.CM bit is 0, writing is enabled under any condition.

chanhvinh.com

CKE[1:0] Bits (Clock Enable)

These bits select the clock source and SCKn pin function.

TEIE Bit (Transmit End Interrupt Enable)

Enables or disables a TEI interrupt request.

A TEI interrupt request is disabled by clearing the TEIE bit to 0.

MPIE Bit (Multi-Processor Interrupt Enable)

When this bit is set to 1 and the data with the multi-processor bit set to 0 is received, the data is not read and setting the status flags ORER and FER in SSR to 1 is disabled. When the data with the multi-processor bit set to 1 is received, the MPIE is automatically cleared to 0, and normal reception is resumed. For details, see section 15.4, Multi-Processor Communications Function.

When the receive data includes the MPB bit is SSR set to 0, the receive data is not transferred from the RSR to the RDR, a receive error is not detected, and setting the flags ORER and FER to 1 is disabled.

When the receive data includes the MPB bit set to 1, the MPB bit is set to 1, the MPIE bit is automatically cleared to 0, the RXI and ERI interrupt requests are enabled (if the RIE bit in SCR is set to 1), and setting the flags ORER and FER to 1 is enabled.

MPIE should be set to 0 if multi-processor communications function is not to be used.

RE Bit (Receive Enable)

Enables or disables serial reception.

When this bit is set to 1, serial reception is started by detecting the start bit in asynchronous mode or the synchronous clock input in clock synchronous mode. Note that SMR should be set prior to setting the RE bit to 1 in order to designate the reception format.

Even if reception is halted by clearing the RE bit to 0, the ORER, FER, and PER flags in SSR are not affected and the previous value is retained.

TE Bit (Transmit Enable)

Enables or disables serial transmission.

When this bit is set to 1, serial transmission is started by writing transmit data to TDR. Note that SMR should be set prior to setting the TE bit to 1 in order to designate the transmission format.

RIE Bit (Receive Interrupt Enable)

Enables or disables RXI and ERI interrupt requests.

An RXI interrupt request is disabled by clearing the RIE bit to 0.

An ERI interrupt request can be cancelled by reading 1 from the ORER, FER, or PER flag in SSR and then clearing the flag to 0, or clearing the RIE bit to 0.

TIE Bit (Transmit Interrupt Enable)

Enables or disables notification of a TXI interrupt request.

Notification of a TXI interrupt request is disabled by clearing the TIE bit to 0.

chanhvinh.com

(2) Smart Card Interface Mode (SMIF in SCMR = 1)

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKE[1:0]	Clock Enable	• When GM in SMR = 0 b1 b0 0 0: Output disabled (The SCKn pin is available for use as an I/O port in accord with the general-purpose I/O port settings.) 0 1: Clock output 1 x: (Setting prohibited) • When GM in SMR = 1 b1 b0 0 0: Output fixed low x 1: Clock output 1 0: Output fixed high	R/W*1
b2	TEIE	Transmit End Interrupt Enable	This bit should be 0 in smart card interface mode.	R/W
b3	MPIE	Multi-Processor Interrupt Enable	This bit should be 0 in smart card interface mode.	R/W
b4	RE	Receive Enable	0: Serial reception is disabled 1: Serial reception is enabled	R/W*2
b5	TE	Transmit Enable	0: Serial transmission is disabled 1: Serial transmission is enabled	R/W*2
b6	RIE	Receive Interrupt Enable	0: RXI and ERI interrupt requests are disabled 1: RXI and ERI interrupt requests are enabled	R/W
b7	TIE	Transmit Interrupt Enable	0: A TXI interrupt request is disabled 1: A TXI interrupt request is enabled	R/W

x: Don't care

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. A 1 can be written only when TE = 0 and RE = 0, while the SMR.CM bit is 1. After setting TE or RE to 1, only 0 can be written in TE and RE. While the SMR.CM bit is 0, writing is enabled under any condition.

For details on interrupt requests, see section 15.8, Interrupt Sources.

CKE[1:0] Bits (Clock Enable)

These bits control the clock output from the SCKn pin.

In GSM mode, clock output can be dynamically switched. For details, see section 15.6.8, Clock Output Control.

TEIE Bit (Transmit End Interrupt Enable)

This bit should be 0 in smart card interface mode.

MPIE Bit (Multi-Processor Interrupt Enable)

This bit should be 0 in smart card interface mode.

RE Bit (Receive Enable)

Enables or disables serial reception.

When this bit is set to 1, serial reception is started by detecting the start bit. Note that SMR should be set prior to setting the RE bit to 1 in order to designate the reception format.

Even if reception is halted by clearing the RE bit to 0, the ORER, FER, and PER flags in SSR are not affected and the previous value is retained.

chanhvinh.com

TE Bit (Transmit Enable)

Enables or disables serial transmission.

When this bit is set to 1, serial transmission is started by writing transmit data to TDR. Note that SMR should be set prior to setting the TE bit to 1 in order to designate the transmission format.

RIE Bit (Receive Interrupt Enable)

Enables or disables RXI and ERI interrupt requests.

An RXI interrupt request is disabled by clearing the RIE bit to 0.

An ERI interrupt request can be cancelled by reading 1 from the ORER, FER, or PER flag in SSR and then clearing the flag to 0, or clearing the RIE bit to 0.

TIE Bit (Transmit Interrupt Enable)

Enables or disables notification of a TXI interrupt request.

Notification of a TXI interrupt request is disabled by clearing the TIE bit to 0.

chanhvinh.com

15.2.7 Serial Status Register (SSR)

Note: • Some bits in SSR have different functions in serial communications interface mode and smart card interface mode.

(1) Serial Communications Interface Mode (SMIF in SCMR = 0)

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	ORER	FER	PER	TEND	MPB	MPBT
Value after reset:	x	x	0	0	0	1	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	MPBT	Multi-Processor Bit Transfer	Sets the multi-processor bit for adding to the transmission frame 0: Data transmission cycles 1: ID transmission cycles	R/W
b1	MPB	Multi-Processor	Value of the multi-processor bit in the reception frame 0: Data transmission cycles 1: ID transmission cycles	R
b2	TEND	Transmit End Flag	0: A character is being transmitted. 1: Character transfer has been completed.	R
b3	PER	Parity Error Flag	0: No parity error occurred 1: A parity error has occurred	R/(W) *1
b4	FER	Framing Error Flag	0: No framing error occurred 1: A framing error has occurred	R/(W) *1
b5	ORER	Overrun Error Flag	0: No overrun error occurred 1: An overrun error has occurred	R/(W) *1
b7, b6	—	Reserved	The read value is undefined. The write value should be 1.	R

Note 1. Only 0 can be written to this bit, to clear the flag.

MPB Bit (Multi-Processor)

Holds the value of the multi-processor bit in the reception frame. This bit does not change when the RE bit in SCR is 0.

TEND Flag (Transmission End Flag)

Indicates completion of transmission.

[Setting conditions]

- Clearing of the SCR.TE bit to 0 (disabling serial transmission operations)
  When the SCR.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- The TDR is not updated at the time of transmission of the tail-end bit of a character being transmitted

[Clearing condition]

- When data for transmission are written to the TDR while the SCR.TE is 1
  When the TEND flag is cleared in response to writing of data for transmission to the TDR, dummy read the TEND flag before return from interrupt exception processing for TEI interrupt requests when the latter are enabled.

chanhvinh.com

PER Bit (Parity Error Flag)

Indicates that a parity error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When a parity error is detected during reception
  Although receive data when the parity error occurs is transferred to RDR, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to PER after reading PER = 1 (after writing 0 to it, read the PER bit to check that it has actually been cleared to 0.)
  Even when the RE bit in SCR is cleared to 0 (which indicates that serial reception is disabled), the PER flag is not affected and retains its previous value.

FER Bit (Framing Error Flag)

Indicates that a framing error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When the stop bit is 0
  In 2-stop-bit mode, only the first stop bit is checked whether it is 1 but the second stop bit is not checked. Note that although receive data when the framing error occurs is transferred to RDR, no RXI interrupt request occurs. In addition, when the FER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to FER after reading FER = 1 (after writing 0 to it, read the FER bit to check that it has actually been cleared to 0.)
  Even when the RE bit in SCR is cleared to 0, the FER flag is not affected and retains its previous value.

ORER Bit (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.

[Setting condition]

- When the next data is received before receive data is read from RDR
  In RDR, receive data prior to an overrun error occurrence is retained, but data received after the overrun error occurrence is lost. When the ORER flag is set to 1, subsequent serial reception cannot be performed. Note that, in clock synchronous mode, serial transmission also cannot continue.

[Clearing condition]

- When a 0 is written to ORER after reading ORER = 1 (after writing a 0 to it, read the ORER bit to check that it has actually been cleared to 0.)
  Even when the RE bit in SCR is cleared to 0, the ORER flag is not affected and retains its previous value.

chanhvinh.com

(2) Smart Card Interface Mode (SMIF in SCMR = 1)

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	ORER	ERS	PER	TEND	MPB	MPBT
Value after reset:	x	x	0	0	0	1	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	MPBT	Multi-Processor Bit Transfer	This bit should be set to 0 in smart card interface mode.	R/W
b1	MPB	Multi-Processor	This bit is not used in smart card interface mode. It should be set to 0.	R
b2	TEND	Transmit End Flag	0: A character is being transmitted. 1: Character transfer has been completed.	R
b3	PER	Parity Error Flag	0: No parity error occurred 1: A parity error has occurred	R/(W) *1
b4	ERS	Error Signal Status Flag	0: Low error signal not responded 1: Low error signal responded	R/(W) *1
b5	ORER	Overrun Error Flag	0: No overrun error occurred 1: An overrun error has occurred	R/(W) *1
b7, b6	—	Reserved	The read value is undefined. The write value should be 1.	R

Note 1. Only 0 can be written to this bit, to clear the flag.

MPBT Bit (Multi-Processor Bit Transfer)

This bit should be set to 0 in smart card interface mode.

MPB Bit (Multi-Processor)

This bit is not used in smart card interface mode. It should be set to 0.

TEND Flag (Transmission End Flag)

With no error signal from the receiving side, this bit is set to 1 when further data for transfer is ready to be transferred to the TDR register.

[Setting conditions]

- When SCR.TE bit = 0 (disabling serial transmission operations)
  When the SCR.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When a specified period has elapsed after the latest transmission of one byte, the ERS flag is 0, and the TDR register is not updated
  The set timing is determined by register settings as listed below.
  When SMR.GM = 0 and SMR.BLK = 0, 12.5 etu after the start of transmission
  When SMR.GM = 0 and SMR.BLK = 1, 11.5 etu after the start of transmission
  When SMR.GM = 1 and SMR.BLK = 0, 11.0 etu after the start of transmission
  When SMR.GM = 1 and SMR.BLK = 1, 11.0 etu after the start of transmission

[Clearing condition]

- When data for transmission are written to the TDR while the SCR.TE is 1

chanhvinh.com

PER Flag (Parity Error Flag)

Indicates that a parity error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When a parity error is detected during reception
  Although receive data when the parity error occurs is transferred to RDR, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to PER after reading PER = 1 (After writing 0 to it, read the PER bit to check that it has actually been cleared to 0.)
  Even when the RE bit in SCR is cleared to 0 (which indicates that serial reception is disabled), the PER flag is not affected and retains its previous value.

ERS Flag (Error Signal Status Flag)

[Setting condition]

- When a low error signal is sampled

[Clearing condition]

- When 0 is written to ERS after reading ERS = 1

ORER Flag (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.

[Setting condition]

- When the next data is received before receive data is read from RDR
  In RDR, the receive data prior to an overrun error occurrence is retained, but data received following the overrun error occurrence is lost. When the ORER flag is set to 1, subsequent serial reception cannot be performed.

[Clearing condition]

- When 0 is written to ORER after reading ORER = 1 (After writing 0 to it, read the ORER bit to check that it has actually been cleared to 0.)
  Even when the RE bit in SCR is cleared to 0, the ORER flag is not affected and retains its previous value.

chanhvinh.com

15.2.8 Smart Card Mode Register (SCMR)

	b7	b6	b5	b4	b3	b2	b1	b0
	BCP2	—	—	—	SDIR	SINV	—	SMIF
Value after reset:	1	1	1	1	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	SMIF	Smart Card Interface Mode Select	0: Serial communications interface mode 1: Smart card interface mode	R/W*1
b1	—	Reserved	This bit is read as 1. The write value should be 1.	R
b2	SINV	Transmitted/Received Data Invert	0: TDR contents are transmitted as they are. Receive data is stored as it is in RDR. 1: TDR contents are inverted before being transmitted. Receive data is stored in inverted form in RDR.	R/W*1
b3	SDIR	Transmitted/Received Data Transfer Direction	0: Transfer with LSB-first 1: Transfer with MSB-first	R/W*1
b6 to b4	—	Reserved	This bit is read as 1. The write value should be 1.	R
b7	BCP2	Base Clock Pulse 2	Selects the number of base clock cycles in combination with the SMR.BCP[1:0] bits. Setting values in the SCMR.BCP2 bit and SMR.BCP[1:0] bits BCP2 BCP1 BCP0 0 0 0: 93 clock cycles (S = 93)*2 0 0 1: 128 clock cycles (S = 128)*2 0 1 0: 186 clock cycles (S = 186)*2 0 1 1: 512 clock cycles (S = 512)*2 1 0 0: 32 clock cycles (S = 32)*2 (Initial Value) 1 0 1: 64 clock cycles (S = 64)*2 1 1 0: 372 clock cycles (S = 372)*2 1 1 1: 256 clock cycles (S = 256)*2	R/W*1

Note 1. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).
Note 2. S is the value of S in BRR (see section 15.2.9, Bit Rate Register (BRR)).

SMIF Bit (Smart Card Interface Mode Select)

When this bit is set to 1, smart card interface mode is selected.
When this bit is set to 0, asynchronous or clock synchronous mode is selected.

SINV Bit (Transmitted/Received Data Invert)

Inverts the transmit/receive data logic level. This bit does not affect the logic level of the parity bit. To invert the parity bit, invert the PM bit in SMR.

SDIR Bit (Transmitted/Received Data Transfer Direction)

Selects the serial/parallel conversion format.

BCP2 Bit (Base Clock Pulse 2)

Selects the number of base clock cycles in a 1-bit data transfer time in smart card interface mode. Set this bit in combination with the SMR.BCP[1:0] bits.

chanhvinh.com

15.2.9 Bit Rate Register (BRR)



BRR is an 8-bit register that adjusts the bit rate.
As each SCI channel has independent baud-rate generator control, different bit rates can be set for each. Table 15.4 lists the relationships between the setting (N) in the BRR and the bit rate (B) for normal asynchronous mode, multi-processor transfer, clock synchronous mode, and smart card interface mode.
The initial value of BRR is FFh.
BRR can be read from by the CPU at all times, but it can be written to only when the TE and RE bits in SCR are 0.

Table 15.4 Relationships between N Setting in BRR and Bit Rate B

Mode	ABCS Bit in SEMR	BRR Setting	Error
Asynchronous, multi-processor transfer	0	$N = \frac{P1\phi \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \{\frac{P1\phi \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1\} \times 100$
	1	$N = \frac{P1\phi \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \{\frac{P1\phi \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1\} \times 100$
Clock synchronous		$N = \frac{P1\phi \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	
Smart card interface		$N = \frac{P1\phi \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$Error\ (\%) = \{\frac{P1\phi \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1\} \times 100$

B: Bit rate (bps)
N: BRR setting for baud rate generator ($0 \le N \le 255$)
P1ϕ: Operating frequency (MHz)
n and S: Determined by the SMR setting listed in the following table.

Table 15.5 Clock Source Settings

SMR Setting		
CKS[1:0] Bits	**Clock Source**	**n**
0 0	P1ϕ clock	0
0 1	P1ϕ/4 clock	1
1 0	P1ϕ/16 clock	2
1 1	P1ϕ/64 clock	3

chanhvinh.com

Table 15.6 Base Clock Settings in Smart Card Interface Mode

SCMR Setting	SMR Setting	Base Clock Cycles for	
BCP2 Bit	BCP[1:0] Bits	One-bit Period	S
0	0 0	93 clock cycles	93
0	0 1	128 clock cycles	128
0	1 0	186 clock cycles	186
0	1 1	512 clock cycles	512
1	0 0	32 clock cycles	32
1	0 1	64 clock cycles	64
1	1 0	372 clock cycles	372
1	1 1	256 clock cycles	256

Table 15.7 lists sample N settings in BRR in normal asynchronous mode. Table 15.8 lists the maximum bit rate settable for each operating frequency. Examples of BRR (N) settings in clock synchronous mode are listed in Table 15.10. Examples of BRR (N) settings in smart card interface mode are listed in Table 15.12. In smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, see section 15.6.4, Receive Data Sampling Timing and Reception Margin. Table 15.9 and Table 15.11 list the maximum bit rates with external clock input.

When the asynchronous mode base clock select bit (ABCS) in the serial extended mode register (SEMR) is set to 1 in asynchronous mode, the bit rate is two times that of listed in Table 15.7.

Table 15.7 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode)

	Operating Frequency P1ϕ (MHz)								
	50			64			66.67		
Bit Rate (bps)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	221	–0.02						
150	3	162	–0.15	3	207	0.16	3	216	0.01
300	3	80	0.47	3	103	0.16	3	108	–0.45
600	2	162	–0.15	2	207	0.16	2	216	0.01
1200	2	80	0.47	2	103	0.16	2	108	–0.45
2400	1	162	–0.15	1	207	0.16	1	216	0.01
4800	1	80	0.47	1	103	0.16	1	108	–0.45
9600	0	162	–0.15	1	51	0.16	0	216	0.01
19200	0	80	0.47	0	103	0.16	0	108	–0.45
31250	0	49	0.00	0	63	0.00	0	66	–0.50
38400	0	40	–0.76	0	51	0.16	0	53	0.47

[Legend]

Space: Setting prohibited.

-: Can be set, but there will be error.

Note: • This is an example when the ABCS bit in SEMR is 0.
When the ABCS bit is set to 1, the bit rate is two times.

chanhvinh.com

Table 15.8 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode)

P1φ (MHz)	Maximum Bit Rate(bps)	n	N
50	1562500	0	0
64	2000000	0	0
66.67	2083333	0	0

Note: • When the ABCS bit in SEMR is set to 1, the bit rate is two times.

Table 15.9 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

P1φ (MHz)	External Input Clock(MHz)	Maximum Bit Rate(bps)	
		SEMR.ABCS bit = 0	SEMR.ABCS bit = 1
50	12.5	781250	1562500
64	16	1000000	2000000
66.67	16.667	1041667	2083333

Table 15.10 BRR Settings for Various Bit Rates (Clock Synchronous Mode)

Bit Rate (bps)	Operating Frequency P1φ (MHz)					
	50		64		66.67	
	n	N	n	N	n	N
110						
250						
500						
1k	3	194	3	249		
2.5k	3	77	3	99	3	103
5k	2	155	2	199	2	207
10k	2	77	2	99	2	103
25k	1	124	1	159	1	166
50k	1	62	1	79	1	82
100k	0	124	0	159	0	166
250k	0	49	0	63	0	66
500k	0	24	0	31	0	32
1M	—	—	0	15	0	16
2.5M	0	4	—	—	—	—
5M	—	—	—	—	—	—

Space: Setting prohibited.
—: Can be set, but there will be error.

Table 15.11 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode)

P1φ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bps)
50	8.3333	8333333.3
64	10.6667	10666666.7
66.67	11.1111	11111100

chanhvinh.com

Table 15.12 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372)

Bit Rate (bps)	P1ϕ (MHz)								
	50			64			66.67		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	6	0.01	0	8	0.44	0	8	3.72

Table 15.13 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 372)

P1ϕ (MHz)	Maximum Bit Rate (bps)	n	N
50	67204	0	0
64	86022	0	0
66.67	89610	0	0

chanhvinh.com

15.2.10 Serial Extended Mode Register (SEMR)

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	NFEN	ABCS	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	ABCS	Asynchronous Mode Base Clock Select	(Valid only in asynchronous mode) 0: Selects 16 base clock cycles for 1-bit period 1: Selects 8 base clock cycles for 1-bit period	R/W*1
b5	NFEN	Digital Noise Filter Function Enable	(In asynchronous mode) 0: Noise cancellation function for the RXDn input signal is disabled. 1: Noise cancellation function for the RXDn input signal is enabled. The NFEN bit should be 0 in any mode other than above.	R/W*1
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).

NFEN Bit (Digital Noise Filter Function Enable)

This bit enables or disables the digital noise filter function.

When the function is enabled, noise cancellation is applied to the RXDn input signal in asynchronous mode.

In any mode other than above, set the NFEN bit to 0 to disable the digital noise filter function.

When the function is disabled, input signals are transferred as is, as internal signals.

15.2.11 Noise Filter Setting Register (SNFR)

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	NFCS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	NFCS[2:0]	Noise Filter Clock Select	In asynchronous mode, the standard setting for the base clock is as follows. b2 b0 0 0 0: The clock signal divided by 1 is used with the noise filter. Other values: Do not make settings other than those listed above.	R/W*1
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR are 0. (both serial transmission and reception are disabled.)

NFCS[2:0] Bits (Noise Filter Clock Select)

These bits select the sampling clock for the digital noise filter. To use the noise filter in asynchronous mode, set these bits to 000b.

chanhvinh.com

15.2.12 Extended Function Control Register (SECR)

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CTSE	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R
b1	CTSE	CTS Enable	0: CTS pin function is disabled (RTS output function is enabled). 1: CTS pin function is enabled	R/W*1
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writing to this bit is only possible when the RE and TE bits in the SCR are 0 (both serial transmission and reception are disabled).

SECR is used to select the extension settings in asynchronous and clock-synchronous modes.

CTSE Bit (CTS Enable)

Set this bit to 1 if the CTS control signal is used for control of transmission and reception. The RTS signal is output when this bit is set to 0. Set this bit to 0 in smart card interface mode.

chanhvinh.com

15.3 Operation in Asynchronous Mode

Figure 15.2 shows the general format for asynchronous serial communications.
One frame consists of a start bit (low level), transmit/receive data, a parity bit, and stop bits (high level).
In asynchronous serial communications, the communications line is usually held in the mark state (high level).
The SCI monitors the communications line, and when it goes to the space state (low level), recognizes a start bit and starts serial communications.
Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

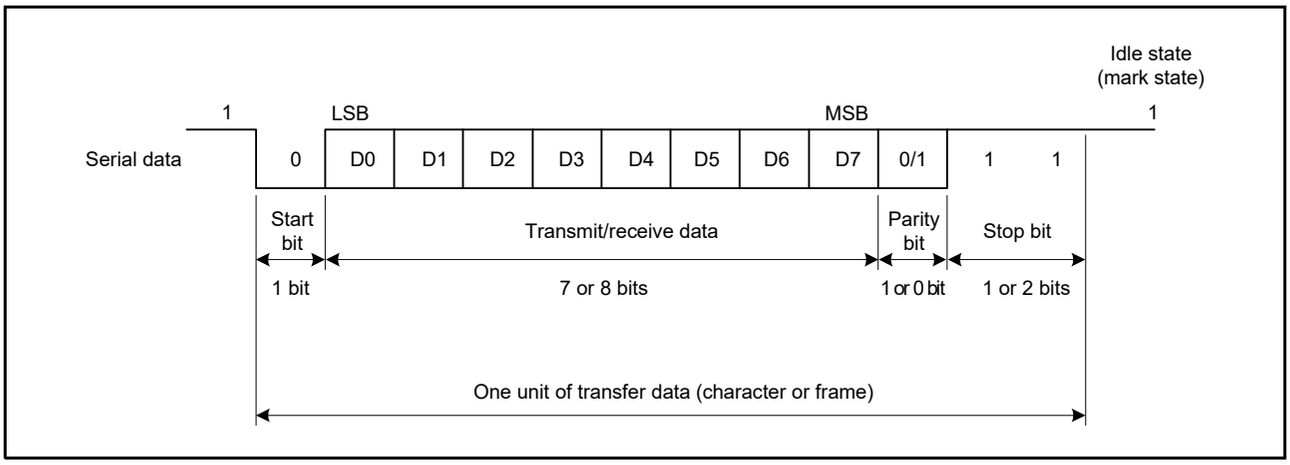


Figure 15.2 Data Format in Asynchronous Serial Communications (Example with 8-Bit Data, Parity, Two Stop Bits)

15.3.1 Serial Data Transfer Format

Table 15.14 lists the serial data transfer formats that can be used in asynchronous mode.
Any of 12 transfer formats can be selected according to the SMR setting. For details of multi-processor function, see section 15.4, Multi-Processor Communications Function.

chanhvinh.com

Table 15.14 Serial Transfer Formats (Asynchronous Mode)

SMR Setting				Serial Transfer Format and Frame Length
CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12
0	0	0	0	S | 8-bit data | STOP
0	0	0	1	S | 8-bit data | STOP | STOP
0	1	0	0	S | 8-bit data | P | STOP
0	1	0	1	S | 8-bit data | P | STOP | STOP
1	0	0	0	S | 7-bit data | STOP
1	0	0	1	S | 7-bit data | STOP | STOP
1	1	0	0	S | 7-bit data | P | STOP
1	1	0	1	S | 7-bit data | P | STOP | STOP
0	—	1	0	S | 8-bit data | MPB | STOP
0	—	1	1	S | 8-bit data | MPB | STOP | STOP
1	—	1	0	S | 7-bit data | MPB | STOP
1	—	1	1	S | 7-bit data | MPB | STOP | STOP

S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multi-processor bit

chanhvinh.com

15.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the SCI operates on a base clock with a frequency of 16 times*1 the bit rate.
In reception, the SCI samples the falling edge of the start bit using the base clock, and performs internal synchronization. Since receive data is sampled at the rising edge of the 8th pulse*1 of the base clock, data is latched at the middle of each bit, as shown in Figure 15.3. Thus the reception margin in asynchronous mode is determined by formula (1) below.

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\,F - \frac{D - 0.5}{N}\,(1+F) \right| \times 100\ [\%] \quad \cdots \text{ Formula (1)}$$

M: Reception margin
N: Ratio of bit rate to clock (N = 16 when ABCS in SEMR = 0, N = 8 when ABCS in SEMR = 1)
D: Duty cycle of clock (D = 0.5 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute value of clock frequency deviation

Assuming values of F = 0 and D = 0.5 in formula (1), the reception margin is determined by the formula below.

$$M = \{0.5 - 1/(2 \times 16)\} \times 100\ (\%) = 46.875\%$$

However, this is only the computed value, and a margin of 20% to 30% should be allowed in system design.

Note 1. This is an example when the ABCS bit in SEMR is 0. When the ABCS bit is 1, a frequency of 8 times the bit rate is used as a base clock and receive data is sampled at the rising edge of the 4th pulse of the base clock

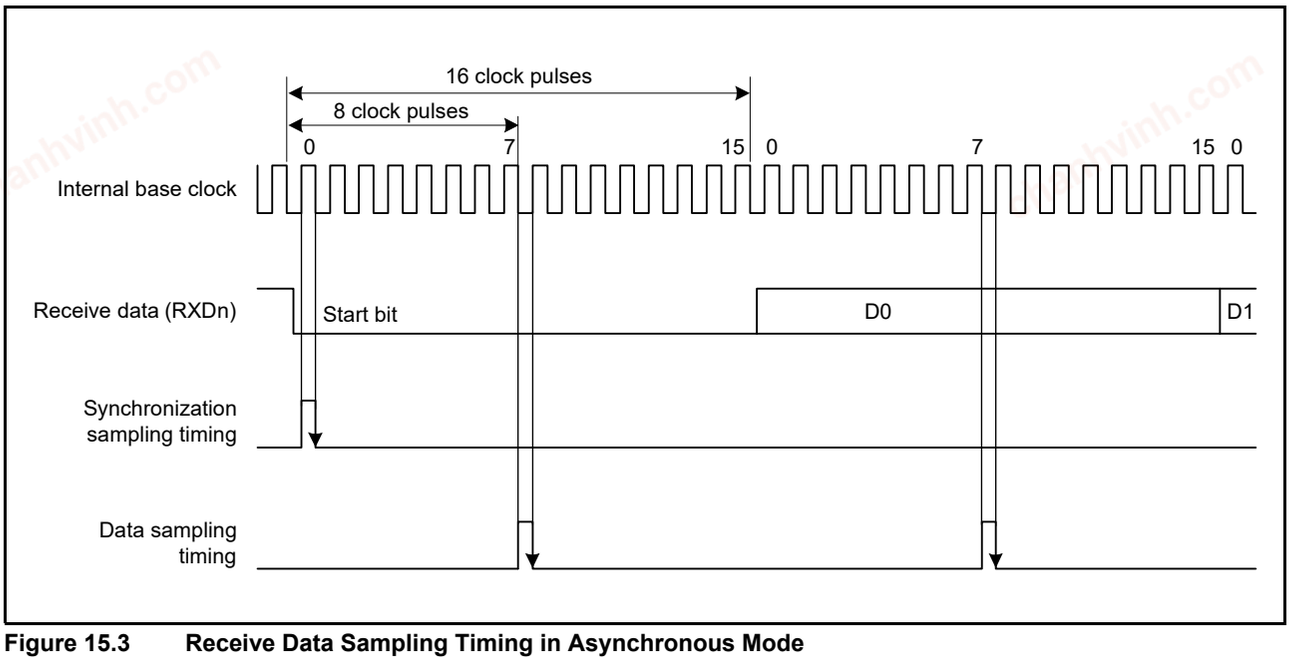


Figure 15.3 Receive Data Sampling Timing in Asynchronous Mode

chanhvinh.com

15.3.3 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input to the SCKn pin can be selected as the SCI's transfer clock, according to the setting of the CM bit in SMR and the CKE[1:0] bits in SCR.
When an external clock is input to the SCKn pin, the clock frequency should be 16 times the bit rate (when ABCS in SEMR = 0) and 8 times the bit rate (when ABCS in SEMR = 1).
When the SCI is operated on an internal clock, the clock can be output from the SCKn pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in Figure 15.4.

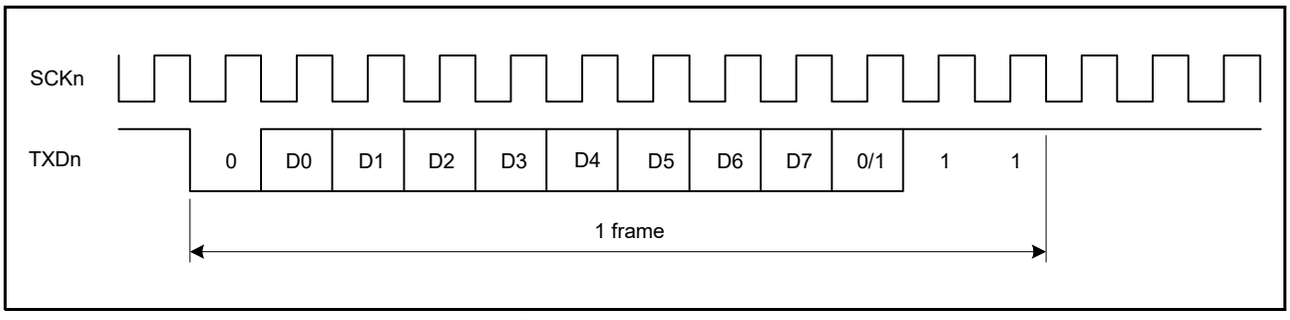


Figure 15.4 Phase Relationship between Output Clock and Transmit Data (Asynchronous Mode: SMR.CHR = 0, PE = 1, MP = 0, STOP = 1)

15.3.4 CTS and RTS Functions

The CTS function is the use of input on the CTSn# pin in transmission control. Setting the SECR.CTSE bit to 1 enables the CTS function. When the CTS function is enabled, placing the low level on the CTSn# pin causes transmission to start.
Applying the low level to the CTS# pin while transmission is in progress does not affect transmission of the current frame, which continues.
In the RTS function, by using the function of output on the RTSn# pin, a low level is output when reception becomes possible. Conditions for output of the low and high level are shown below.

[Conditions for low-level output]
Satisfaction of all conditions listed below

- The value of the RE bit in the SCR is 1
- Reception is not in progress
- There are no received data yet to be read
- The ORER, FER, and PER flags in the SSR are all 0

[Condition for high-level output]

- Any of the conditions for the low level not being satisfied

chanhvinh.com

15.3.5 SCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, start by writing the initial value “00h” to SCR and then continue through the procedure for SCI given in the sample flowchart (Figure 15.5). Whenever the operating mode or transfer format is changed, SCR must be initialized before the change is made.

When the external clock is used in asynchronous mode, ensure that the clock signal is supplied even during initialization.

Note that clearing the SCR.RE bit to 0 initializes neither the ORER, FER, and PER flags in SSR nor RDR.

Moreover, note that switching the value of the SCR.TE bit from 1 to 0 or 0 to 1 while the SCR.TIE bit is 1 leads to the generation of a TXI interrupt request.

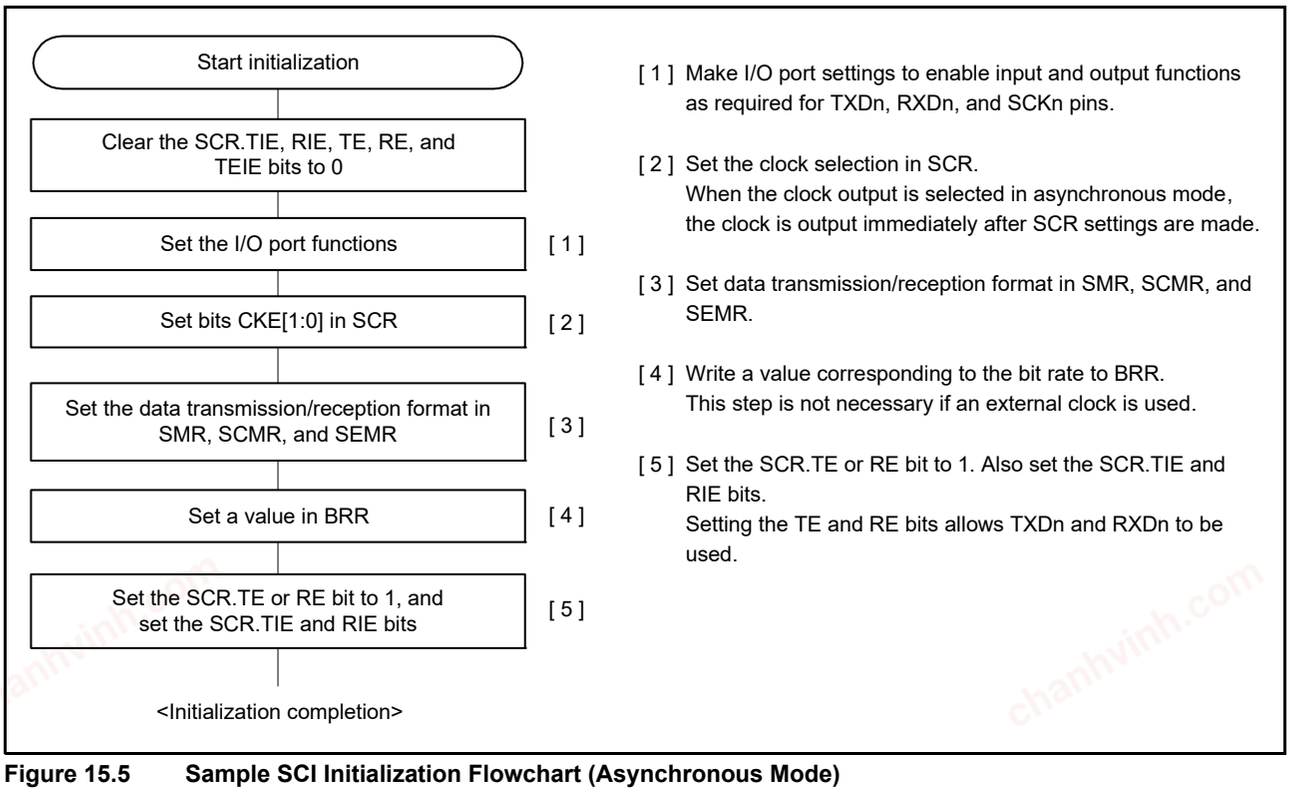


Figure 15.5 Sample SCI Initialization Flowchart (Asynchronous Mode)

chanhvinh.com

15.3.6 Serial Data Transmission (Asynchronous Mode)

Figure 15.6 shows an example of the operation for serial transmission in asynchronous mode.
In serial transmission, the SCI operates as described below.

1. The SCI transfers data from TDR to TSR when data is written to TDR in the TXI interrupt processing routine. The TXI interrupt request at the beginning of transmission is generated when the TE bit in SCR is set to 1 after the TIE bit in SCR is set to 1 or when these two bits are set to 1 simultaneously by a single instruction.
2. Transmission starts after the CTSE bit in SECR is set to 0 (disabling the CTS function) and a low level on the CTS# pin causes data transfer from TDR to TSR. If the TIE bit in SCR is 1 at this time, a TXI interrupt request is generated. Continuous transmission is obtainable by writing the next data for transmission to TDR in the TXI interrupt processing routine before transmission of the current data for transmission is completed. When TEI interrupt requests are in use, set the SCR.TIE bit to 0 (disabling TXI requests) and the SCR.TEIE bit to 1 (enabling TEI requests) after the last of the data to be transmitted are written to the TDR from the processing routine for TXI requests.
3. Data is sent from the TXDn pin in the following order: start bit, transmit data, parity bit or multi-processor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks for updating of (writing to) TDR at the time of stop bit output.
5. When TDR is updated, setting of the CTSE bit in SECR to 0 (CTS function disabled) or a low level input on the CTSn# pin cause the next transfer of the next data for transmission from TDR to TSR and sending of the stop bit, after which serial transmission of the next frame starts.
6. If TDR is not updated, the TEND flag in SSR is set to 1, the stop bit is sent, and then the mark state is entered in which 1 is output. If the TEIE flag in SCR is 1 at this time, the TEND flag in SSR is set to 1 and a TEI interrupt request is generated.

Figure 15.7 shows a sample flowchart for serial transmission in asynchronous mode.

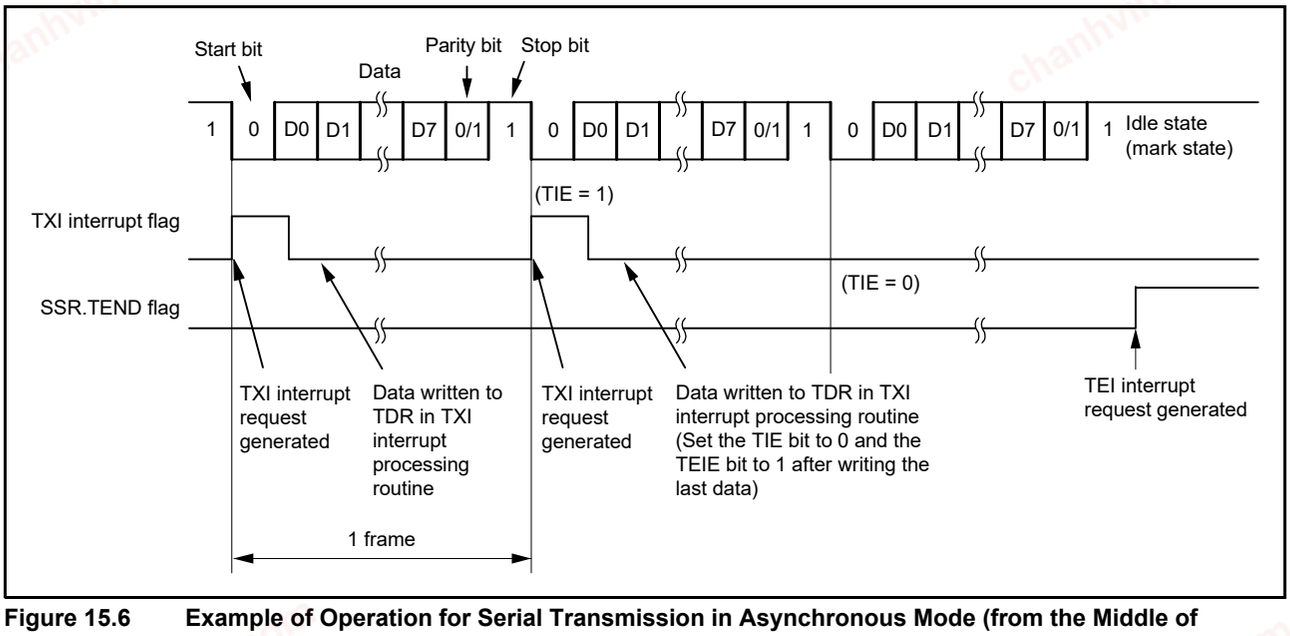
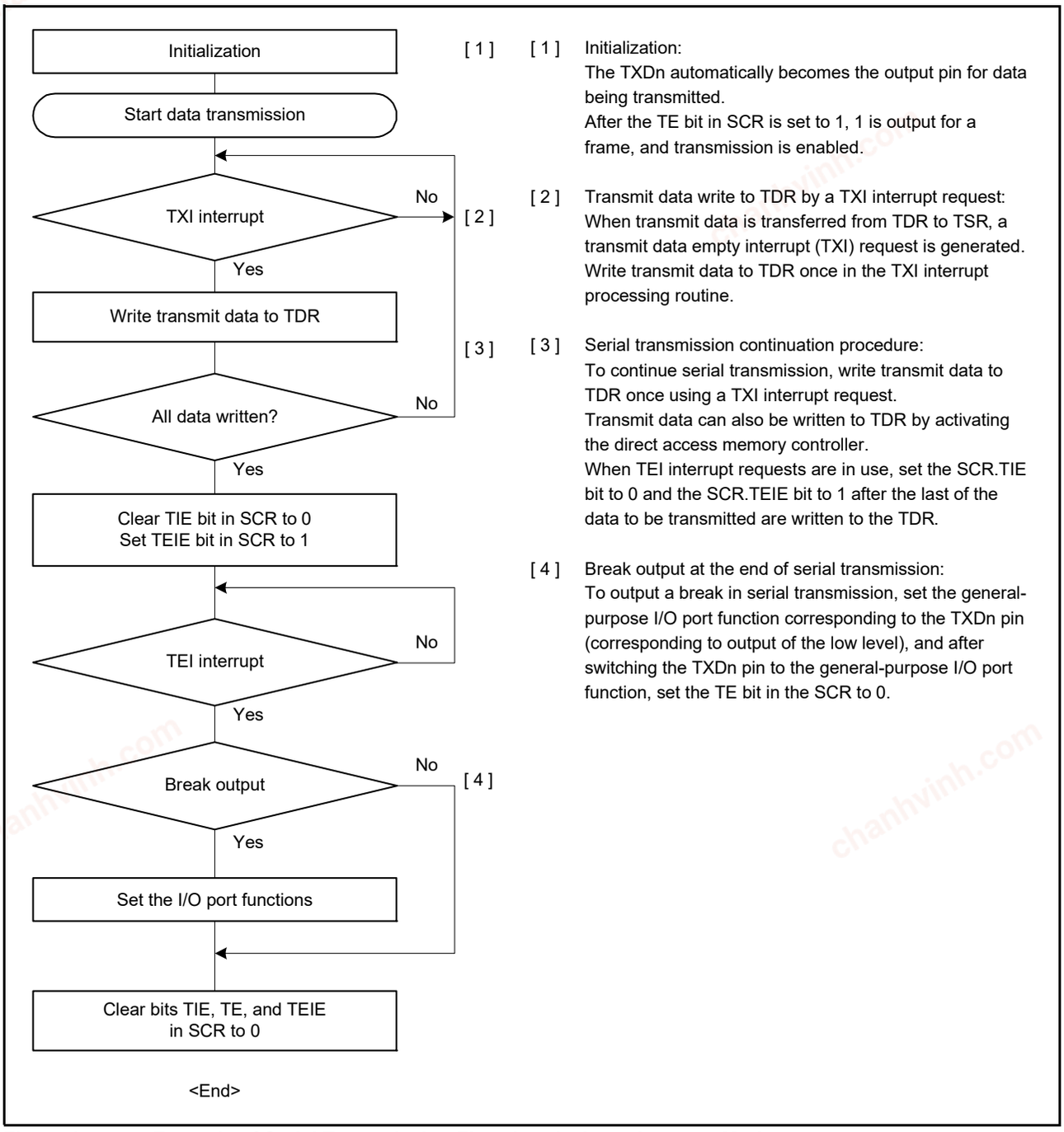


Figure 15.6 Example of Operation for Serial Transmission in Asynchronous Mode (from the Middle of Transmission until Transmission Completion) (Example with 8-Bit Data, Parity, One Stop Bit)

chanhvinh.com



[1] Initialization:
The TXDn automatically becomes the output pin for data being transmitted.
After the TE bit in SCR is set to 1, 1 is output for a frame, and transmission is enabled.

[2] Transmit data write to TDR by a TXI interrupt request:
When transmit data is transferred from TDR to TSR, a transmit data empty interrupt (TXI) request is generated.
Write transmit data to TDR once in the TXI interrupt processing routine.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to TDR once using a TXI interrupt request.
Transmit data can also be written to TDR by activating the direct access memory controller.
When TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted are written to the TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set the general-purpose I/O port function corresponding to the TXDn pin (corresponding to output of the low level), and after switching the TXDn pin to the general-purpose I/O port function, set the TE bit in the SCR to 0.

Figure 15.7 Example of Serial Transmission Flowchart in Asynchronous Mode

chanhvinh.com

15.3.7 Serial Data Reception (Asynchronous Mode)

Figure 15.8 and Figure 15.9 show examples of the operation for serial data reception in asynchronous mode.
In serial data reception, the SCI operates as described below.

1. When the value of the RE bit in SCR becomes 1, the output signal on the RTSn# pin goes to the low level.
2. When the SCI monitors the communications line and detects a start bit, it performs internal synchronization, stores receive data in RSR, and checks the parity bit and stop bit.
3. If an overrun error occurs, the ORER flag in SSR is set to 1. If the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR.
4. If a parity error is detected, the PER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated.
5. If a framing error (when the stop bit is 0) is detected, the FER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated.
6. When reception finishes successfully, receive data is transferred to RDR. If the RIE bit in SCR is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to RDR in this RXI interrupt processing routine before reception of the next receive data is completed. Reading out the received data that have been transferred to RDR causes the RTSn# pin to output the low level.

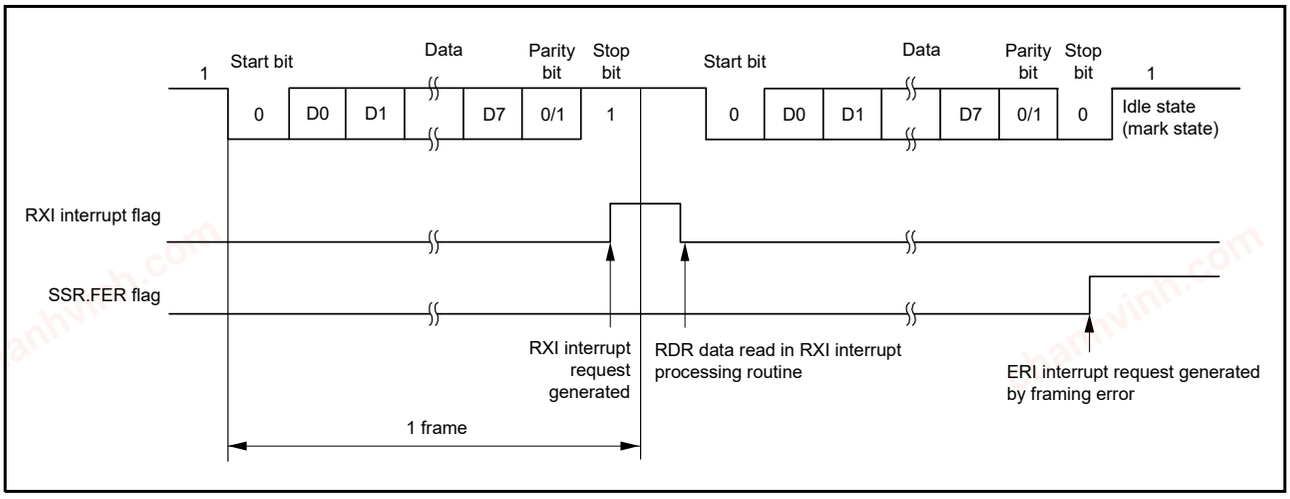


Figure 15.8 Example of SCI Operation for Serial Reception in Asynchronous Mode (1) (when RTS Function is not Used) (Example with 8-Bit Data, Parity, One Stop Bit)

chanhvinh.com

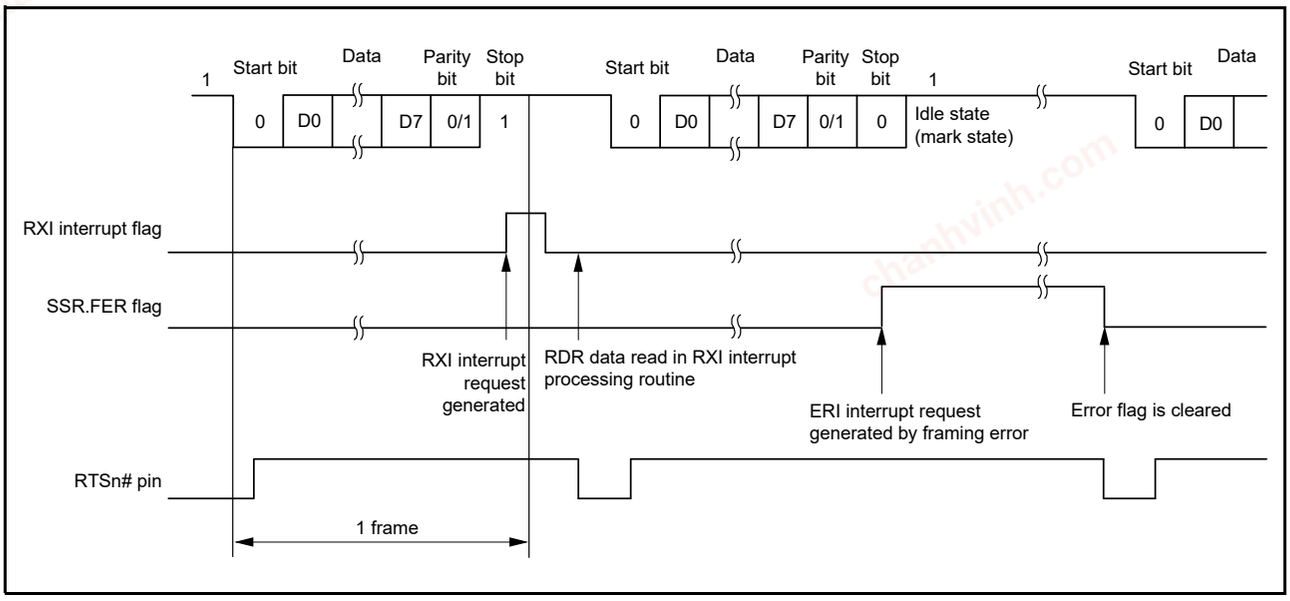


Figure 15.9 Example of SCI Operation for Serial Reception in Asynchronous Mode (2) (when RTS Function Is Used) (Example with 8-Bit Data, Parity, One Stop Bit)

Table 15.15 lists the states of the SSR status flags and receive data handling when a receive error is detected.
If a receive error is detected, an ERI interrupt request is generated but an RXI interrupt request is not generated. Data reception cannot be resumed while the receive error flag is 1. Accordingly, clear the ORER, FER, and PER bits to 0 before resuming reception. Moreover, be sure to read the RDR during overrun error processing.
Figure 15.10 and Figure 15.11 show samples of flowcharts for serial data reception.

Table 15.15 SSR Status Flags and Receive Data Handling

SSR Status Flag				
ORER	**FER**	**PER**	**Receive Data**	**Receive Error Type**
1	0	0	Lost	Overrun error
0	1	0	Transferred to RDR	Framing error
0	0	1	Transferred to RDR	Parity error
1	1	0	Lost	Overrun error + framing error
1	0	1	Lost	Overrun error + parity error
0	1	1	Transferred to RDR	Framing error + parity error
1	1	1	Lost	Overrun error + framing error + parity error

chanhvinh.com

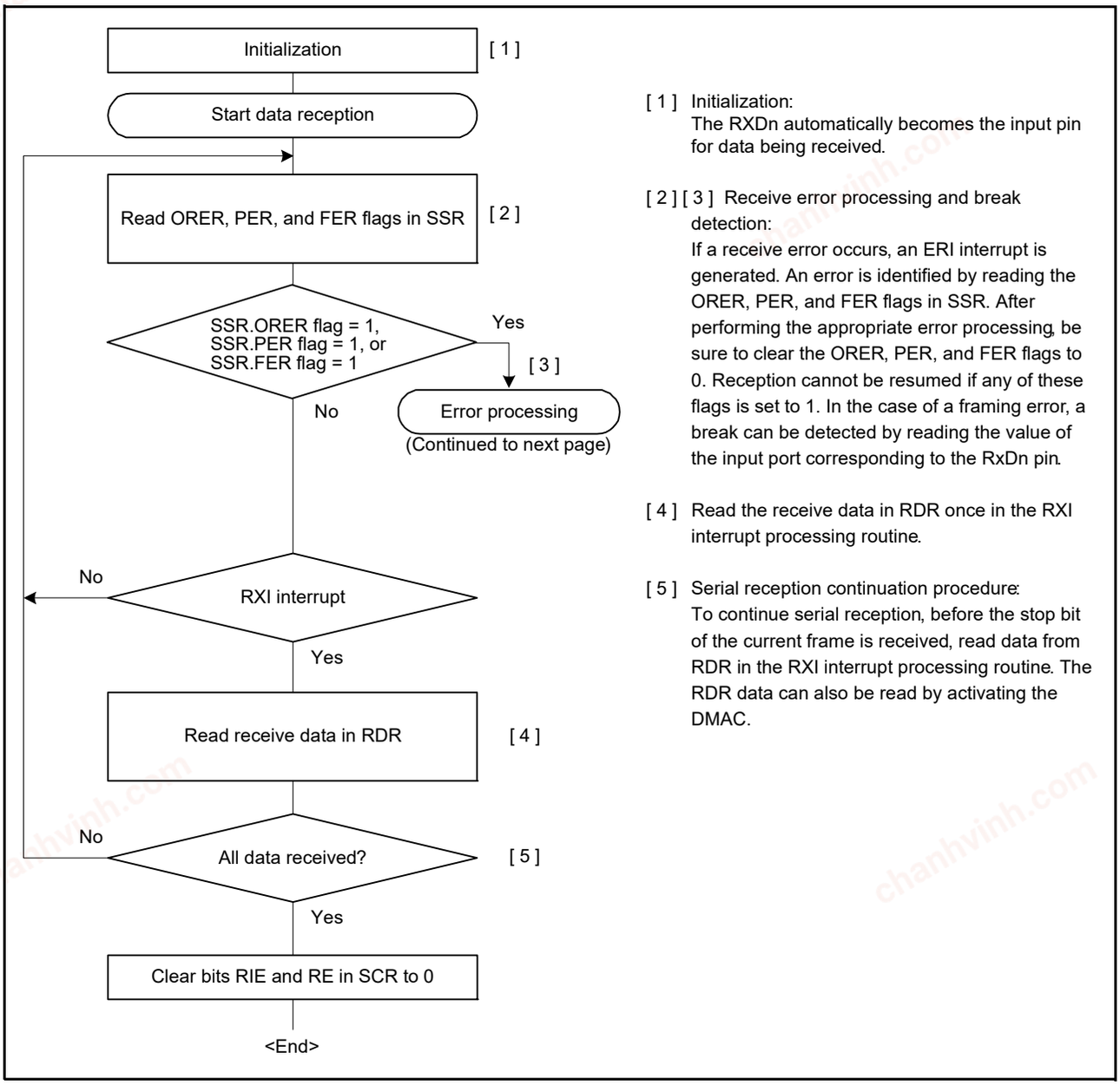


Figure 15.10 Example of Serial Reception Flowchart (1) (Asynchronous Mode)

chanhvinh.com

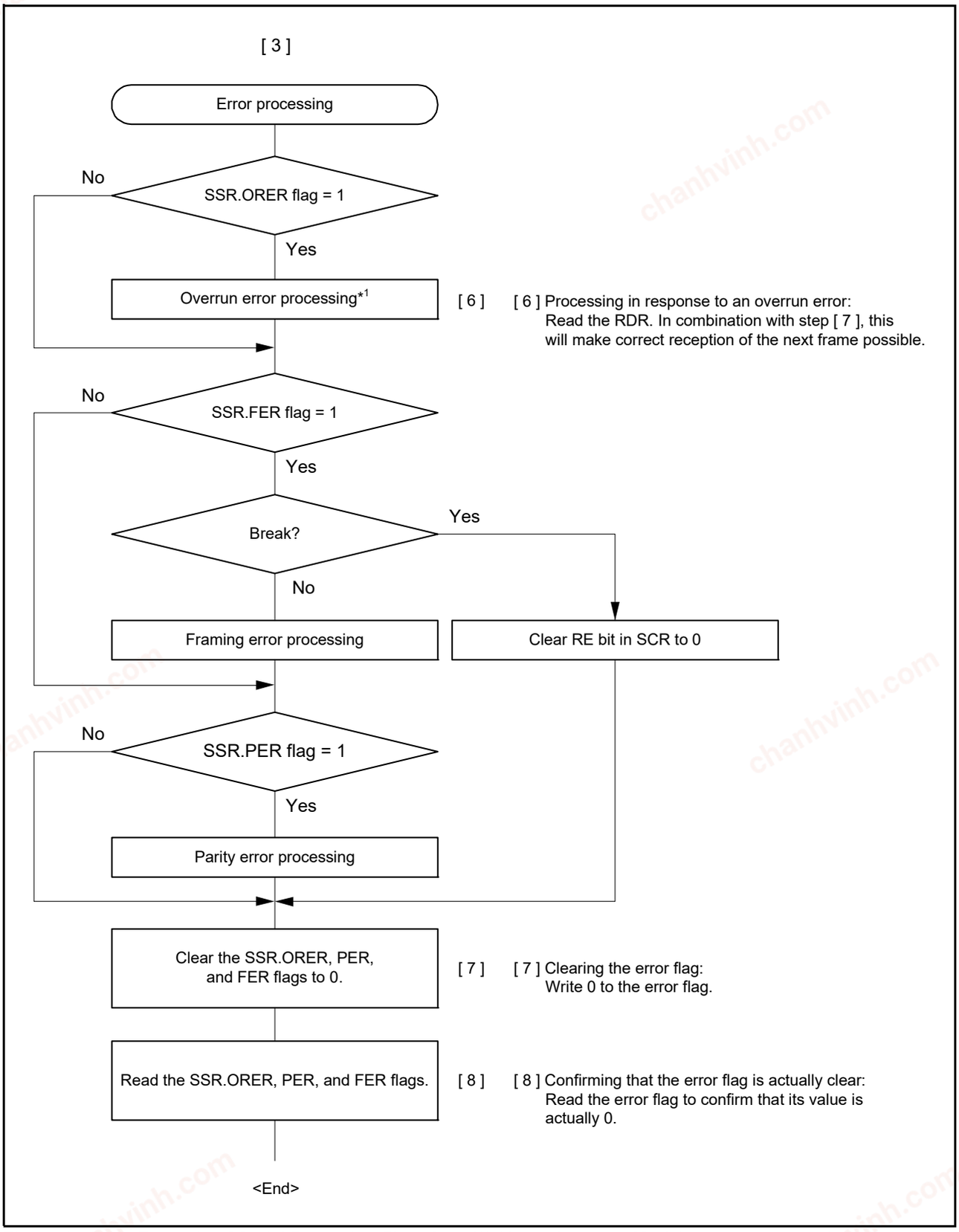


Figure 15.11 Example of Serial Reception Flowchart (2) (Asynchronous Mode)

chanhvinh.com

15.4 Multi-Processor Communications Function

Using the multi-processor communication functions enables to transmit and receive data by sharing a communication line between multiple processors by using asynchronous serial communication in which the multi-processor bit is added. In multi-processor communication, a unique ID code is allocated to each receiving station. Serial communication cycles consist of an ID transmission cycle to specify the receiving station and a data transmission cycle to transmit data to the specified receiving station. The multi-processor bit is used to distinguish between the ID transmission cycle and the data transmission cycle. When the multi-processor bit is set to 1, it indicates the ID transmission cycle and when the multi-processor bit is set to 0, it indicates the data transmission cycle. Figure 15.12 shows an example of communication between processors by using a multi-processor format. First, a transmitting station transmits communication data in which the multi-processor bit set to 1 is added to the ID code of the receiving station. Next, the transmitting station transmits the communication data in which the multi-processor bit set to 0 is added to the transmission data. Upon receiving the communication data in which the multi-processor bit is set to 1, the receiving station compares the received ID with the ID of the receiving station itself and if the two match, receives the communication data that is subsequently transmitted. If the received ID does not match with the ID of the receiving station, the receiving station skips the communication data until again receiving the communication data in which the multi-processor bit is set to 1.

For supporting this function, the SCI provides the MPIE bit in SCR. When the MPIE bit in SCR is set to 1, transfer of receive data from the RSR to the RDR, detection of a reception error, and setting the respective status flags ORER and FER in SSR are disabled until reception of data in which the multi-processor bit is set to 1. Upon receiving a reception character in which the multi-processor bit is set to 1, the MPBT bit in SSR is set to 1 and the MPIE bit in SCR is automatically cleared, thus returning to a normal reception operation. During this time, an RXI interrupt is generated if the RIE bit in SCR is set.

When the multi-processor format is specified, specification of the parity bit is disabled. Apart from this, there is no difference from the operation in the normal asynchronous mode. A clock which is used for the multi-processor communication is also the same as the clock used in the normal asynchronous mode.

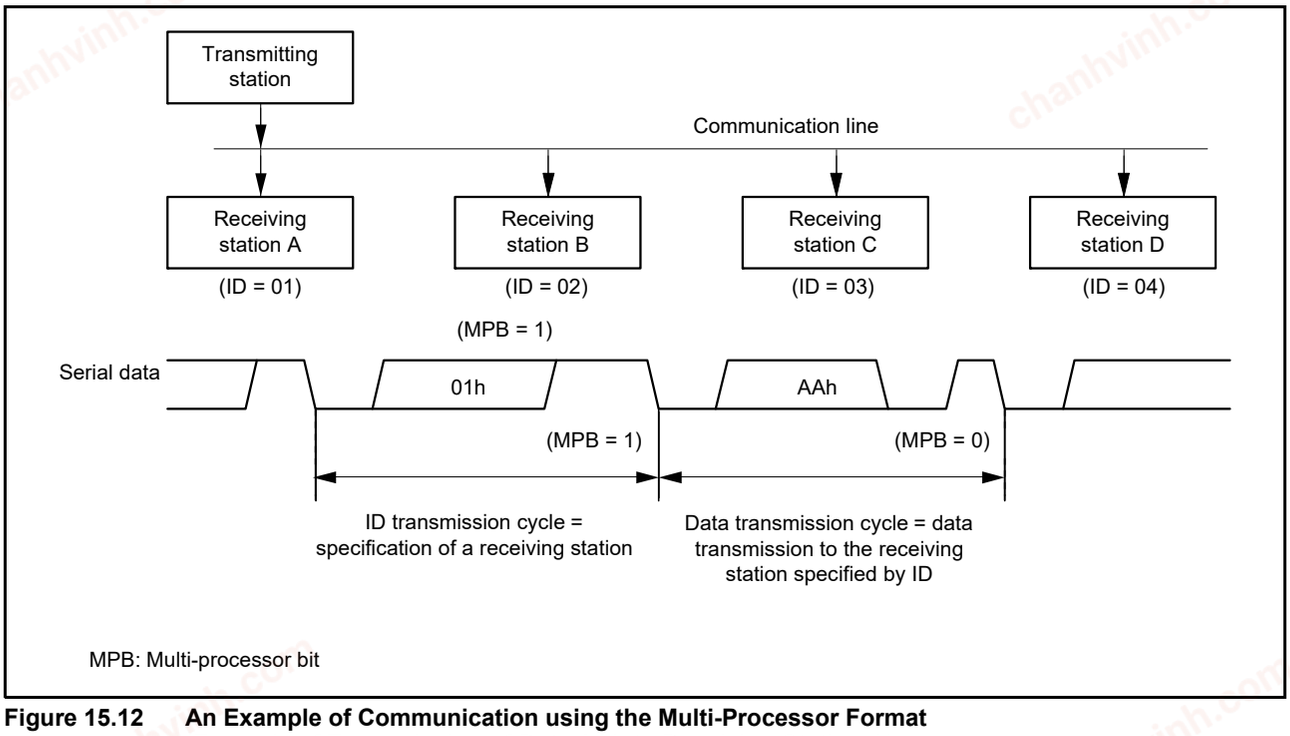


Figure 15.12 An Example of Communication using the Multi-Processor Format (Example of Transmission of Data AAh to Receiving Station A)

chanhvinh.com

15.4.1 Multi-Processor Serial Data Transmission

Figure 15.13 is a sample flowchart of multi-processor data transmission. In the ID transmission cycle, the ID should be transmitted with the MPBT bit in SSR set to 1. In the data transmission cycle, the data should be transmitted with the MPBT bit set to 0. The other operations are the same as the operations in asynchronous mode.

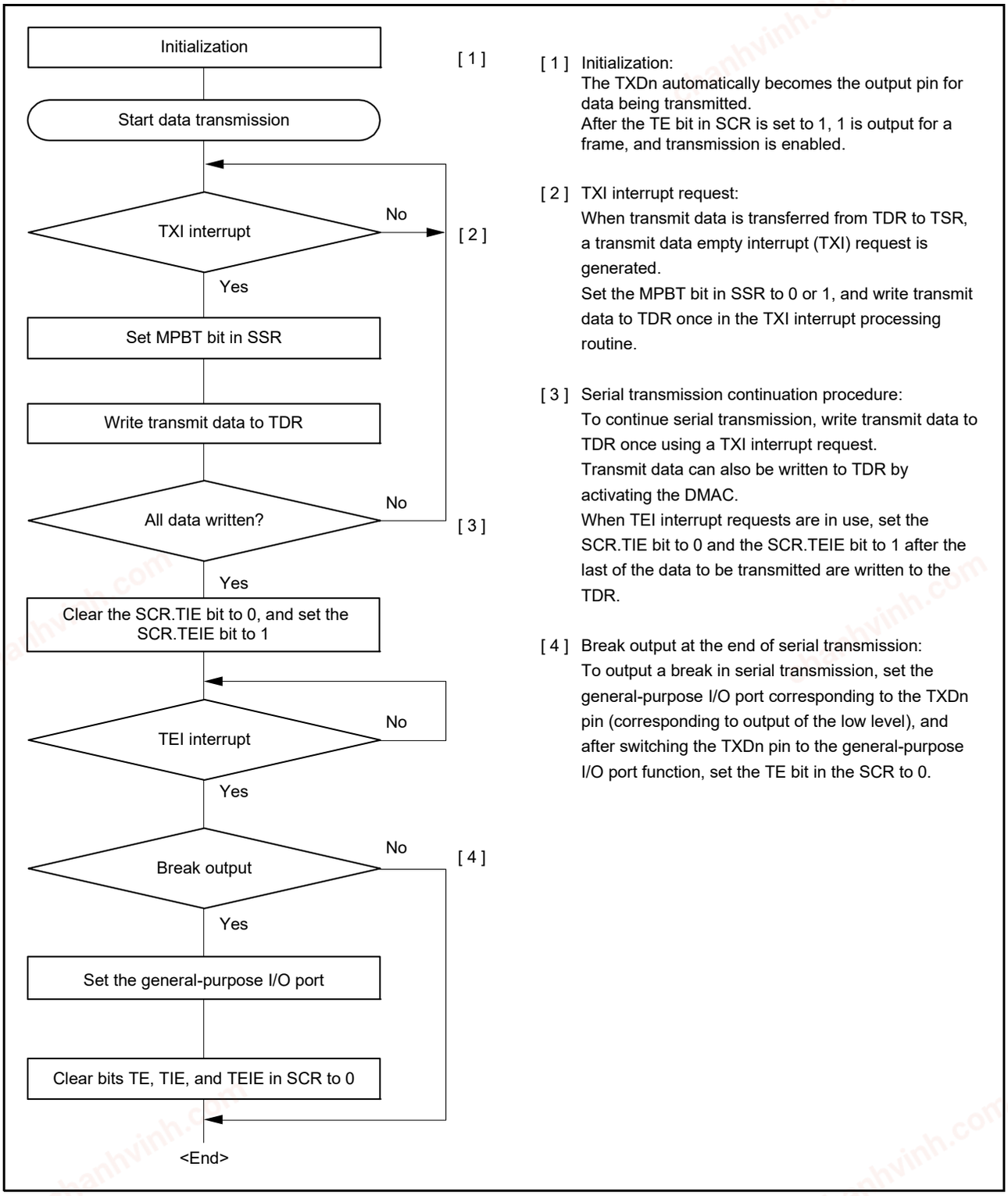


Figure 15.13 Example of Multi-Processor Serial Transmission Flowchart

chanhvinh.com

15.4.2 Multi-Processor Serial Data Reception

Figure 15.15 and Figure 15.16 are sample flowcharts of multi-processor data reception. When the MPIE bit in SCR is set to 1, reading the communication data is skipped until reception of the communication data in which the multi-processor bit is set to 1. When the communication data in which the multi-processor bit is set to 1 is received, the received data is transferred to RDR. During this time, the RXI interrupt request is generated. The other operations are the same as the operations in asynchronous mode.
Figure 15.14 is the example of operation for reception.

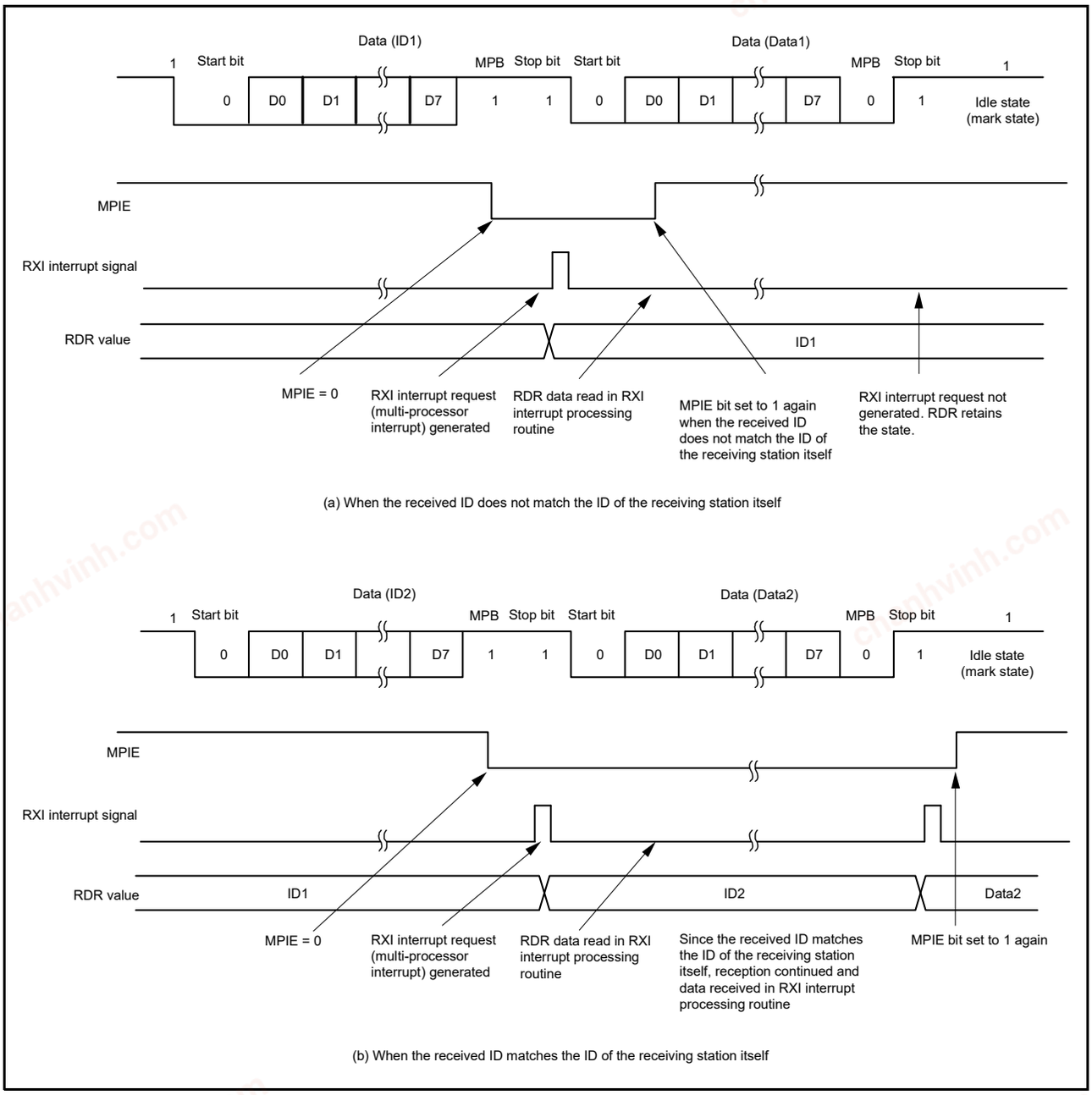


Figure 15.14 Example of Reception (8-Bit Data/Multi-Processor Bit/One Stop Bit)

chanhvinh.com

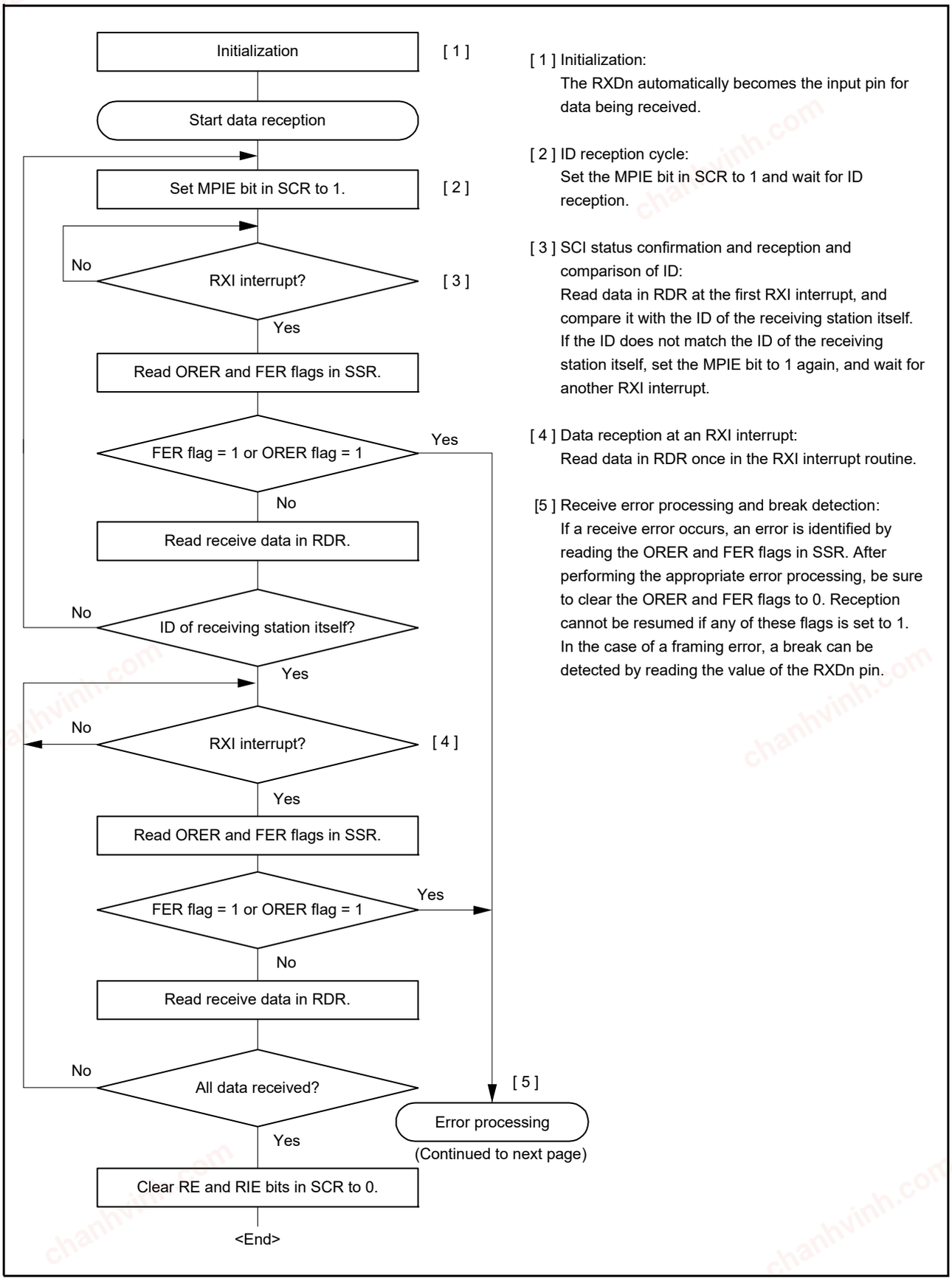


Figure 15.15 Example of Multi-Processor Serial Reception Flowchart (1)

chanhvinh.com

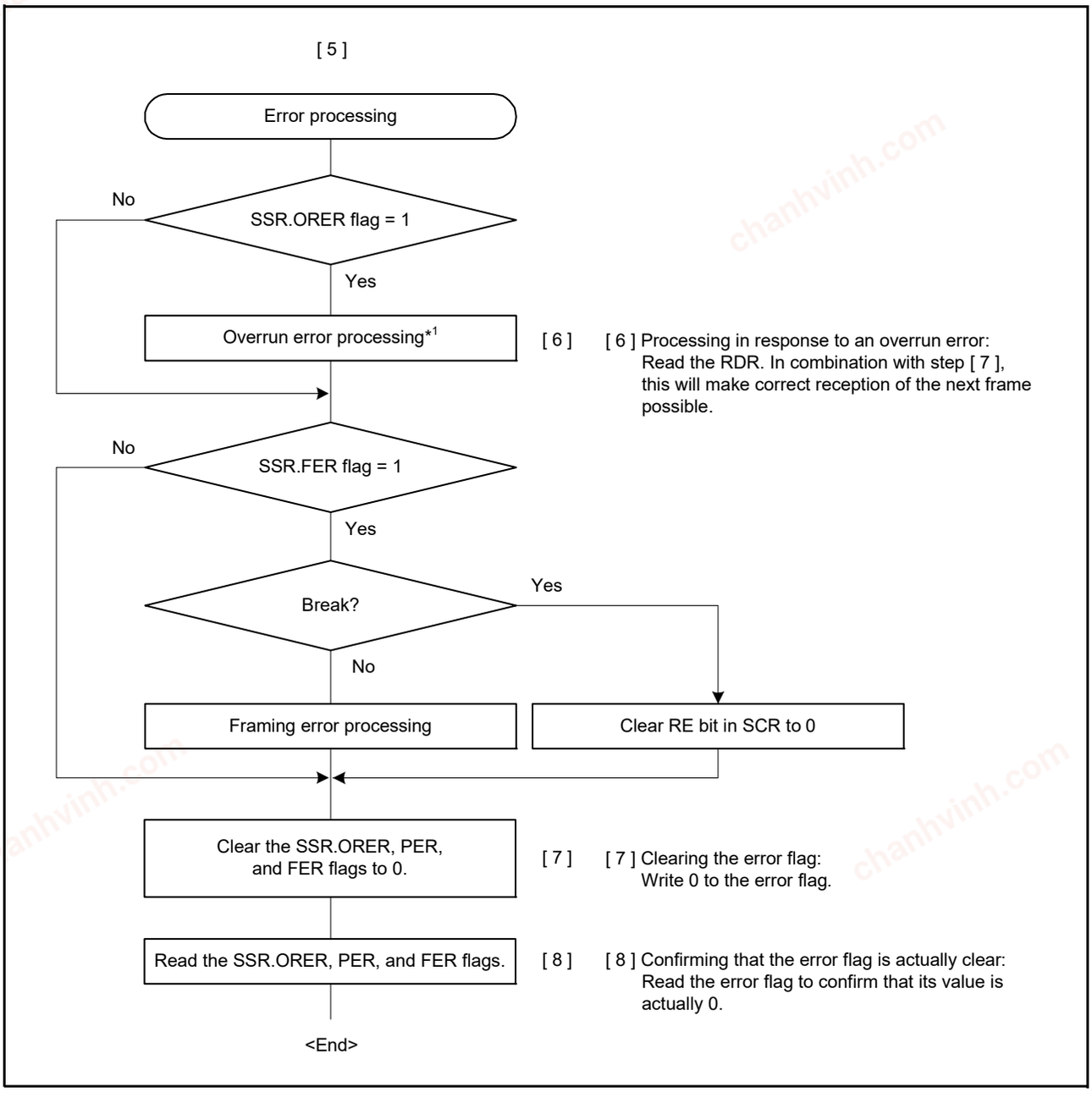


Figure 15.16 Example of Multi-Processor Serial Reception Flowchart (2)

chanhvinh.com

15.5 Operation in Clock Synchronous Mode

Figure 15.17 shows the data format for clock synchronous serial data communications.
In clock synchronous mode, data is transmitted or received in synchronization with clock pulses. One character in transfer data consists of 8-bit data. In clock synchronous mode, no parity bit can be added.
In data transmission, the SCI outputs data from one falling edge of the synchronization clock to the next. In data reception, the SCI receives data in synchronization with the rising edge of the synchronization clock. After 8-bit data is output, the transmission line holds the last bit output state.
Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications by use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so that the next transmit data can be written during transmission or the previous receive data can be read during reception, enabling continuous data transfer.

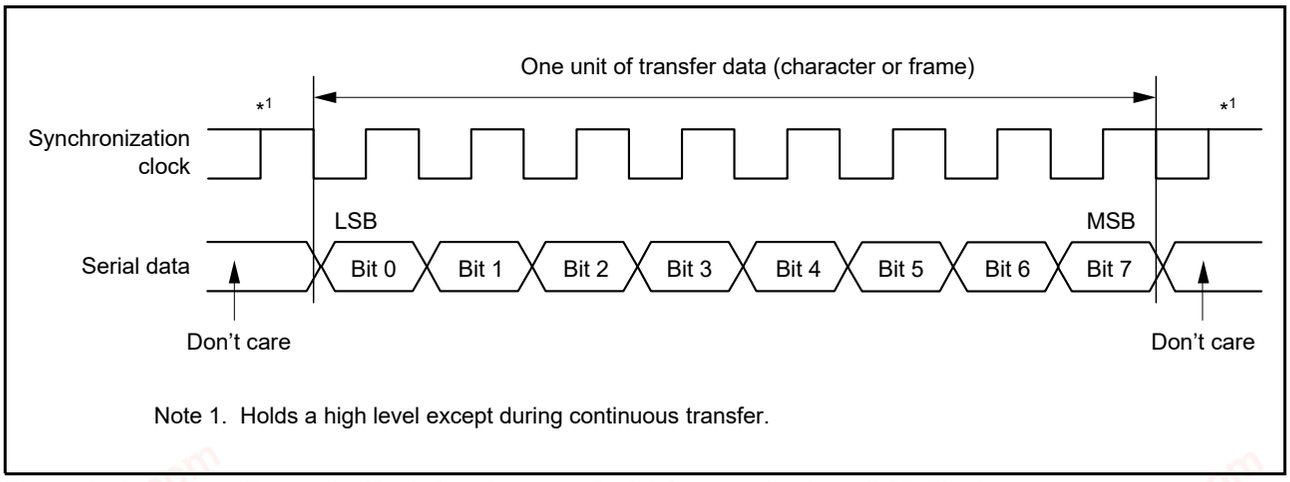


Figure 15.17 Data Format in Clock Synchronous Serial Communications (LSB-First)

15.5.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCKn pin can be selected, according to the setting of the CKE[1:0] bits in SCR.
When the SCI is operated on an internal clock, the synchronization clock is output from the SCKn pin. Eight synchronization clock pulses are output in the transfer of one character, and when no transfer is performed the clock is held high. However, when only data reception is performed, output of the synchronizing clock signal continues until the CTS function is enabled and the high level is input on the CTSn# pin, an overflow error occurs, or the RE bit in SCR is set to 0. When the CTS function is enabled, the synchronous clock signal output is stopped if the CTSn# pin input is high on completion of the frame reception.

15.5.2 CTS and RTS Functions

In the CTS function, CTSn# pin input is used to control reception/transmission start when the clock source is the internal clock. Setting the SECR.CTSE bit to 1 enables the CTS function.
When the CTS function is enabled, placing the low level on the CTSn# pin causes reception/transmission to start.
In the RTS function, RTSn# pin output is used to request reception/transmission start when the clock source is an external synchronizing clock. A low level is output when serial communications become possible. Conditions for output of the low and high level are shown below.

chanhvinh.com

[Conditions for low-level output]
Satisfaction of all conditions listed below

- The value of the RE or TE bit in the SCR is 1
- Neither transmission nor reception is in progress
- There are no received data yet to be read (when the SCR.RE bit is 1)
- Transmit data has been written (when the SCR.TE bit is 1)
- ORER flag in SSR is 0

[Condition for high-level output]
Any of the conditions for the low level not being satisfied

15.5.3 Initialization (Clock Synchronous Mode)

Before transmitting and receiving data, start by writing the initial value “00h” to the SCR and then continue through the procedure for SCI given in the sample flowchart (Figure 15.18). Whenever the operating mode or transfer format is changed, the SCR must be initialized before the change is made.
Note that clearing the SCR.RE bit to 0 initializes neither the ORER, FER, and PER flags in SSR nor RDR.
Moreover, note that switching the value of the SCR.TE bit from 1 to 0 or 0 to 1 while the SCR.TIE bit is 1 leads to the generation of a TXI interrupt request.

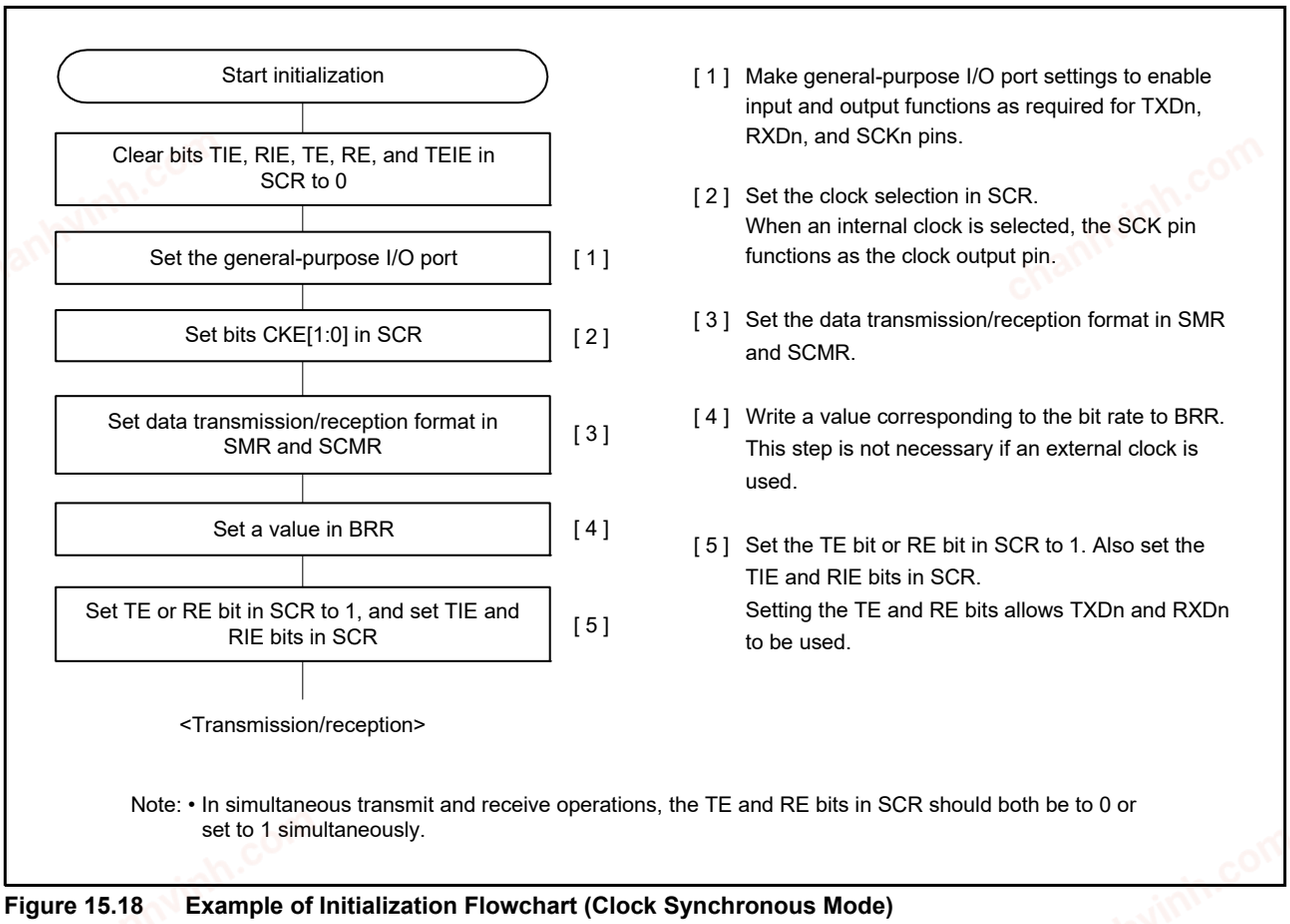


Figure 15.18 Example of Initialization Flowchart (Clock Synchronous Mode)

chanhvinh.com

15.5.4 Serial Data Transmission (Clock Synchronous Mode)

Figure 15.19 shows an example of the operation for serial transmission in clock synchronous mode.
In serial data transmission, the SCI operates as described below.

1. The SCI transfers data from TDR to TSR when data is written to TDR in the TXI interrupt processing routine. The TXI interrupt request at the beginning of transmission is generated when the TE bit in SCR is set to 1 after the TIE bit in SCR is set to 1 or when these two bits are set to 1 simultaneously by a single instruction.
2. After transferring data from TDR to TSR, the SCI starts transmission. When the SCR.TIE bit is set to 1 at this time, a TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to TDR in this TXI interrupt processing routine before transmission of the current transmit data has finished. When TEI interrupt requests are in use, set the SCR.TIE bit to 0 (disabling TXI requests) and the SCR.TEIE bit to 1 (enabling TEI requests) after the last of the data to be transmitted are written to the TDR from the processing routine for TXI requests.
3. 8-bit data is sent from the TXDn pin in synchronization with the output clock when clock output mode has been specified and in synchronization with the input clock when use of an external clock has been specified. Output of the clock signal is suspended until the input CTS signal is at the low level while the CTSE bit in SECR is 1 (enabling the CTS function).
4. The SCI checks for updating of (writing to) the TDR at the time of the last bit output.
5. When TDR is updated, the next transmit data is transferred from TDR to TSR, and serial transmission of the next frame is started.
6. If TDR is not updated, set the SSR flag in TEND to 1 and the TXDn pin retains the output state of the last bit. If the TEIE bit in SCR is 1 at this time, a TEI interrupt request is generated. The SCKn pin is held high.

Figure 15.20 shows a sample flowchart of serial data transmission.
Transmission will not start while a receive error flag (ORER, FER, or PER in SSR) is set to 1. Be sure to clear the receive error flags to 0 before starting transmission. Note that clearing the RE bit in SCR to 0 does not clear the receive error flags.

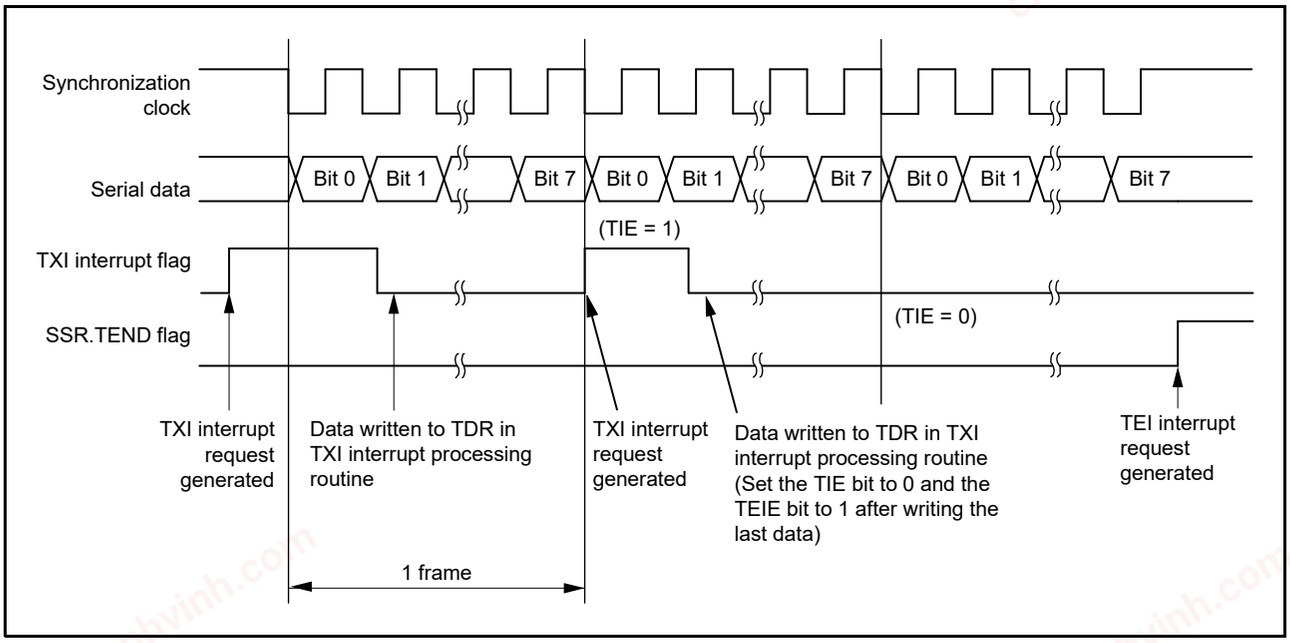
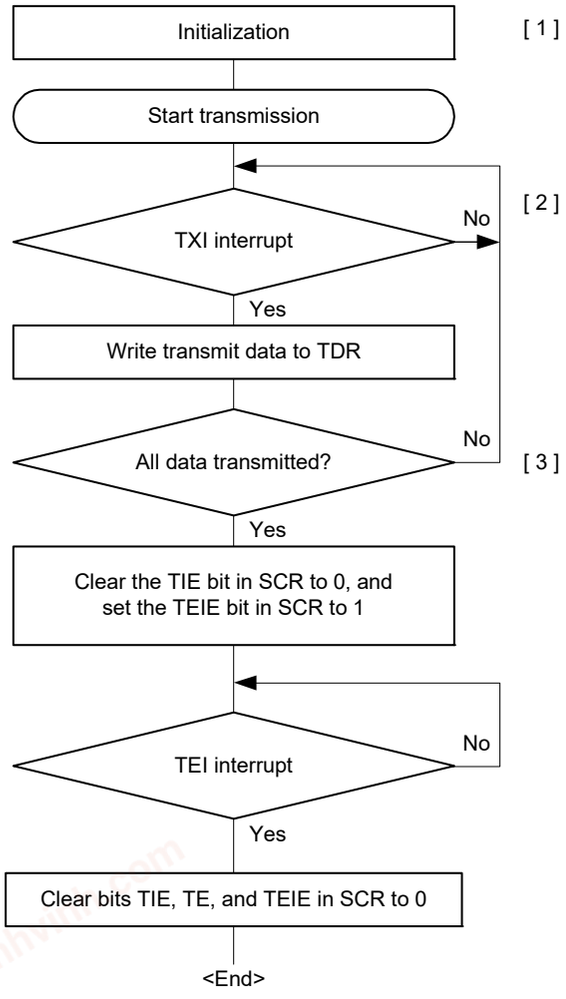


Figure 15.19 Example of Operation for Serial Transmission in Clock Synchronous Mode (from the Middle of Transmission until Transmission Completion)

chanhvinh.com



[1] Initialization:
The TXDn automatically becomes the output pin for data being transmitted.

[2] Writing transmit data write to TDR by a TXI interrupt request:
When transmit data is transferred from TDR to TSR, a transmit data empty interrupt (TXI) request is generated.
Transmit data is written to TDR once from the processing routine for TXI requests.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to TDR upon accepting a transmit data empty interrupt (TXI). Transmit data can also be written to TDR by activating the DMAC by the TXI interrupt request.
When TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted are written to the TDR.

Note: • When the external clock is in use (the value of the SCR.CKE[1:0] bits is 10b or 11b), the rising edge on the SCK pin for the last bit sets the SSR.TEND flag to 1. Clearing the SCR.TE bit to 0 immediately after this may lead to insufficient received-data hold time on the receiver side.

Figure 15.20 Example of Serial Transmission Flowchart (Clock Synchronous Mode)

chanhvinh.com

15.5.5 Serial Data Reception (Clock Synchronous Mode)

Figure 15.21 and Figure 15.22 show examples of SCI operation for serial reception in clock synchronous mode. In serial data reception, the SCI operates as described below.

1. The value of the RE bit in SCR becoming 1 places the signal output on the RTS pin at the low level (when the RTS function is in use).
2. The SCI performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and stores the receive data in RSR.
3. If an overrun error occurs, the ORER bit in SSR is set to 1. If the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR.
4. When reception finishes successfully, receive data is transferred to RDR. If the RIE bit in SCR is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to RDR in this RXI interrupt processing routine before reception of the next receive data is completed. Reading out the received data that have been transferred to RDR causes the RTSn# pin to output the low level (when the RTS function is in use).

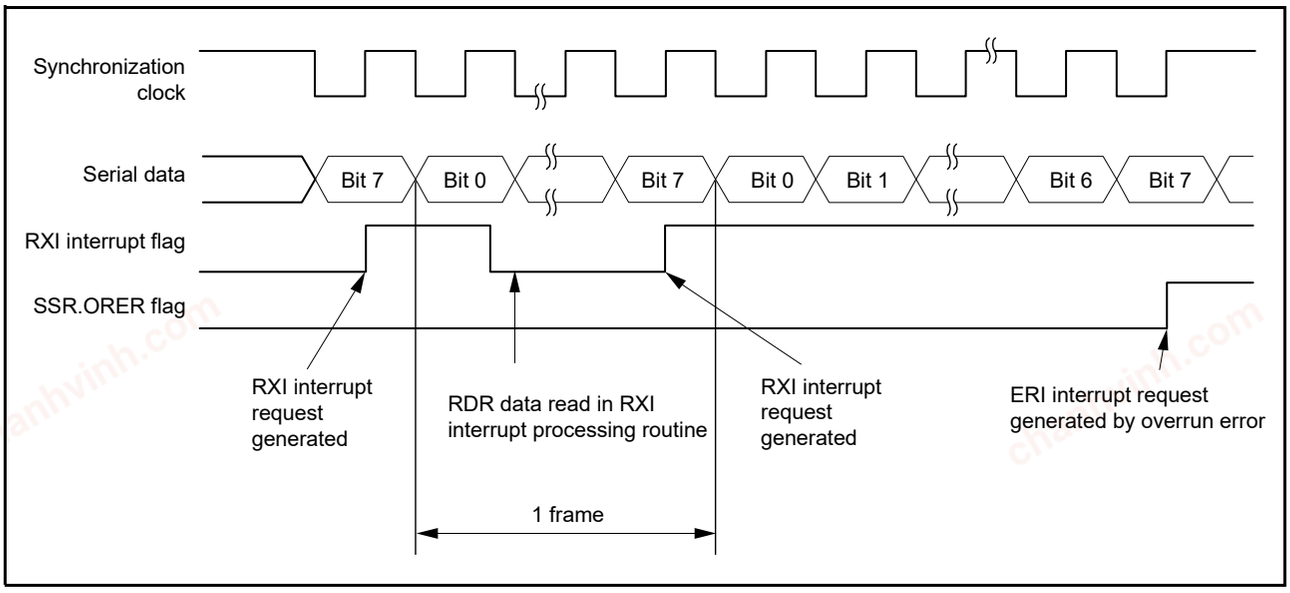


Figure 15.21 Example of Operation for Serial Reception in Clock Synchronous Mode (1) (when RTS Function is not Used)

chanhvinh.com

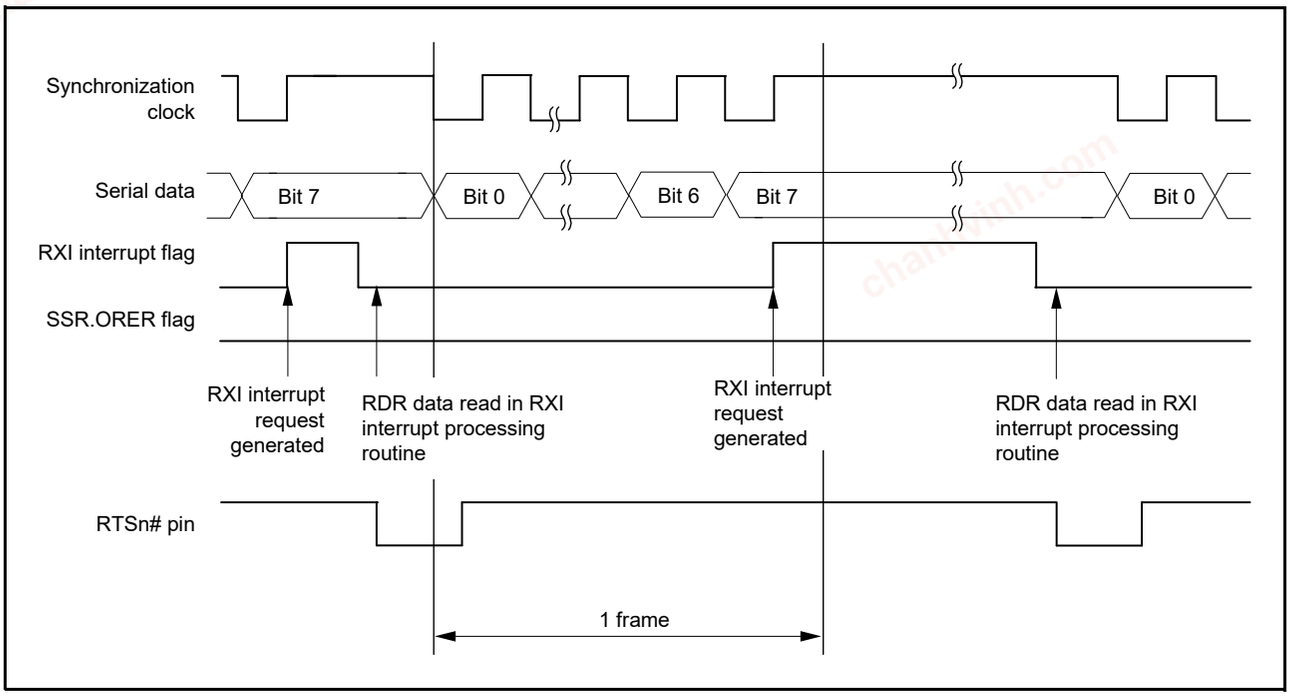


Figure 15.22 Example of Operation for Serial Reception in Clock Synchronous Mode (2) (when RTS Function is Used)

Data transfer cannot be resumed while a receive error flag is 1. Accordingly, clear the ORER, FER, and PER bits in SSR to 0 before resuming reception. Moreover, be sure to read the RDR during overrun error processing.

Figure 15.23 shows a sample flowchart for serial data reception.

chanhvinh.com

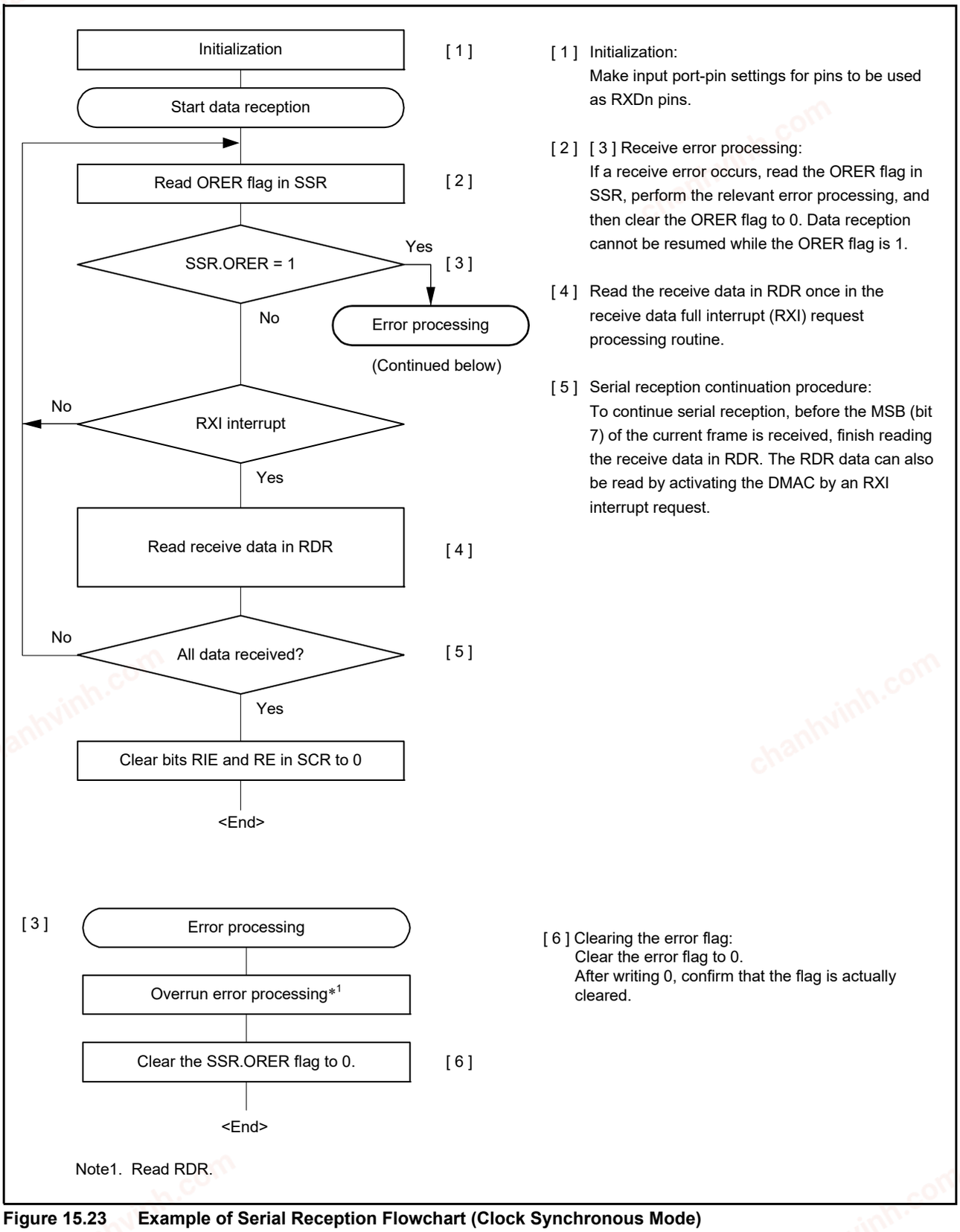


Figure 15.23 Example of Serial Reception Flowchart (Clock Synchronous Mode)

chanhvinh.com

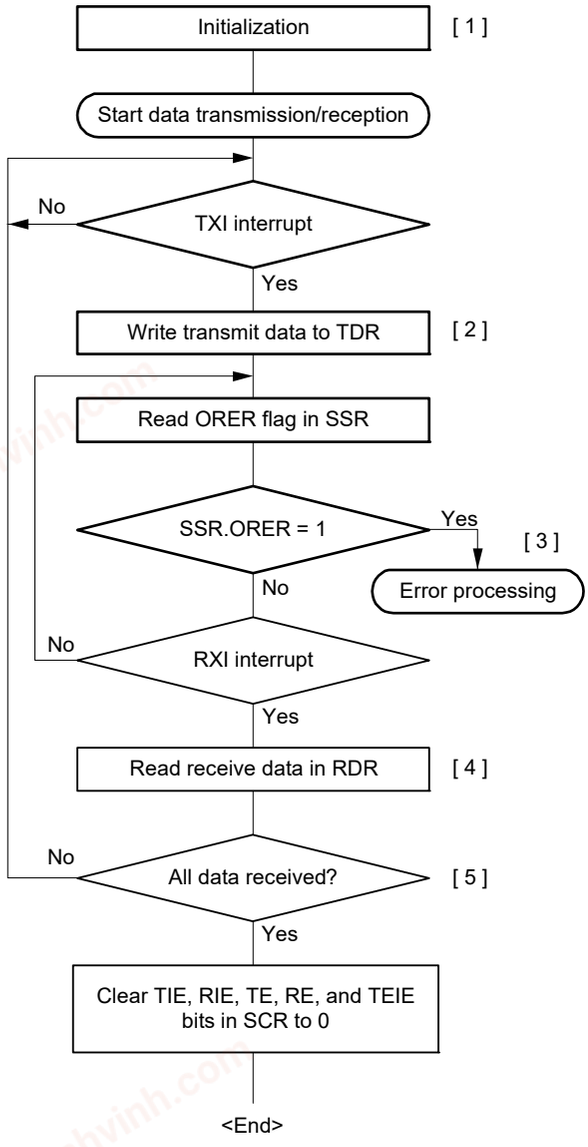
15.5.6 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode)

Figure 15.24 shows a sample flowchart for simultaneous serial transmit and receive operations in clock synchronous mode.

After initializing the SCI, the following procedure should be used for simultaneous serial data transmit and receive operations.

To switch from transmit mode to simultaneous transmit and receive mode, check that the SCI has finished transmission by reading that the TEND flag in SSR is 1, and then initialize the SCR register. Then set the TIE, RIE, TE, and RE bits in SCR to 1 simultaneously by a single instruction.

To switch from receive mode to simultaneous transmit and receive mode, check that the SCI has finished reception, and then clear the RIE and RE bits to 0. Then check that the receive error flags (ORER, FER, and PER in SSR) are 0, and then set the TIE, RIE, TE, and RE bits in SCR to 1 simultaneously by a single instruction.



[1] Initialization:
The TXDn pin can act as the output pin for transmitted data and the RXDn pin can act as the input pin for received data at the same time.

[2] Transmit data write:
Write transmit data to TDR once in the TXI interrupt request processing routine.

[3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR, perform the relevant error processing, and then clear the ORER flag to 0. Data reception cannot be resumed while the ORER flag is 1.

[4] Reading receive data:
Read the receive data in RDR once in the RXI interrupt request processing routine.

[5] Serial transmission/reception continuation procedure:
To continue serial transmission and reception, before the MSB (bit 7) of the current frame is received, finish reading the receive data in RDR by the RXI interrupt. Also, before the MSB (bit 7) of the current frame is transmitted, write data to TDR by the TXI interrupt.
Transmit data can also be written to TDR by activating the DMAC by a transmit data empty interrupt (TXI) request. Similarly, the RDR data can also be read by activating the DMAC by a receive data full interrupt (RXI) request.

Note: • When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TIE, RIE, TE, RE, and TEIE bits in SCR to 0, and then set TIE, RIE, TE, and RE bits to 1 simultaneously.

Figure 15.24 Example of Simultaneous Serial Transmission and Reception Flowchart (Clock Synchronous Mode)

chanhvinh.com

15.6 Operation in Smart Card Interface Mode

The SCI supports the smart card (IC card) interface conforming to the ISO/IEC 7816-3 (Identification Card) standard. Smart card interface mode can be selected using the appropriate register.

15.6.1 Sample Connection

Figure 15.25 shows a sample connection between a smart card (IC card) and this LSI.
As in the figure, since this LSI communicates with an IC card using a single transmission line, interconnect the TXDn and RXDn pins and pull up the data transmission line to Vcc using a resistor.
Setting the TE and RE bits in SCR to 1 with an IC card disconnected enables closed transmission/reception allowing self-diagnosis.
To supply an IC card with the clock pulses generated by the SCI, input the SCKn pin output to the CLK pin of an IC card. The output port of this LSI can be used to output a reset signal.

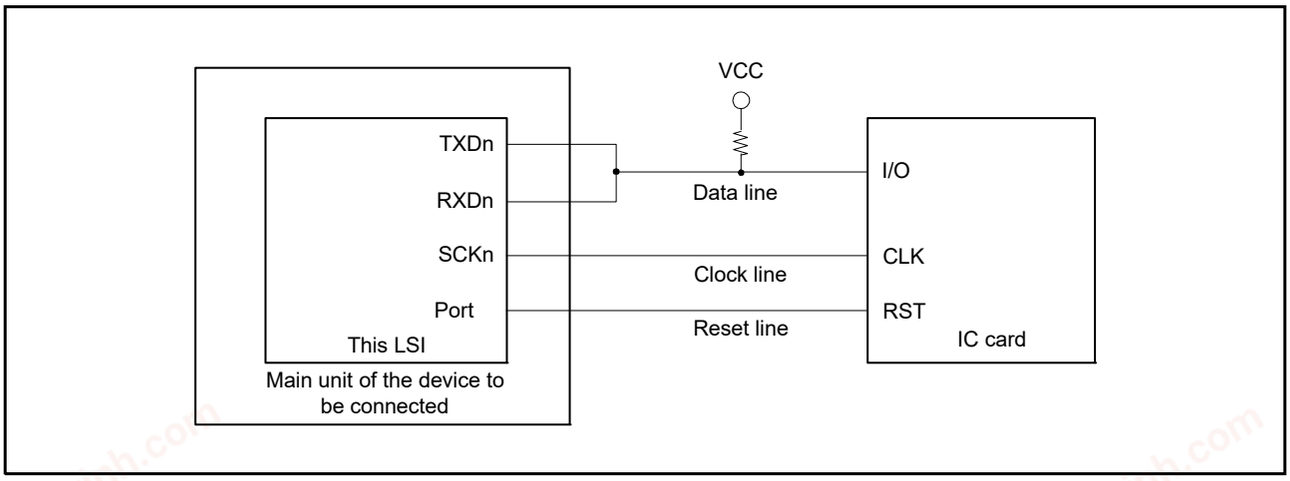


Figure 15.25 Sample Connection with a Smart Card (IC Card)

chanhvinh.com

15.6.2 Data Format (Except in Block Transfer Mode)

Figure 15.26 shows the data transfer formats in smart card interface mode.

- One frame consists of 8-bit data and a parity bit in asynchronous mode.
- During transmission, at least 2 etu (elementary time unit: time required for transferring one bit) is secured as a guard time from the end of the parity bit until the start of the next frame.
- If a parity error is detected during reception, a low-level error signal is output for 1 etu after 10.5 etu has passed from the start bit.
- If an error signal is sampled during transmission, the same data is automatically re-transmitted after at least 2 etu.

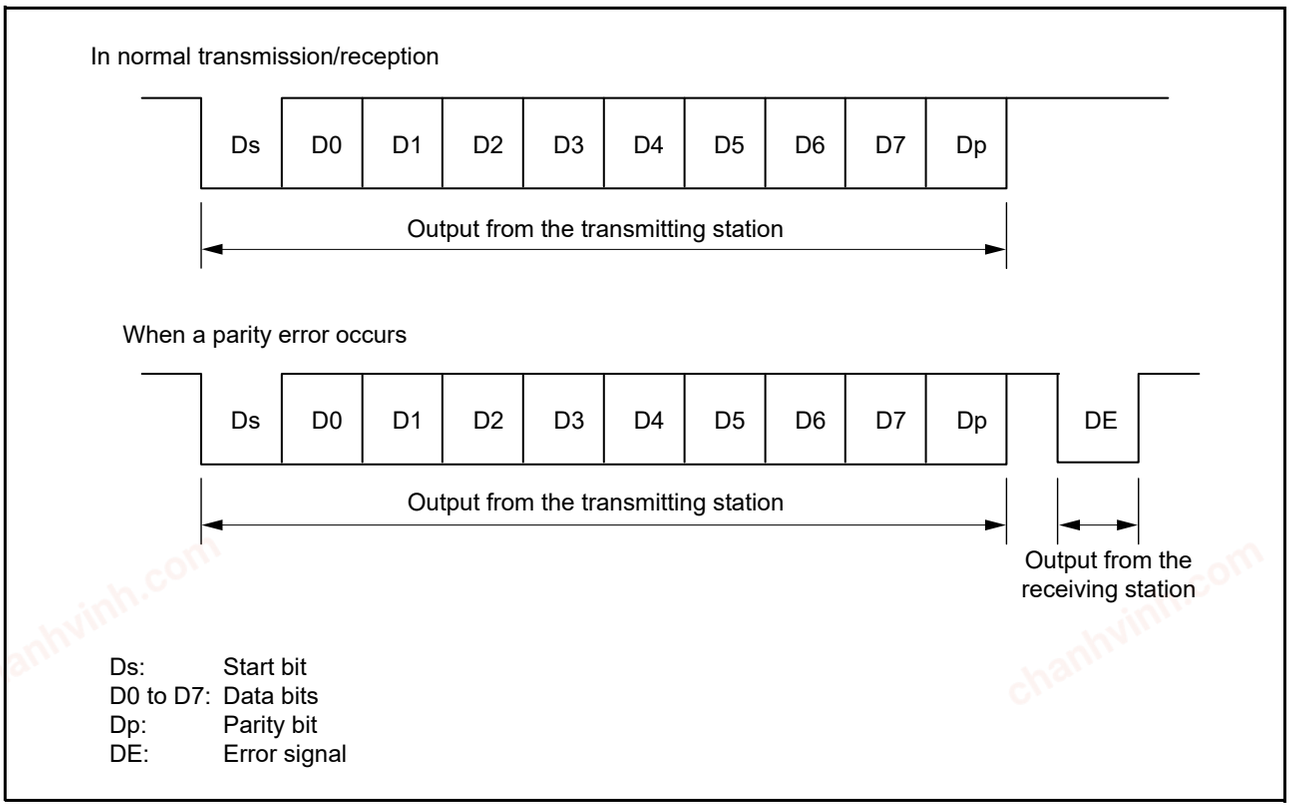


Figure 15.26 Data Formats in Smart Card Interface Mode

For communications with IC cards of the direct convention type and inverse convention type, follow the procedure below.

chanhvinh.com

(1) Direct Convention Type

For the direct convention type, logic levels 1 and 0 correspond to states Z and A, respectively, and data is transferred with LSB-first as the start character, as shown in Figure 15.27. Therefore, data in the start character in the figure is 3Bh. When using the direct convention type, write 0 to both the SDIR and SINV bits in SCMR. Write 0 to the PM bit in SMR in order to use even parity, which is prescribed by the smart card standard.

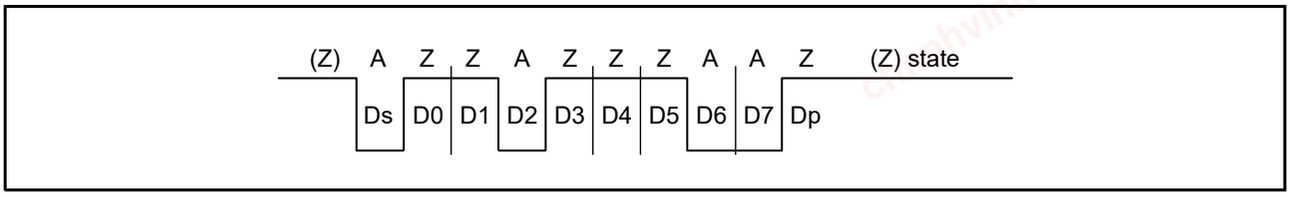


Figure 15.27 Direct Convention (SDIR in SCMR = 0, SINV in SCMR = 0, PM in SMR = 0)

(2) Inverse Convention Type

For the inverse convention type, logic levels 1 and 0 correspond to states A and Z, respectively and data is transferred with MSB-first as the start character, as shown in Figure 15.28. Therefore, data in the start character in the figure is 3Fh. When using the inverse convention type, write 1 to both the SDIR and SINV bits in SCMR. The parity bit is logic level 0 to produce even parity, which is prescribed by the smart card standard, and corresponds to state Z. Since the SCMR.SINV bit of this LSI only inverts data bits D7 to D0, write 1 to the PM bit in SMR to invert the parity bit for both transmission and reception.

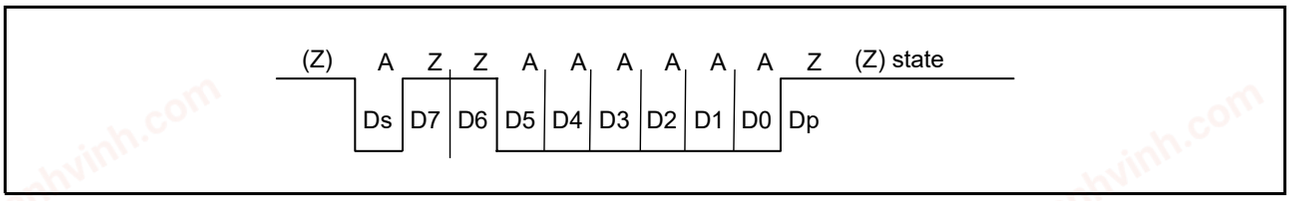


Figure 15.28 Inverse Convention (SDIR in SCMR = 1, SINV in SCMR = 1, PM in SMR = 1)

chanhvinh.com

15.6.3 Block Transfer Mode

Block transfer mode is different from normal smart card interface mode in the following respects.

- Even if a parity error is detected during reception, no error signal is output. Since the PER bit in SSR is set by error detection, clear the PER bit before receiving the parity bit of the next frame.
- During transmission, at least 1 etu is secured as a guard time from the end of the parity bit until the start of the next frame.
- Since the same data is not re-transmitted during transmission, the TEND flag in SSR is set 11.5 etu after transmission start.
- In block transfer mode, the ERS flag in SSR indicates the error signal status as in normal smart card interface mode, but the flag is always read as 0 because no error signal is transferred.

15.6.4 Receive Data Sampling Timing and Reception Margin

Only the internal clock generated by the on-chip baud rate generator can be used as a transfer clock in smart card interface mode.
In this mode, the SCI can operate on a base clock with a frequency of 32, 64, 372, 256, 93, 128, 186, or 512 times the bit rate according to the settings of the BCP2 bit in SCMR and the BCP[1:0] bits in SMR (the frequency is always 16 times the bit rate in normal asynchronous mode).
For data reception, the falling edge of the start bit is sampled with the base clock to perform internal synchronization. Receive data is sampled on the 16th, 32nd, 186th, 128th, 46th, 64th, 93rd, and 256th rising edges of the base clock so that it can be latched at the middle of each bit as shown in Figure 15.29. The reception margin here is determined by the following formula.

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\ F - \frac{\left| D - 0.5 \right|}{N}\ (1+F) \right| \times 100\ [\%]$$

M: Reception margin (%)
N: Ratio of bit rate to clock (N = 32, 64, 372, 256)
D: Duty cycle of clock (D = 0 to 1.0)
L: Frame length (L = 10)
F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5, and N = 372 in the above formula, the reception margin is determined by the formula below.

$$M = \{0.5 - 1/(2 \times 372)\} \times 100\ [\%] = 49.866\%$$

chanhvinh.com

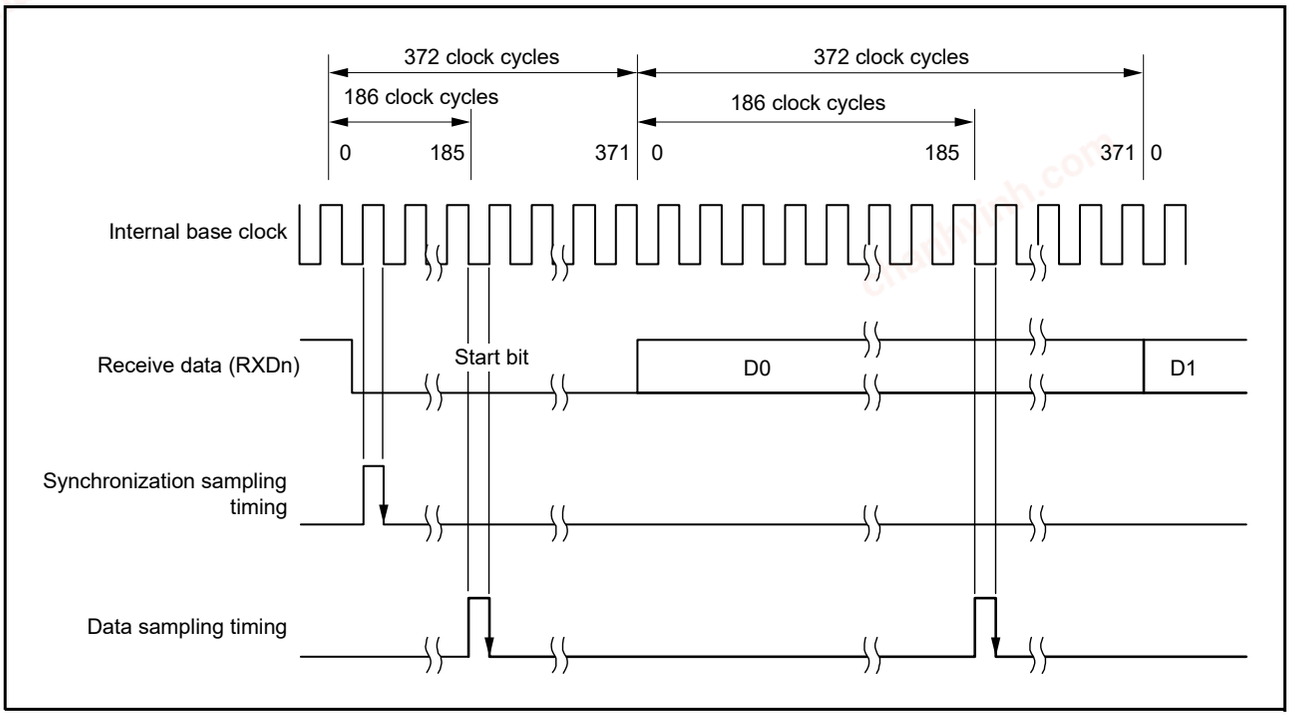


Figure 15.29 Receive Data Sampling Timing in Smart Card Interface Mode (When Clock Frequency is 372 Times the Bit Rate)

15.6.5 Initialization (Smart Card Interface Mode)

Before transmitting and receiving data, initialize the SCI using the following procedure. Initialization is also necessary before switching from transmission to reception and vice versa.

1. Write the initial value “00h” to the SCR.
2. Make general-purpose I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.
3. Set the error flags ORER, ERS, and PER in SSR to 0.
4. Set bits GM, BLK, PM, BCP[1:0], and CKS[1:0] in SMR and the BCP2 bit in SCMR appropriately. Also set the PE bit in SMR to 1.
5. Set bits SDIR, SINV, and SMIF in SCMR appropriately. Then, the TXDn and RXDn pins are placed in the high impedance state.
6. Set the value corresponding to the bit rate in BRR.
7. Set the CKE[1:0] bits in SCR appropriately, and set bits TIE, RIE, TE, RE, and TEIE in SCR to 0 at the same time. When the CKE[1:0] bit in SCR is set to 1, the SCKn pin is allowed to output clock pulses.
8. Set the TIE, RIE, TE, and RE bits in SCR to 1. Setting the TE and RE bits to 1 simultaneously is prohibited except for self-diagnosis.

To change reception mode to transmission mode, first check that reception has completed, and then initialize the SCI. At the end of initialization, set SCR.TE = 1 and SCR.RE = 0. Reception completion can be verified by reading the RXI request, ORER, or PER flag in SSR.
To change transmission mode to reception mode, first check that transmission has completed, and then initialize the SCI. At the end of initialization, set SCR.TE = 0 and SCR.RE = 1. Transmission completion can be verified by reading the TEND flag in SSR.

chanhvinh.com

15.6.6 Serial Data Transmission (Except in Block Transfer Mode)

Serial data transmission in smart card interface mode (except in block transfer mode), in that an error signal is sampled and data can be re-transmitted, is different from that in normal serial communications interface mode. Figure 15.30 shows the data retransfer operation during transmission.

1. When an error signal from the receiver end is sampled after one-frame data has been transmitted, the ERS flag in SSR is set to 1. If the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated. Clear the ERS flag in SSR to 0 before the next parity bit is sampled.
2. For a frame in which an error signal is received, the TEND flag in SSR is not set. Data is retransferred from TDR to TSR allowing automatic data retransmission.
3. If no error signal is returned from the receiver, the ERS flag in SSR is not set to 1.
4. In this case, the SCI judges that transmission of one-frame data (including retransfer) has been completed, and the TEND flag is set. If the TIE bit in SCR is 1 at this time, a TXI interrupt request is generated. Writing transmit data to TDR starts transmission of the next data.

Figure 15.32 shows a sample flowchart of serial transmission. All the processing steps are automatically performed using a TXI interrupt request to activate the DMAC.
When the TEND flag in SSR is set to 1 in transmission, if the TIE bit in SCR is 1, a TXI interrupt request is generated. The DMAC is activated by a TXI interrupt request if the TXI interrupt request is specified as a source of DMAC activation beforehand, allowing transfer of transmit data. The TEND flag in SSR is automatically cleared to 0 when the DMAC transfers the data.
If an error occurs, the SCI automatically re-transmits the same data. During this retransmission, the TEND flag in SSR is kept to 0 and the DMAC is not activated. Therefore, the SCI and DMAC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, since the ERS flag in SSR is not automatically cleared, set the RIE bit in SCR to 1 beforehand to enable an ERI interrupt request to be generated at error occurrence, and clear the ERS flag in SSR to 0.
When transmitting/receiving data using the DMAC, be sure to make settings to enable the DMAC before making SCI settings.
For DMAC settings, see section 9, Direct Memory Access Controller.

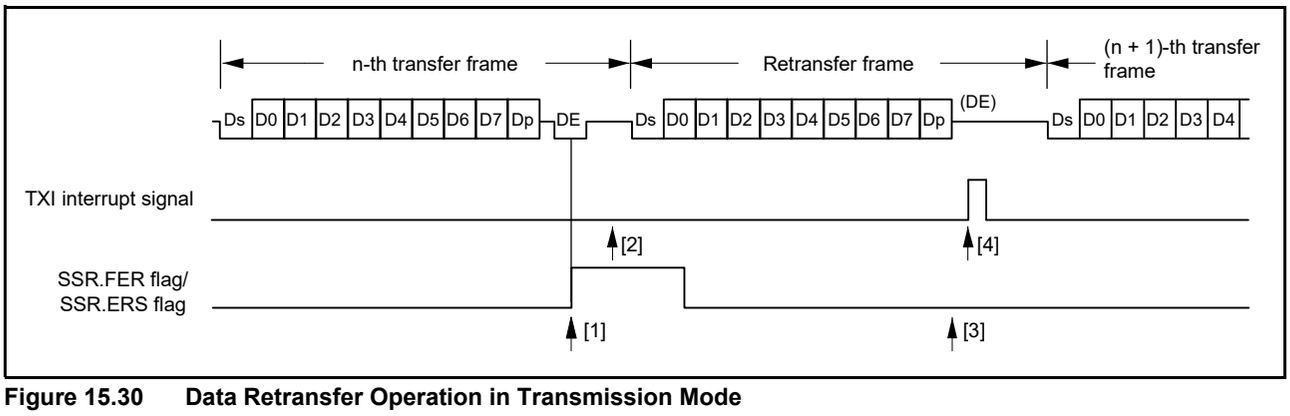


Figure 15.30 Data Retransfer Operation in Transmission Mode

chanhvinh.com

Note that the SSR.TEND flag is set in different timings depending on the GM bit setting in SMR. Figure 15.31 shows the SSR.TEND flag generation timing.

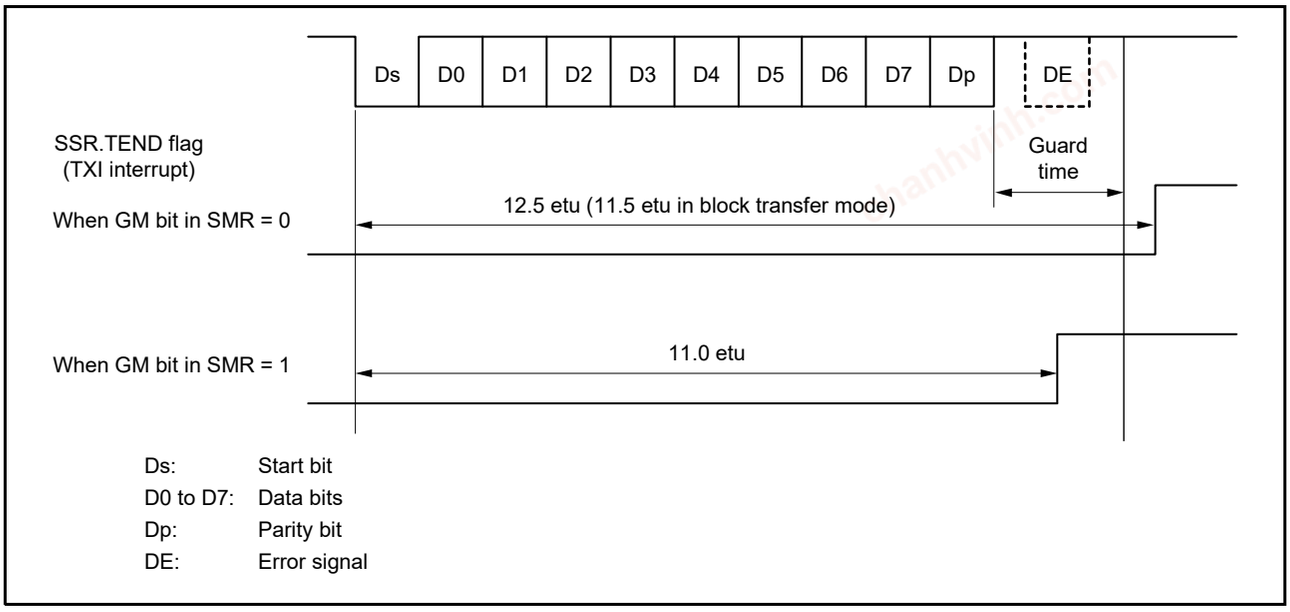


Figure 15.31 SSR.TEND Flag Generation Timing during Transmission

chanhvinh.com

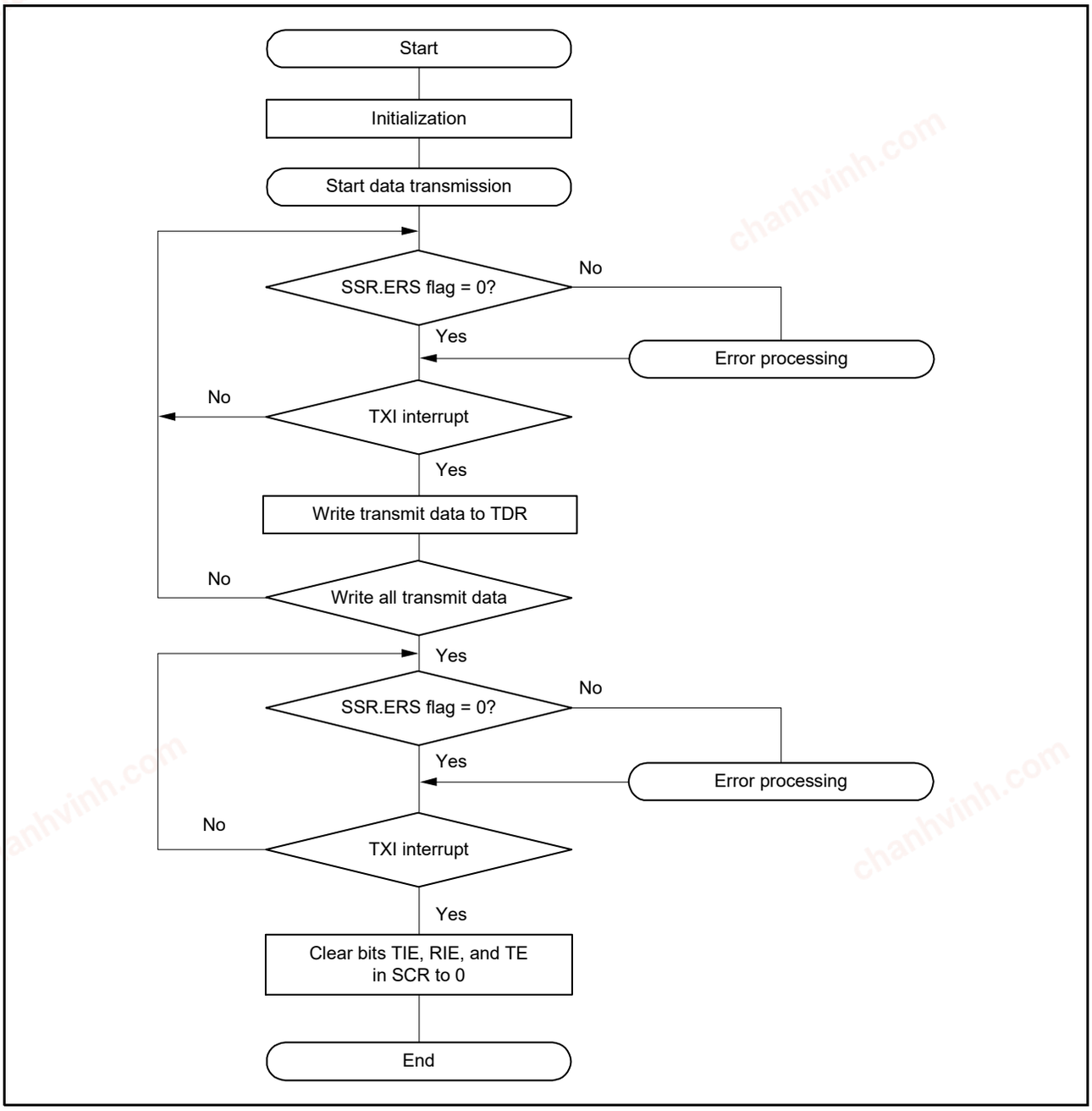


Figure 15.32 Sample Smart Card Interface Transmission Flowchart

chanhvinh.com

15.6.7 Serial Data Reception (Except in Block Transfer Mode)

Serial data reception in smart card interface mode is similar to that in serial communications interface mode. Figure 15.33 shows the data retransfer operation in reception mode.

1. If a parity error is detected in receive data, the PER flag in SSR is set to 1. When the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated. Clear the PER flag in SSR to 0 before the next parity bit is sampled.
2. For a frame in which a parity error is detected, no RXI interrupt is generated.
3. When no parity error is detected, the PER flag in SSR is not set to 1.
4. In this case, data is determined to have been received successfully. When the RIE bit in SCR is 1, an RXI interrupt request is generated.

Figure 15.34 shows a sample flowchart for serial data reception. All the processing steps are automatically performed using an RXI interrupt request to activate the DMAC.

In reception, setting the RIE bit in SCR to 1 allows an RXI interrupt request to be generated. The DMAC is activated by an RXI interrupt request if the RXI interrupt request is specified as a source of DMAC activation beforehand, allowing transfer of receive data.

If an error occurs during reception and either the ORER or PER flag in SSR is set to 1, a receive error interrupt (ERI) request is generated. Clear the error flag after the error occurrence. If an error occurs, the DMAC is not activated and receive data is skipped. Therefore, the number of bytes of receive data specified in the DMAC is transferred.

Even if a parity error occurs and the PER flag in SSR is set to 1 during reception, receive data is transferred to RDR, thus allowing the data to be read.

Note 1. For operations in block transfer mode, see section 15.3, Operation in Asynchronous Mode.

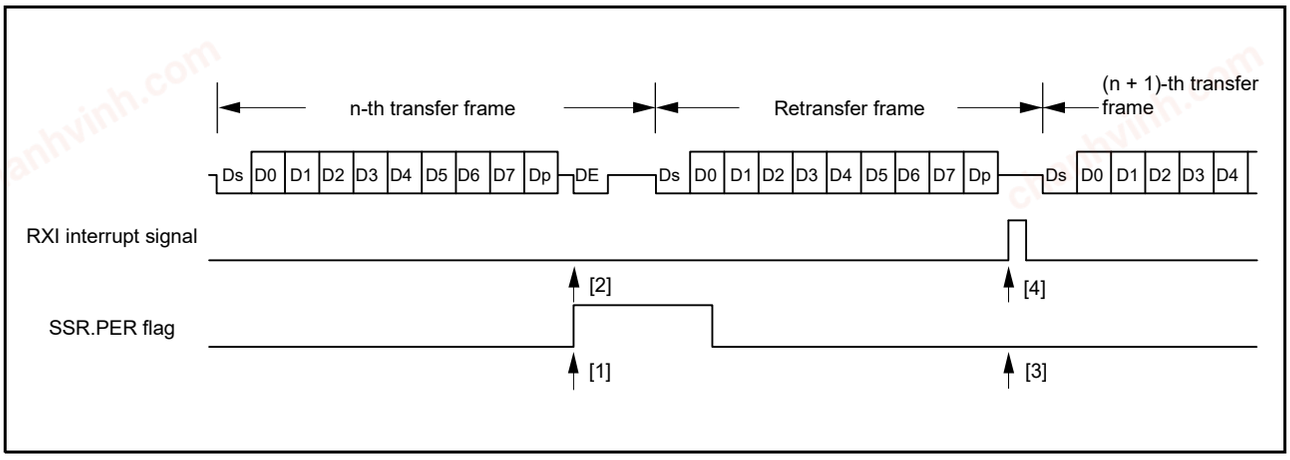


Figure 15.33 Data Retransfer Operation in Reception Mode

chanhvinh.com

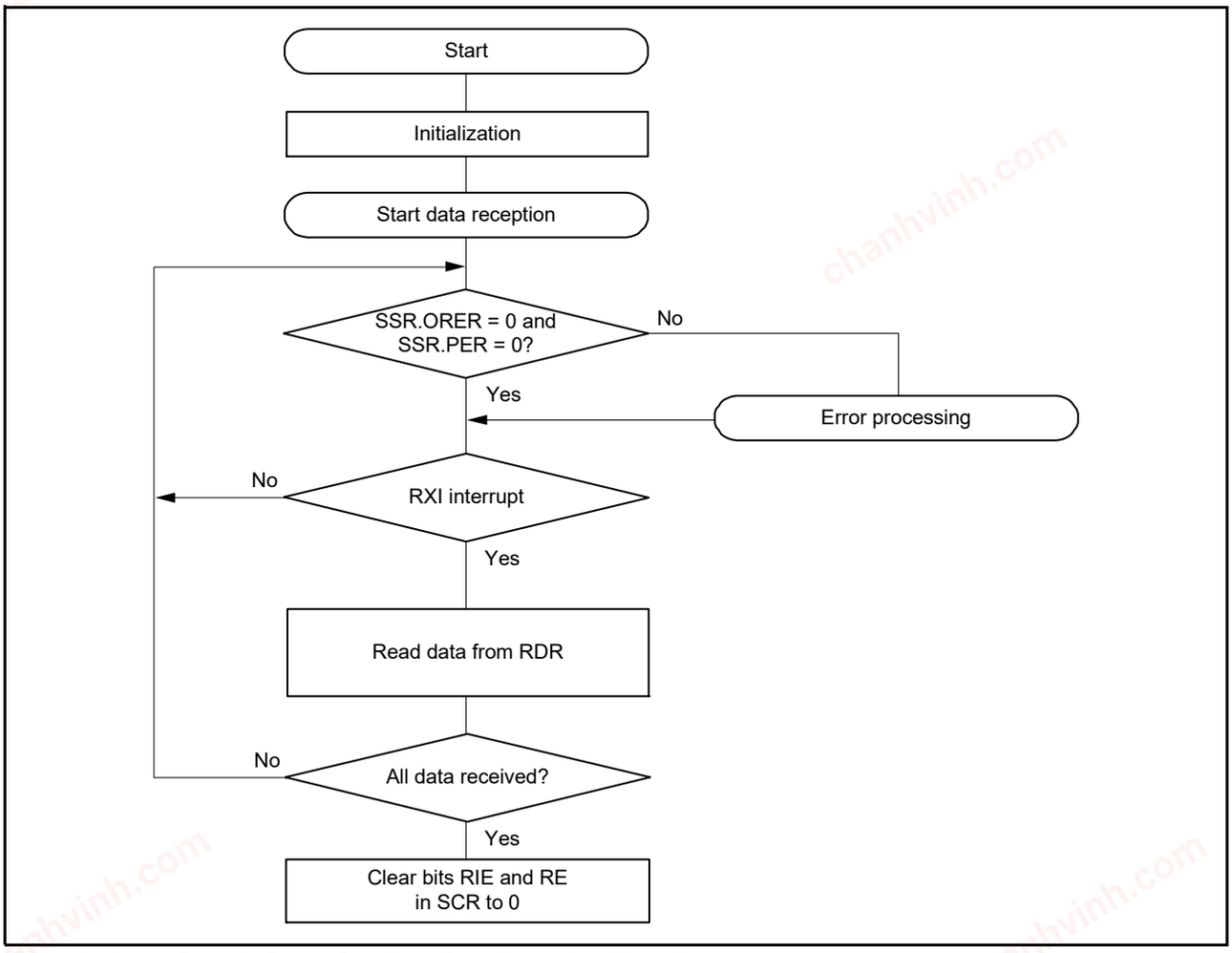


Figure 15.34 Sample Smart Card Interface Reception Flowchart

15.6.8 Clock Output Control

Clock output can be fixed using the CKE[1:0] bits in SCR when the GM bit in SMR is 1. Specifically, the minimum width of a clock pulse can be specified.

Figure 15.35 shows an example of clock output fixing timing when the CKE0 bit is controlled with SMR.GM = 1 and SCR.CKE1 = 0.

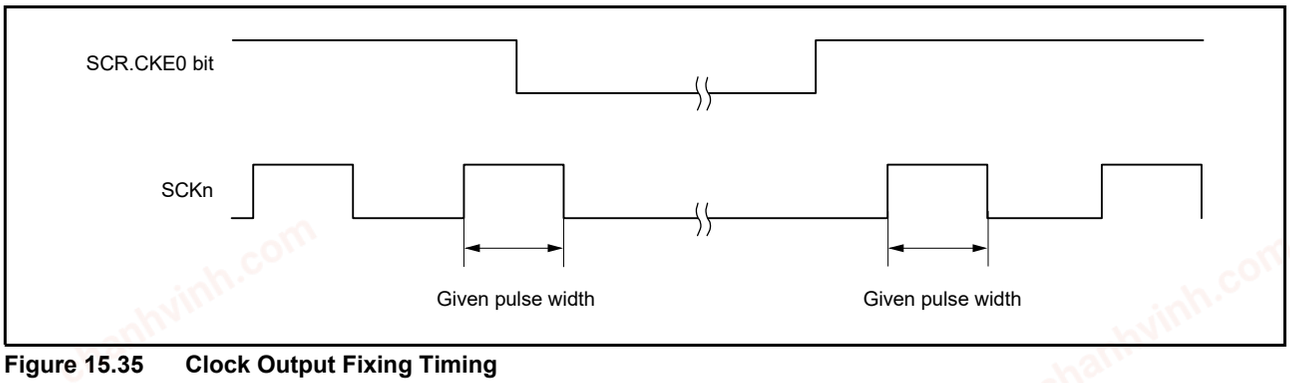


Figure 15.35 Clock Output Fixing Timing

chanhvinh.com

At power-on and transitions to/from software standby mode, use the following procedure to secure the appropriate clock duty cycle.

(1) At Power-On

To secure the appropriate clock duty cycle simultaneously with power-on, use the following procedure.

1. Initially, port input is enabled in the high-impedance state. To fix the potential level, use a pull-up or pull-down resistor.
2. Fix the SCKn pin to the specified output by setting the SCR.CKE[1] bit and general-purpose I/O port.
3. Set SMR and SCMR to enable smart card interface mode.
4. Set the SCR.CKE[0] bit to 1 to start clock output.

(2) At Mode Switching

(a) At transition from smart card interface mode to software standby mode

1. Set low power consumption mode to make the SCKn pin fixed with a desired output value in software standby mode.
2. Write 0 to the TE and RE bits in SCR to stop transmission/reception.
   Simultaneously, set the SCR.CKE[1] bit to the value for the output fixed state in software standby mode.
3. Write 0 to the SCR.CKE[0] bit to stop the clock.
4. Wait for one cycle of the serial clock. In the mean time, the clock output is fixed to the specified level with the duty cycle retained.
5. After switching the SCKn pin to the general-purpose I/O port function, make a transition to software standby mode.

(b) Return from software standby mode to smart card interface mode

6. Cancel software standby mode.
7. Set the SCR.CKE[0] bit to 1 to start clock output. A clock signal with the appropriate duty cycle is then generated.

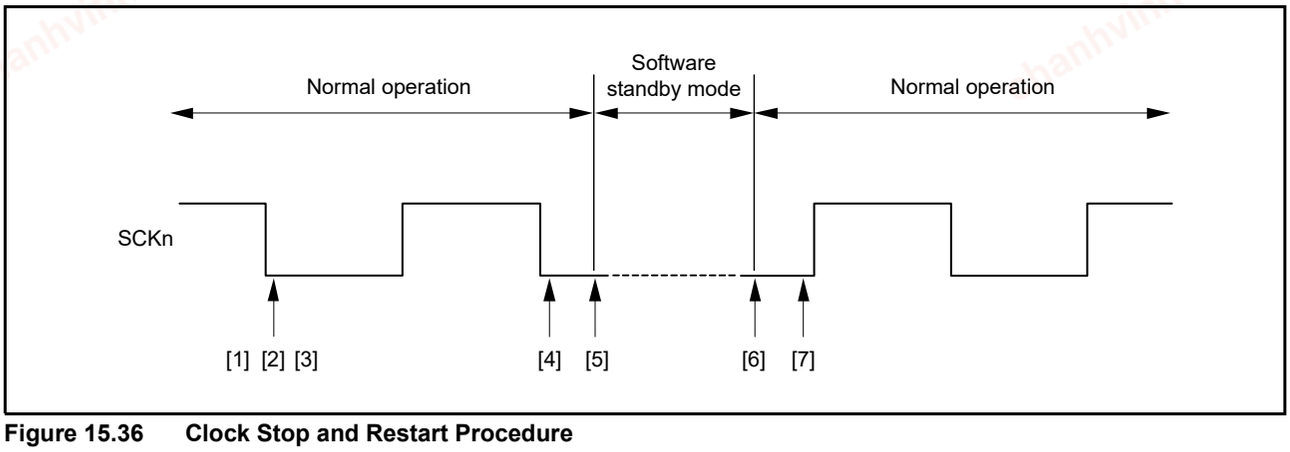


Figure 15.36 Clock Stop and Restart Procedure

chanhvinh.com

15.7 Noise Cancellation Function

Figure 15.37 shows the configuration of the noise filter used for noise cancellation. The noise filter consists of two stages of flip-flop circuits and a match-detection circuit. When the level on the pin matches in three consecutive samples taken at the set sampling interval, the matching level continues to be conveyed internally until the level on the pin again matches in three consecutive samples.

In asynchronous mode, the noise cancellation function can be applied on the RXDn input signal. The period of the base clock (1/16th of a bit-period when SEMR.ABCS = 0 and 1/8th of a bit-period when SEMR.ABCS = 1) is the sampling interval.

If the base clock is stopped with the noise filter enabled and then the clock input is started again, the noise filter operation resumes from where the clock was stopped. If SCR.TE and SCR.RE are set to 0 during base clock input, all of the noise filter flip-flop values are initialized to 1. Accordingly, if the input data is 1 when reception operation resumes, it is determined that a level match is detected and is conveyed to the internal signal. When the level being input corresponds to 0, the initial output of the noise filter is retained until the level matches in three consecutive samples.

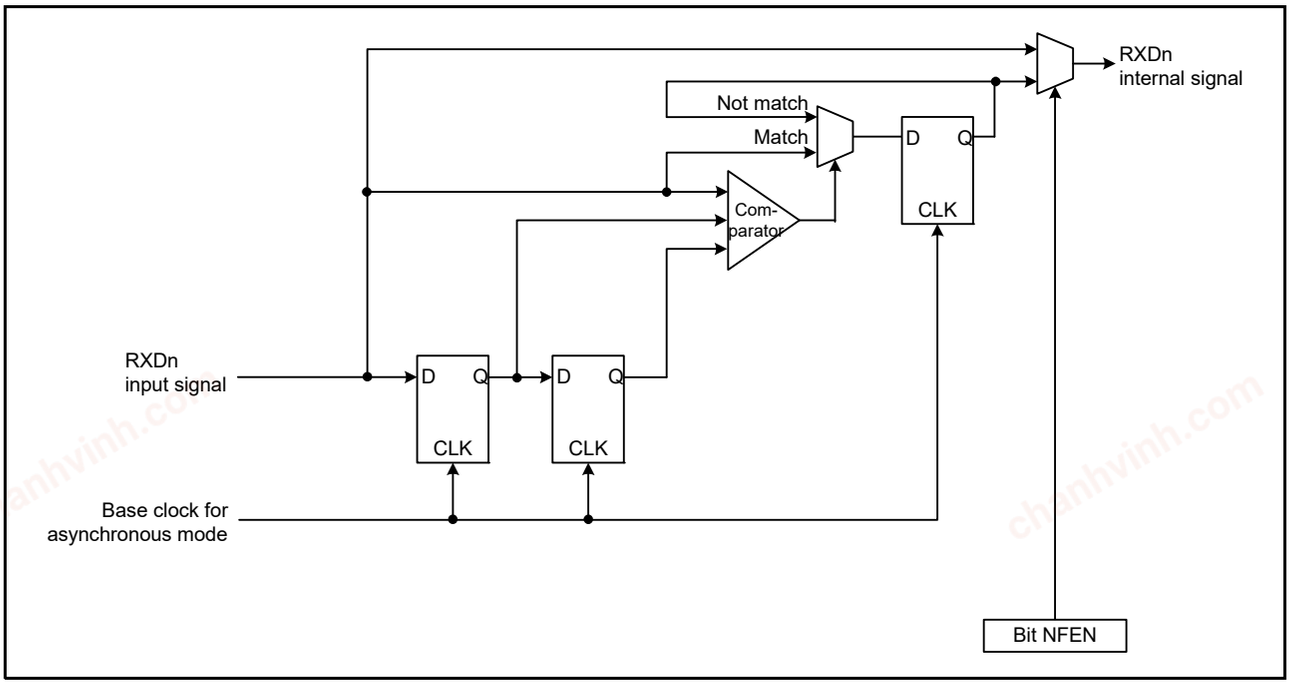


Figure 15.37 Block Diagram of Digital Noise Filter Circuit

chanhvinh.com

15.8 Interrupt Sources

15.8.1 Interrupts in Serial Communications Interface Mode

Table 15.16 lists interrupt sources in serial communication interface mode. Individual interrupt sources can be enabled or disabled with the enable bits in SCR.

If the SCR.TIE bit is 1, a TXI interrupt request is generated when data for transmission are transferred from the TDR to the TSR. A TXI interrupt request can also be generated by setting the SCR.TE bit to 1 after setting the SCR.TIE bit to 1 or by using a single instruction to set the SCR.TE and SCR.TIE bit to 1 at the same time. A TXI interrupt request can activate the DMAC to handle data transfer.

A TXI interrupt request is not generated by setting the SCR.TE bit to 1 while the setting of the SCR.TIE bit is 0 or by setting the SCR.TIE bit to 1 while the setting of the SCR.TE bit is 1.*1

When new data are not written by the time of transmission of the last bit of the current data for transmission and the setting of the SCR.TEIE bit is 1, the SSR.TEND flag becomes 1 and a TEI interrupt request is generated. Furthermore, when the setting of the SCR.TE bit is 1, the SSR.TEND flag retains the value 1 until further data for transmission are written to the TDR, and setting the SCR.TEIE bit to 1 leads to the generation of a TEI interrupt request.

Writing data to the TDR leads to clearing of the SSR.TEND flag and, after a certain time, discarding of the TEI interrupt request.

If the SCR.RIE bit is 1, an RXI interrupt request is generated when received data are stored in the RDR. An RXI interrupt request can activate the DMAC to handle data transfer.

Setting of any from among the ORER, FER, and PER flags in the SSR to 1 while the SCR.RIE bit is 1 leads to the generation of an ERI interrupt request. An RXI interrupt request is not generated at this time. Clearing all three flags (ORER, FER, and PER) leads to discarding of the ERI interrupt request.

Note 1. To temporarily prohibit TXI interrupts at the time of transmission of the last of the data and so on when you wish a new round of transmission to start after handling of the transmission-completed interrupt, control prohibiting and permitting of the interrupt by using the interrupt request enable bit in the interrupt controller rather than using the SCR.TIE bit. This can prevent the suppression of TXI interrupt requests in the transfer of new data.

Table 15.16 Interrupt Sources

Name	Interrupt Source	Interrupt Flag	DMAC Activation
ERI	Receive error	ORER, FER, or PER	Not possible
RXI	Receive data full	—	Possible
TXI	Transmit data empty	—	Possible
TEI	Transmit end	TEND	Not possible

chanhvinh.com

15.8.2 Interrupts in Smart Card Interface Mode

Table 15.17 lists interrupt sources in smart card interface mode. A transmit end interrupt (TEI) request cannot be used in this mode.

Table 15.17 Interrupt Sources

Name	Interrupt Source	Interrupt Flag	DMAC Activation
ERI	Receive error or error signal detection	ORER, PER, or ERS	Not possible
RXI	Receive data full	—	Possible
TXI	Transmit data empty	TEND	Possible

Data transmission/reception using the DMAC is also possible in smart card interface mode. In transmission, when the TEND flag in SSR is set to 1, a TXI interrupt request is generated. This TXI interrupt request activates the DMAC allowing transfer of transmit data if the TXI request is specified beforehand as a source of DMAC activation. The TEND flag in SSR is automatically cleared to 0 when the DMAC transfers the data.
If an error occurs, the SCI automatically re-transmits the same data. During the retransmission, the TEND flag in SSR is kept to 0 and the DMAC is not activated. Therefore, the SCI and DMAC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, the ERS flag in SSR is not automatically cleared to 0 at error occurrence. Therefore, the ERS flag in SSR must be cleared by previously setting the RIE bit in SCR to 1 to enable an ERI interrupt request to be generated at error occurrence.
When transmitting/receiving data using the DMAC, be sure to make settings to enable the DMAC before making SCI settings. For DMAC settings, see section 9, Direct Memory Access Controller.

In reception, an RXI interrupt request is generated when receive data is set to RDR. This RXI interrupt request activates the DMAC allowing transfer of receive data if the RXI request is specified beforehand as a source of DMAC activation. If an error occurs, the error flag is set. Therefore, the DMAC is not activated and an ERI interrupt request is issued to the CPU instead; the error flag must be cleared.

chanhvinh.com

15.9 Usage Notes

15.9.1 Setting the Module Standby Function

SCI operation can be started and stopped by setting the module standby mode. With the value after a reset, SCI operations are stopped. The registers of the modules only become accessible after release from the module standby state. For details, refer to section 42, Power-Down Modes.

15.9.2 Break Detection and Processing

When a framing error is detected, a break can be detected by reading the RXDn pin value directly. In a break, the input from the RXDn pin becomes all 0s, and so the FER flag in SSR is set to 1 (framing error), and the PER flag in SSR may also be set to 1 (parity error). The SCI continues the receive operation even after a break is received. Therefore, note that even if the FER flag in SSR is cleared to 0 (no framing error), it will be set to 1 again.

15.9.3 The Mark State and Production of Breaks

When the SCR.TE bit is 0 (prohibiting serial transmission), setting the general-purpose I/O port makes selection of the level and direction (input or output) of the TXDn pin possible. If this is done, the TXDn pin can be placed in the mark state to send a break at the time of data transmission. Until the SCR.TE bit is set to 1 (permitting serial transmission), the general-purpose I/O port is used to set the TXDn pin to output 1 and set the pin mode to a general-purpose I/O port pin, and thus place the transfer circuit in the mark state (state of having the value 1). On the other hand, to output a break at the time of data transmission, set the TXDn pin to output 0 and make the pin mode settings for a general-purpose I/O port pin. When the SCR.TE bit is set to 0, the transmission section is initialized regardless of the current state of transmission.

15.9.4 Receive Error Flags and Transmit Operations (Clock Synchronous Mode Only)

Transmission cannot be started when a receive error flag (ORER) in SSR is set to 1, even if data is written to TDR. Be sure to clear the receive error flags to 0 before starting transmission. Note also that the receive error flags cannot be cleared to 0 even if the RE bit in SCR is cleared to 0 (serial reception disabled).

15.9.5 Writing Data to TDR

Data can always be written to TDR. However, if new data is written to TDR when transmit data is remaining in TDR, the previous data in TDR is lost because it has not been transferred to TSR yet. Be sure to write transmit data to TDR in the TXI interrupt request processing routine.

15.9.6 Restrictions on Clock Synchronous Transmission

When the external clock source is used as a synchronization clock, update TDR by the DMAC and wait for at least five P1ϕ clock cycles before allowing the transmit clock to be input. If the transmit clock is input within four clock cycles after TDR is updated, the SCI may malfunction.

15.9.7 Restrictions on Using DMAC

When using the DMAC to read RDR, be sure to set the receive end interrupt (RXI) as the activation source of the relevant channel.

15.9.8 Points to Note on Starting Transfer

On the generation of an interrupt request for the interrupt controller at the point where transfer starts, follow the procedure below to clear interrupt requests before permitting operations (by setting the SCR.TE or SCR.RE bit to 1).

- Confirm that transfer has stopped (the setting of the SCR.TE or SCR.RE bits is 0).
- Set the corresponding interrupt enable bit (SCR.TIE or SCR.RIE) to 0.
- Read out the corresponding interrupt enable bit (SCR.TIE or SCR.RIE bit) to check that it has actually become 0.
- Clear the interrupt status flag in the interrupt controller to 0.

chanhvinh.com

15.9.9 SCI Operations during Low Power Consumption State

(1) Transmission

When making settings for the module standby state or in transitions to software standby, stop operations (by setting the TIE, TE, and TEIE bits in the SCR to 0) after switching the TXDn pin to the general-purpose I/O port pin function. Clearing the TE bit in SCR to 0 resets the TSR and the TEND bit in the SSR. The states of the output pins in the software standby mode depend on the settings for the power-down modes. When transitions to these states are made during transmission, the data being transmitted become indeterminate.

To transmit data in the same transmission mode after cancellation of the low power consumption state, set the TE bit in SCR to 1, read SSR, and write data to TDR sequentially to start data transmission. To transmit data with a different transmission mode, initialize the SCI first.

Figure 15.38 shows a sample flowchart for transition to software standby mode during transmission.

(2) Reception

Before specifying the module standby state or making a transition to software standby mode, stop the receive operations (RE = 0 in SCR). If transition is made during data reception, the data being received will be invalid.

To receive data in the same reception mode after cancellation of the low power consumption state, set the RE bit in SCR to 1, and then start reception. To receive data in a different reception mode, initialize the SCI first.

chanhvinh.com

Figure 15.39 shows a sample flowchart for transition to software standby mode during reception.

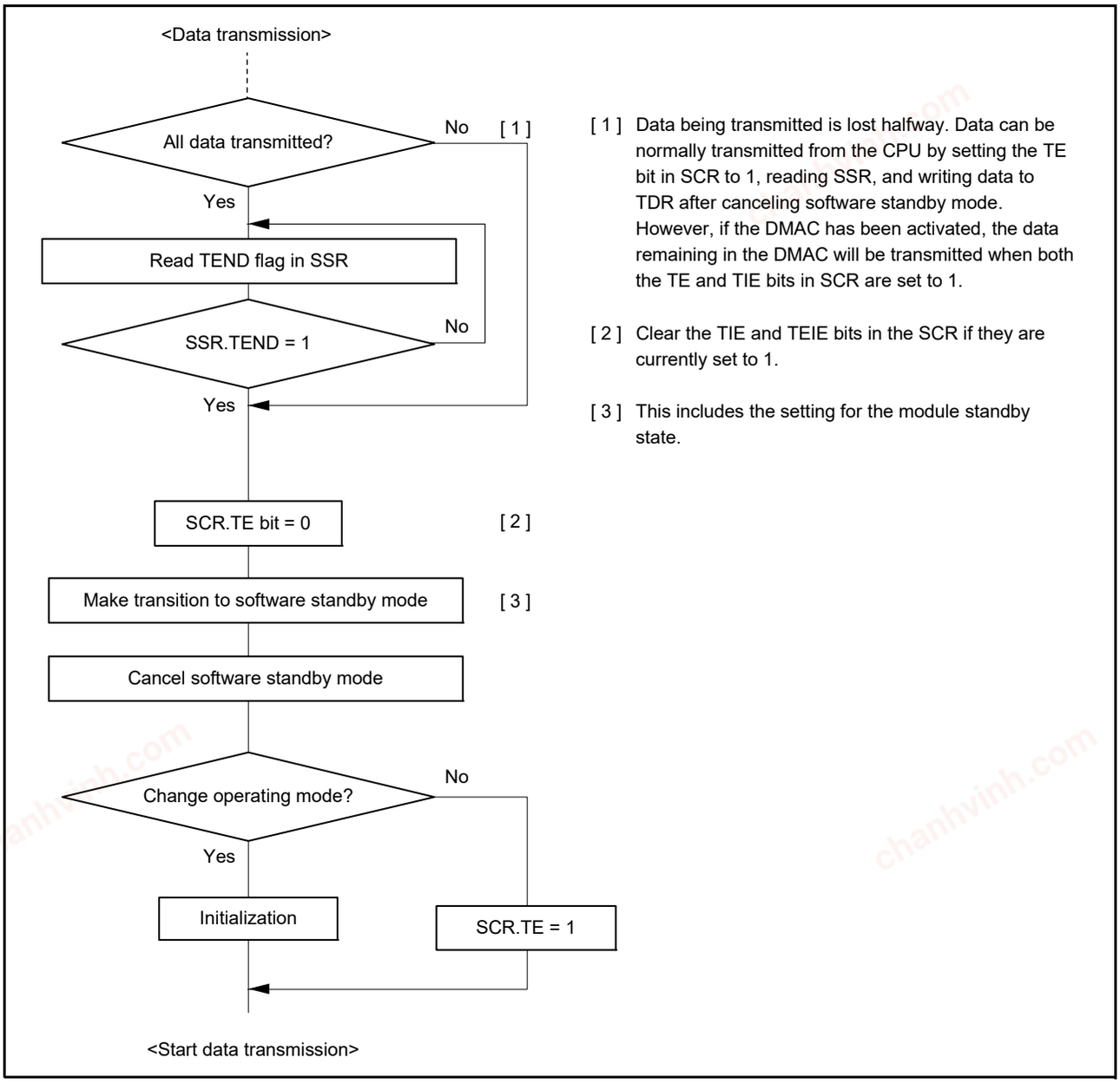


Figure 15.38 Example of Flowchart for Transition to Software Standby Mode during Transmission

chanhvinh.com

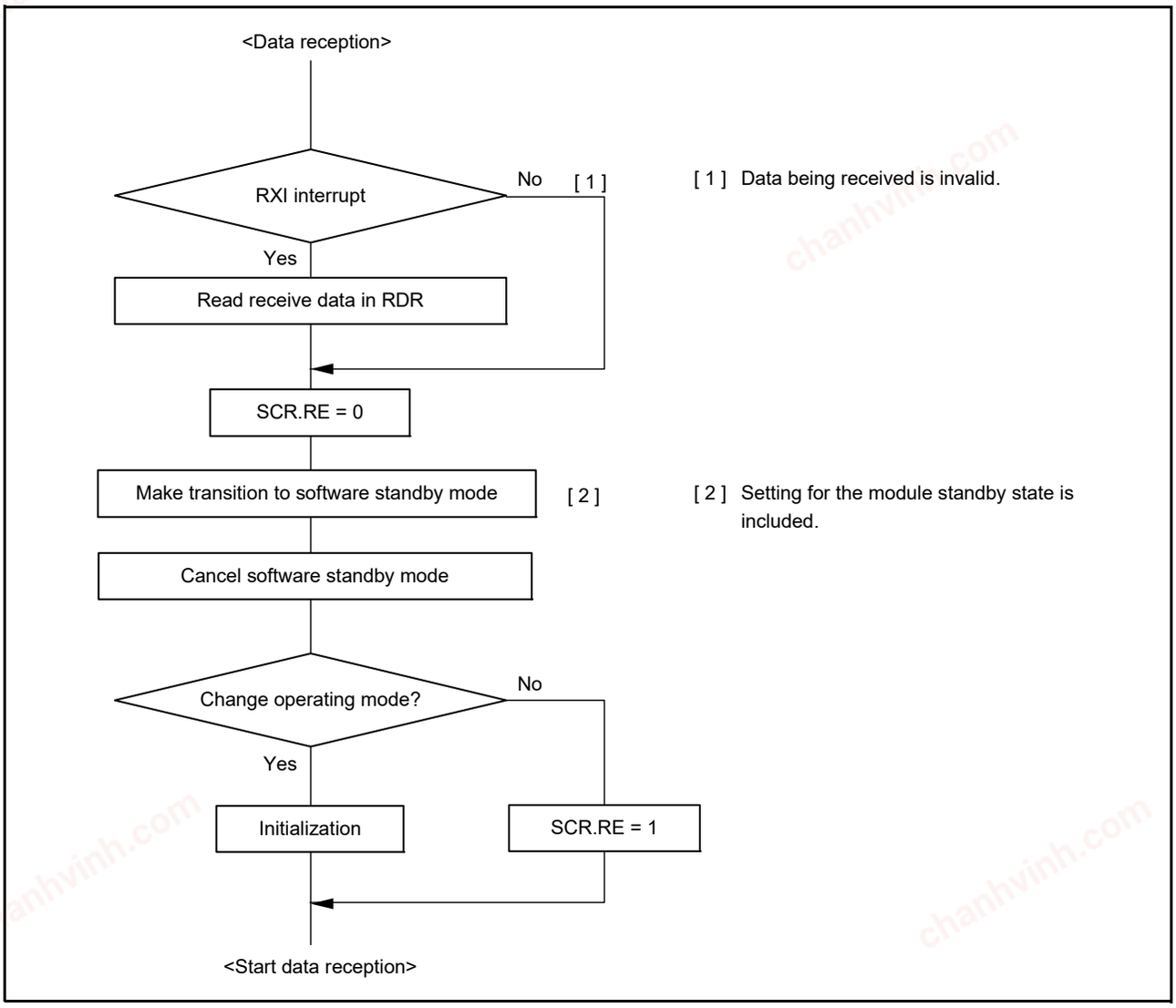


Figure 15.39 Example of Flowchart for Transition to Software Standby Mode during Reception

15.9.10 External Clock Input in Clock Synchronous Mode

In clock synchronous mode, the external clock SCKn must be input as follows:
High-pulse period, low-pulse period = 2 P1ϕ clock cycles or more, period = 6 P1ϕ clock cycles or more

chanhvinh.com

15.10 IrDA Communications

In combination with the on-chip IrDA module, the channel 0 serial communications interface (SCI) transmits and receives waveforms conforming with version 1.0 of the Infrared Data Association (IrDA) standard.

When the IrDA function is enabled by the IRE bit in the IRCR register, the SCI_TXD0 and SCI_RXD0 signals transmitted and received on channel 0 are encoded to and decoded from waveforms conforming with the standard. Connecting these signals to an infrared transceiver enables infrared transmission and reception conforming with the standard.

The standard allows starting communications at a transfer rate of 9600 bps and changing the rate as required. The IrDA module does not have the function of automatically changing the transfer rate. To change the transfer rate of IrDA communications, change the transfer rate of the SCI.

Figure 15.40 shows the block diagram. Table 15.18 shows the pin configuration of the IrDA.

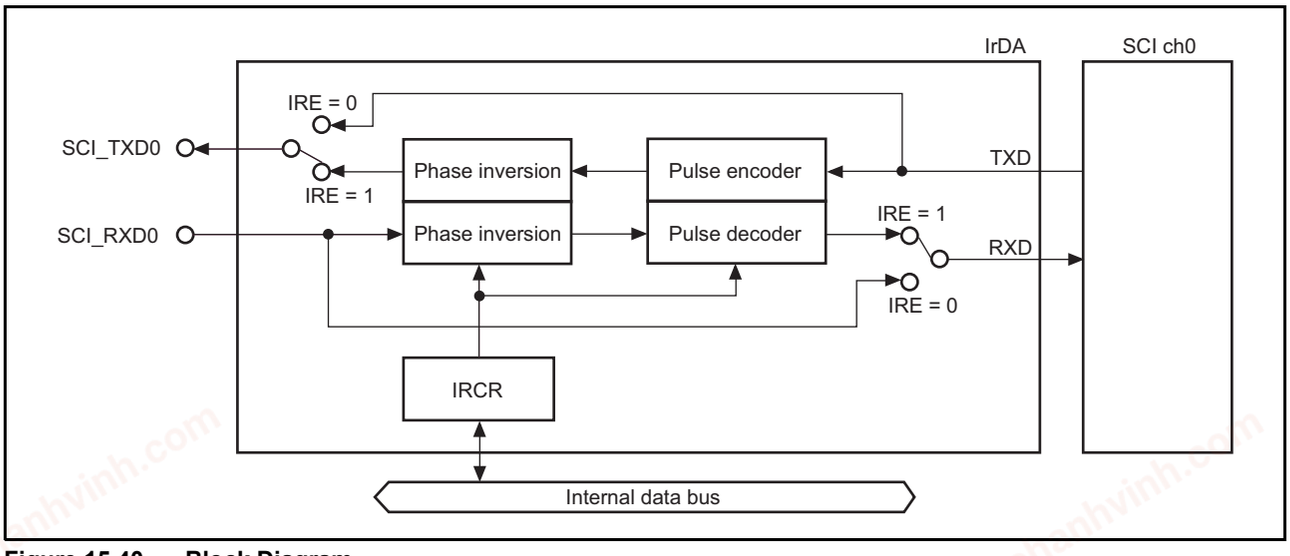


Figure 15.40 Block Diagram

Table 15.18 Pin Configuration

Name	Pin Name	I/O	Description
IrDA transmit data pin	SCI_TXD0	O	IrDA transmit data output
IrDA receive data pin	SCI_RXD0	I	IrDA receive data input

chanhvinh.com

15.11 IrDA Register Description

Table 15.19 shows the register configuration.

Table 15.19 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
IrDA control register	IRCR	R/W	H'00	H'E8014000	8

15.11.1 IrDA Control Register (IRCR)

IRCR is the register which sets the operation of the IrDA module.

Bit:	7	6	5	4	3	2	1	0
	IRE	IRCKS[2:0]			IRTXINV	IRRXINV	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	IRE	0	R/W	IrDA Enable Sets whether the SCI_TXD0 and SCI_RXD0 pins have normal serial function or IrDA function. 0: IrDA function is disabled (The TXD signal transmitted on channel 0 is directly output to the SCI_TXD0 pin) The RXD signal received on channel 0 is directly input to the SCI_RXD0 pin) 1: IrDA function is enabled (The TXD signal transmitted on channel 0 is output to the SCI_TXD0 pin after encoded) The RXD signal received on channel 0 is input to the SCI_RXD0 pin after decoded)
6 to 4	IRCKS[2:0]	000	R/W	IrDA Clock Select Sets the pulse width when the SCI_TXD0 output pulse is encoded with the IRE bit set to 1. 000: B × 3/16 (B = bit rate) 001: P1ϕ/2 010: P1ϕ/4 011: P1ϕ/8 100: P1ϕ/16 101: P1ϕ/32 110: P1ϕ/64 111: P1ϕ/128
3	IRTXINV	0	R/W	SCI_TXD0 Data Sense Switch Sets whether or not to invert the logic levels for SCI_TXD0 output. 0: Data for transmission are directly output to SCI_TXD0. The pulse width set by the IRCKS bits is the width at the high level. 1: Data for transmission are output to SCI_TXD0 after inversion. The pulse width set by the IRCKS bits is the width at the low level.
2	IRRXINV	0	R/W	SCI_RXD0 Data Sense Switch Sets whether or not to invert the logic levels of SCI_RXD0 input. 0: SCI_RXD0 input is directly used as receive data. 1: SCI_RXD0 input is used as receive data after inversion.
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

15.12 IrDA Operation

15.12.1 Flow of IrDA Setting

To set the operation of the IrDA module, follow the procedure below.

1. Make the general-purpose I/O port settings.
2. Set the IRCR register.
3. Set the relevant registers of the SCI.

15.12.2 Transmission

In transmission with the IrDA function enabled, serial data from the TXD pin for the SCI (UART frame data) are converted to IR frames (see Figure 15.41). When the IRTXINV bit is 0 and the value of the serial data is 0, a high-level pulse is output to the SCI_TXD0 pin for three sixteenths of one-bit period (initial value). The width of the high-level pulse can be changed by the IRCKS[2:0] bits in the IRCR register. The IrDA standard stipulates that the high-level pulse width is at least 1.41 µs and no greater than ((3/16 + 2.5%) × bit rate) or ((3/16 × bit rate) + 1.08) µs. When P1ϕ is 66.67 MHz, the width of the high-level pulse can be set from 1.41 µs to 1.92 µs. When the value of a bit of the serial data is 1, no pulse is output.

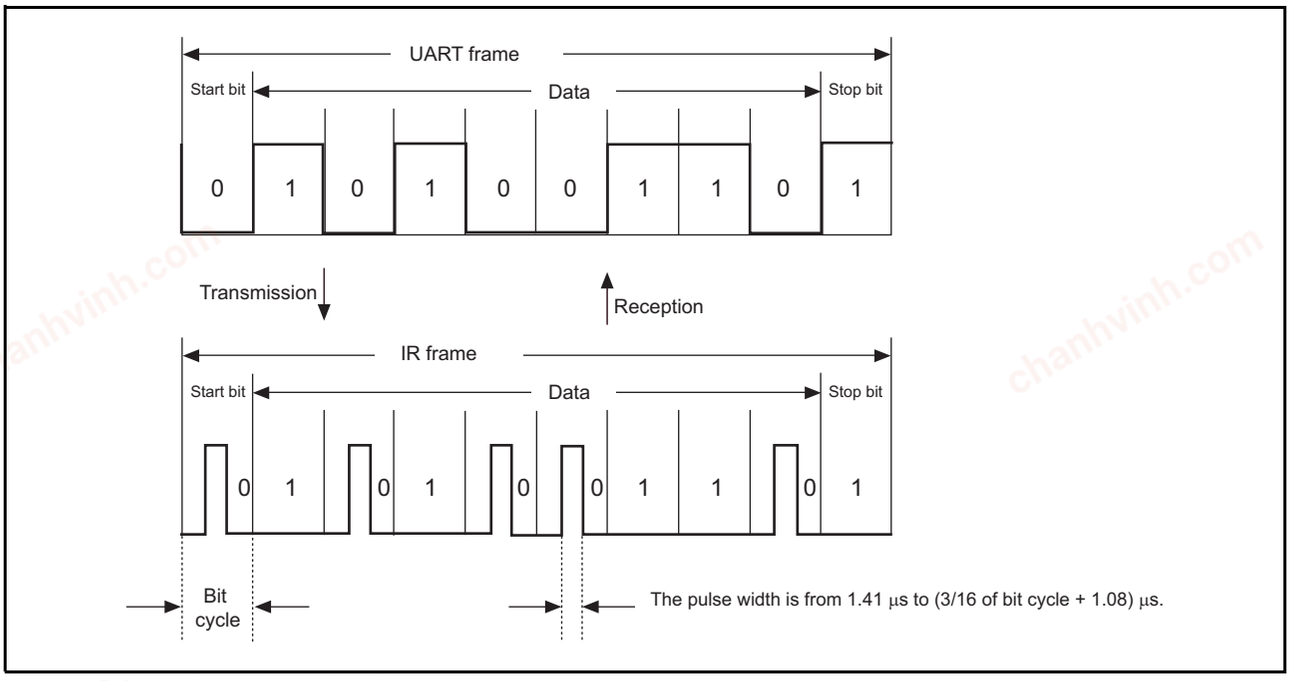


Figure 15.41 Example of IrDA Transmission and Reception

chanhvinh.com

15.12.3 Reception

In reception with the IRDA function enabled, IR frame data from the SCI_RXD0 pin are converted to serial data and output to the RXD pin for the SCI. When the IRRXINV bit is 0, a bit with the value 0 is output on the detection of a high-level pulse. When no pulse is received during one-bit period, a bit with the value 1 is output. Pulses shorter than the lower limit (1.41 µs) are not recognized.

15.12.4 Selection of High-Level Pulse Width

Table 15.20 shows the correspondence between the applicable IRCKS[2:0] bit setting (shortest pulse width), operating frequency P1ϕ, and bit rate when the pulse width is shortened below (bit rate × 3/16) in transmission.

Table 15.20 IRCKS[2:0] Bit Setting

	Bit Rate (bps) (Upper Row) / Bit Cycle × 3/16 (µs) (Lower Row)					
	2400	**9600**	**19200**	**38400**	**57600**	**115200**
P1ϕ (MHz)	**78.13**	**19.53**	**9.77**	**4.88**	**3.26**	**1.63**
50	111	111	111	111	111	—*1
64	111	111	111	111	111	—*2
66.67	111	111	111	111	111	—*2

Note 1. The bit rate cannot be set at the SCI.
Note 2. A pulse width shorter than (bit rate × 3/16) cannot be set.

chanhvinh.com

15.13 Notes on Using the IrDA Module

15.13.1 Shortest Pulse Width in Reception

Pulses shorter than the lower limit (1.41 µs) are not recognized.

15.13.2 Asynchronous Basic Clock for Serial Communication Interface

The IrDA module receives the basic clock with a frequency which is 16 times as high as the communication bit rate from the SCI and operates in combination with the clock. Although one-bit period of the SCI can be set to 16 or 8 clock cycles, the IrDA module supports the SCI that one-bit period is set to 16 clock cycles.

chanhvinh.com

16. Renesas Serial Peripheral Interface

This LSI circuit includes three independent Renesas serial peripheral interfaces.
This module is capable of full-duplex synchronous serial communication.

16.1 Features

This module has the following features.

- SPI transfer functions
  Use of MOSI (master out/slave in), MISO (master in/slave out), SSL (slave select), and RSPCK (SPI clock) signals allows for serial communications through SPI operation (four-wire method).
  Capable of serial communications in master/slave mode
  Supports mode fault error detection (only in SPI slave mode)
  Supports overrun error detection (only in SPI slave mode)
  Switching of the polarity of the serial transfer clock
  Switching of the clock phase of serial transfer
- Data format
  MSB-first/LSB-first selectable
  Transfer bit-length is selectable as 8, 16, or 32 bits.
- Bit rate
  RSPCK can be divided by a maximum of 4096 in master mode
  RSPCK can be generated by dividing P1ϕ by the on-chip baud rate generator.
  An externally input clock can be used as a serial clock.
- Buffer configuration
  8 bytes for transmission and 32 bytes for reception
- SSL control function
  One SSL signal for each channel
  In master mode, outputs SSL signal.
  In slave mode, inputs SSL signal.
  Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay)
  Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units)
  Controllable delay from RSPCK stoppage to SSL output negation (SSL negation delay)
  Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units)
  Controllable wait for next-access SSL output assertion (next-access delay)
  Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units)
  Function for changing SSL polarity
- Control in master transfer
  A transfer of up to four commands can be executed sequentially in looped execution.
  For each command, the following can be set:
  SSL signal value, bit rate, RSPCK polarity/phase, transfer data length, LSB/MSB first, burst, RSPCK delay, SSL negation delay, and next-access delay.
  A transfer can be initiated by writing to the transmit buffer.
  A transfer can be initiated by clearing the SPTEF bit.
  MOSI signal value specifiable in SSL negation
- Interrupt sources
  Maskable interrupt sources:
  Receive interrupt (receive buffer full)
  Transmit interrupt (transmit buffer empty)
  Error interrupt (mode fault, overrun)

chanhvinh.com

- Others
  Provides loop back mode
  Provides a function for disabling (initializing) this module

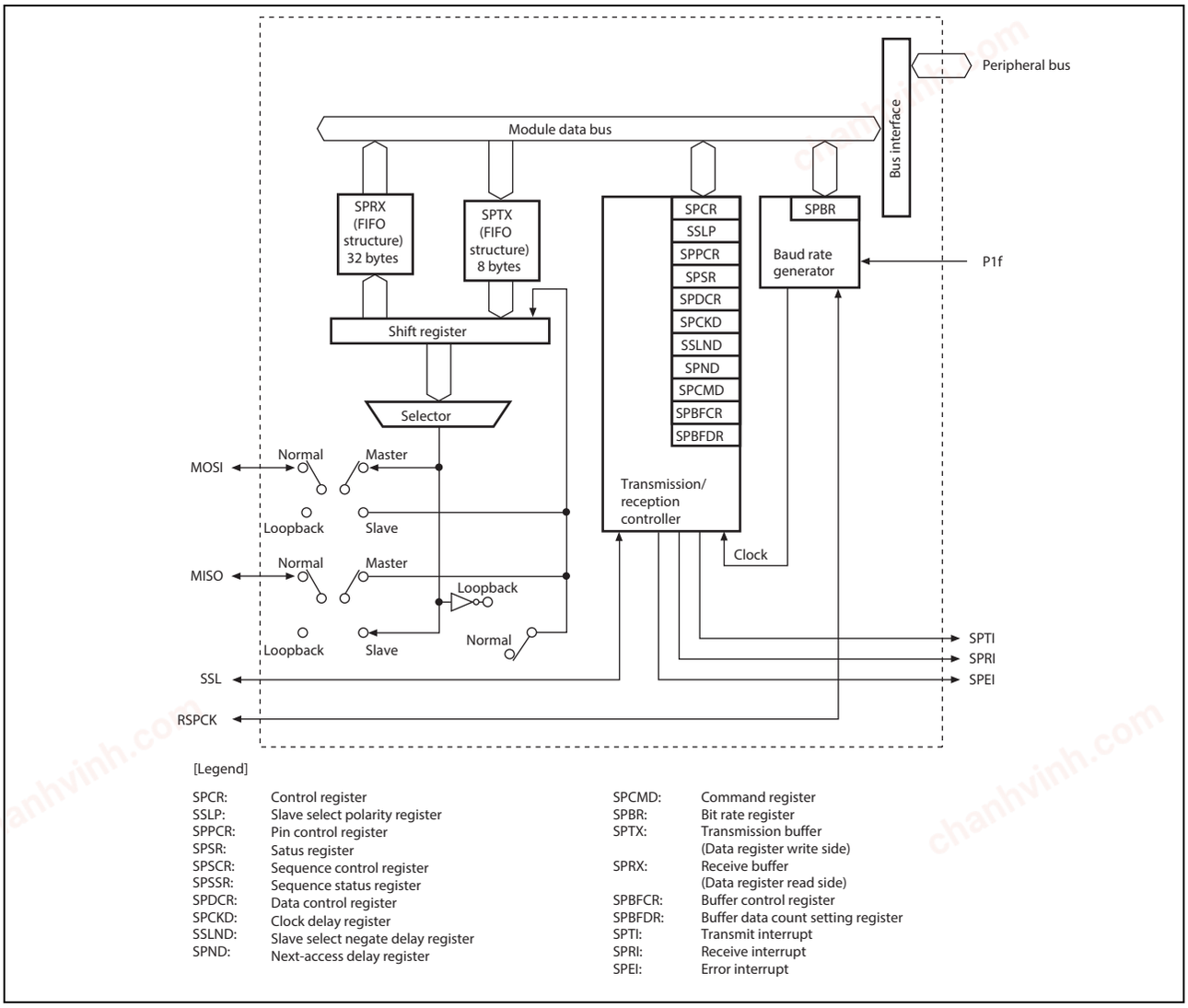


Figure 16.1 Block Diagram (for One Channel)

chanhvinh.com

16.2 Input/Output Pins

Table 16.1 shows the pin configuration. This module automatically switches the input/output direction of the SSL pin. SSL is set as an output in master mode and as an input in slave mode. Pins RSPCK, MOSI, and MISO are automatically set as inputs or outputs according to the setting of master or slave and the level input on SSL (see section 16.4.2, Pin Control).

Table 16.1 Pin Configuration

Channel	Pin Name	Pin Name	I/O	Function
0	Clock pin	RSPCK0	I/O	Clock input/output
	Master transmit data pin	MOSI0	I/O	Master transmit data
	Slave transmit data pin	MISO0	I/O	Slave transmit data
	Slave select 0 pin	SSL00	I/O	Slave selection
1	Clock pin	RSPCK1	I/O	Clock input/output
	Master transmit data pin	MOSI1	I/O	Master transmit data
	Slave transmit data pin	MISO1	I/O	Slave transmit data
	Slave select 0 pin	SSL10	I/O	Slave selection
2	Clock pin	RSPCK2	I/O	Clock input/output
	Master transmit data pin	MOSI2	I/O	Master transmit data
	Slave transmit data pin	MISO2	I/O	Slave transmit data
	Slave select 0 pin	SSL20	I/O	Slave selection

Note: • In the description of the pins, the channel is omitted and pin names are described as RSPCK, MOSI, MISO, and SSL.

chanhvinh.com

16.3 Register Descriptions

Table 16.2 shows the register configuration. These registers enable this module to perform the following controls: specifying master/slave modes, specifying a transfer format, and controlling the transmitter and receiver.

Table 16.2 Register Configuration

Channel	Register Name	Abbreviation*1	R/W	Initial Value	Address	Access Size
0	Control register_0	SPCR_0	R/W	H'00	H'E800C800	8
	Slave select polarity register_0	SSLP_0	R/W	H'00	H'E800C801	8
	Pin control register_0	SPPCR_0	R/W	H'00	H'E800C802	8
	Status register_0	SPSR_0	R/(W)*2	H'60	H'E800C803	8
	Data register_0	SPDR_0	R/W	Undefined	H'E800C804	8, 16, 32
	Sequence control register_0	SPSCR_0	R/W	H'00	H'E800C808	8
	Sequence status register_0	SPSSR_0	R	H'00	H'E800C809	8
	Bit rate register_0	SPBR_0	R/W	H'FF	H'E800C80A	8
	Data control register_0	SPDCR_0	R/W	H'20	H'E800C80B	8
	Clock delay register_0	SPCKD_0	R/W	H'00	H'E800C80C	8
	Slave select negation delay register_0	SSLND_0	R/W	H'00	H'E800C80D	8
	Next-access delay register_0	SPND_0	R/W	H'00	H'E800C80E	8
	Command register0_0	SPCMD0_0	R/W	H'070D	H'E800C810	16
	Command register1_0	SPCMD1_0	R/W	H'070D	H'E800C812	16
	Command register2_0	SPCMD2_0	R/W	H'070D	H'E800C814	16
	Command register3_0	SPCMD3_0	R/W	H'070D	H'E800C816	16
	Buffer control register_0	SPBFCR_0	R/W	H'00	H'E800C820	8
	Buffer data count setting register_0	SPBFDR_0	R	H'0000	H'E800C822	16
1	Control register_1	SPCR_1	R/W	H'00	H'E800D000	8
	Slave select polarity register_1	SSLP_1	R/W	H'00	H'E800D001	8
	Pin control register_1	SPPCR_1	R/W	H'00	H'E800D002	8
	Status register_1	SPSR_1	R/(W)*2	H'60	H'E800D003	8
	Data register_1	SPDR_1	R/W	Undefined	H'E800D004	8, 16, 32
	Sequence control register_1	SPSCR_1	R/W	H'00	H'E800D008	8
	Sequence status register_1	SPSSR_1	R	H'00	H'E800D009	8
	Bit rate register_1	SPBR_1	R/W	H'FF	H'E800D00A	8
	Data control register_1	SPDCR_1	R/W	H'20	H'E800D00B	8
	Clock delay register_1	SPCKD_1	R/W	H'00	H'E800D00C	8
	Slave select negation delay register_1	SSLND_1	R/W	H'00	H'E800D00D	8
	Next-access delay register_1	SPND_1	R/W	H'00	H'E800D00E	8
	Command register0_1	SPCMD0_1	R/W	H'070D	H'E800D010	16
	Command register1_1	SPCMD1_1	R/W	H'070D	H'E800D012	16
	Command register2_1	SPCMD2_1	R/W	H'070D	H'E800D014	16
	Command register3_1	SPCMD3_1	R/W	H'070D	H'E800D016	16
	Buffer control register_1	SPBFCR_1	R/W	H'00	H'E800D020	8
	Buffer data count setting register_1	SPBFDR_1	R	H'0000	H'E800D022	16

chanhvinh.com

Table 16.2 Register Configuration

Channel	Register Name	Abbreviation*1	R/W	Initial Value	Address	Access Size
2	Control register_2	SPCR_2	R/W	H'00	H'E800D800	8
	Slave select polarity register_2	SSLP_2	R/W	H'00	H'E800D801	8
	Pin control register_2	SPPCR_2	R/W	H'00	H'E800D802	8
	Status register_2	SPSR_2	R/(W) *2	H'60	H'E800D803	8
	Data register_2	SPDR_2	R/W	Undefined	H'E800D804	8, 16, 32
	Sequence control register_2	SPSCR_2	R/W	H'00	H'E800D808	8
	Sequence status register_2	SPSSR_2	R	H'00	H'E800D809	8
	Bit rate register_2	SPBR_2	R/W	H'FF	H'E800D80A	8
	Data control register_2	SPDCR_2	R/W	H'20	H'E800D80B	8
	Clock delay register_2	SPCKD_2	R/W	H'00	H'E800D80C	8
	Slave select negation delay register_2	SSLND_2	R/W	H'00	H'E800D80D	8
	Next-access delay register_2	SPND_2	R/W	H'00	H'E800D80E	8
	Command register0_2	SPCMD0_2	R/W	H'070D	H'E800D810	16
	Command register1_2	SPCMD1_2	R/W	H'070D	H'E800D812	16
	Command register2_2	SPCMD2_2	R/W	H'070D	H'E800D814	16
	Command register3_2	SPCMD3_2	R/W	H'070D	H'E800D816	16
	Buffer control register_2	SPBFCR_2	R/W	H'00	H'E800D820	8
	Buffer data count setting register_2	SPBFDR_2	R	H'0000	H'E800D822	16

Note 1. In the description of the register names, the channel is omitted.
Note 2. Only 0 can be written to clear the flag.

chanhvinh.com

16.3.1 Control Register (SPCR)

SPCR sets the operating mode. If the MSTR and MODFEN bits are changed while the function of this module is enabled by setting the SPE bit to 1, subsequent operations cannot be guaranteed.

Bit:	7	6	5	4	3	2	1	0
	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Function
7	SPRIE	0	R/W	Receive Interrupt Enable Enables or disables generation of receive interrupt requests (SPRI) when the number of receive data units in the receive buffer (SPRX) is equal to or greater than the specified receive buffer data triggering number and the SPRF flag in SPSR is set to 1. 0: Disables the generation of receive interrupt requests. 1: Enables the generation of receive interrupt requests.
6	SPE	0	R/W	Function Enable Setting this bit to 1 enables the module function. When the MODF bit in the status register (SPSR) is 1, the SPE bit cannot be set to 1 (see section 16.4.6, Error Detection). Setting the SPE bit to 0 disables the module function, and initializes a part of the module function (see section 16.4.7, Initialization). 0: Disables the module function. 1: Enables the module function.
5	SPTIE	0	R/W	Transmit Interrupt Enable Enables or disables generation of transmit interrupt requests (SPTI) when the number of transmit data units in the transmit buffer (SPTX) is equal to or less than the specified transmit buffer data triggering number and the SPTEF flag in SPSR is set to 1. 0: Disables the generation of transmit interrupt requests. 1: Enables the generation of transmit interrupt requests.
4	SPEIE	0	R/W	Error Interrupt Enable Enables or disables the generation of error interrupt requests when this module detects a mode fault error and sets the MODF bit in the status register (SPSR) to 1, or when this module detects an overrun error and sets the OVRF bit in SPSR to 1 (see section 16.4.6, Error Detection). 0: Disables the generation of error interrupt requests. 1: Enables the generation of error interrupt requests. Note: This bit is valid only in SPI slave mode.
3	MSTR	0	R/W	Master/Slave Mode Select Selects master/slave mode. According to MSTR bit settings, this module determines the direction of the RSPCK, MOSI, MISO, and SSL pins. 0: Slave mode 1: Master mode
2	MODFEN	0	R/W	Mode Fault Error Detection Enable Enables or disables the detection of a mode fault error (see section 16.4.6, Error Detection). 0: Disables the detection of a mode fault error. 1: Enables the detection of a mode fault error. Note: This bit is valid only in SPI slave mode. When master mode is specified with the MSTR bit, this bit should always be cleared to 0.
1, 0	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.

chanhvinh.com

16.3.2 Slave Select Polarity Register (SSLP)

SSLP sets the polarity of the SSL signal. If the contents of SSL0P are changed while the function of this module is enabled by setting the SPE bit in the control register (SPCR) to 1, subsequent operations cannot be guaranteed.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SSL0P
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Function
7 to 1	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
0	SSL0P	0	R/W	SSL Signal Polarity Setting Sets the polarity of the SSL signal. The value of SSL0P indicates the active polarity of the SSL signal. 0: SSL signal 0-active 1: SSL signal 1-active

16.3.3 Pin Control Register (SPPCR)

SPPCR sets the modes of the pins. If the contents of this register are changed while the function of this module is enabled by setting the SPE bit in the control register (SPCR) to 1, subsequent operations cannot be guaranteed.

Bit:	7	6	5	4	3	2	1	0
	—	—	MOIFE	MOIFV	—	—	—	SPLP
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Function
7, 6	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
5	MOIFE	0	R/W	MOSI Idle Value Fixing Enable Fixes the MOSI output value when this module in master mode is in an SSL negation period (including the SSL retention period during a burst transfer). When MOIFE is 0, this module outputs the last output value from the previous serial transfer during the SSL negation period to the MOSI pin. (The value is undefined when CPHA is 0). When MOIFE is 1, this module outputs the fixed value set in the MOIFV bit to the MOSI pin. 0: MOSI output value equals the last output value from previous transfer. (The value is undefined when CPHA is 0). 1: MOSI output value equals the value set in the MOIFV bit.
4	MOIFV	0	R/W	MOSI Idle Fixed Value If the MOIFE bit is 1 in master mode, this module, according to MOIFV bit settings, determines the MOSI signal value during the SSL negation period (including the SSL retention period during a burst transfer). 0: MOSI Idle fixed value equals 0. 1: MOSI Idle fixed value equals 1.
3 to 1	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
0	SPLP	0	R/W	Loopback When the SPLP bit is set to 1, this module shuts off the path between the MISO pin and the shift register, and between the MOSI pin and the shift register, and connects (reverses) the input path and the output path for the shift register. 0: Normal mode 1: Loopback mode

chanhvinh.com

16.3.4 Status Register (SPSR)

SPSR indicates the operating status.

Bit:	7	6	5	4	3	2	1	0
	SPRF	TEND	SPTEF	—	—	MODF	—	OVRF
Initial value:	0	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R/(W)*	R	R/(W)*

Note: * Only 0 can be written to clear the flag after reading 1.

Bit	Bit Name	Initial Value	R/W	Function
7	SPRF	0	R	Receive Buffer Full Flag Indicates that the number of receive data units in the receive buffer (SPRX) is equal to or greater than the receive buffer data triggering number specified in the buffer control register (SPBFCR). 0: The number of receive data units in the receive buffer is less than the receive buffer data triggering number. 1: The number of receive data units in the receive buffer is equal to or greater than the receive buffer data triggering number. [Clearing conditions] • The receive buffer data is read until the number of data units in the receive buffer becomes less than the specified receive buffer data triggering number. • Receive buffer data reset is enabled. • Power-on reset [Setting condition] • The number of data units in the receive buffer is equal to or greater than the specified receive buffer data triggering number.
6	TEND	1	R	Transmit End This bit is set to 1 when transmission is completed, and this bit is 0 when transmission is not completed. [Clearing condition] • When transmit data are transferred from the transmit register to the shift register. [Setting condition] • When the number of data units in the transmit buffer (SPTX) is zero when a serial transfer is completed. Note: This bit is valid only in SPI master mode.
5	SPTEF	1	R	Transmit Buffer Empty Flag Indicates that the number of transmit data units in the transmit buffer (SPTX) is equal to or less than the transmit buffer data triggering number specified in the buffer control register (SPBFCR). 0: The number of transmit data units in the transmit buffer is equal to or greater than the specified transmit buffer data triggering number. 1: The number of transmit data units in the transmit buffer is less than the specified transmit buffer data triggering number. [Clearing condition] • When data is written to the transmit buffer until the number of transmit data units in the transmit buffer exceeds the specified transmit buffer data triggering number. [Setting conditions] • When the number of transmit data units in the transmit buffer is less than the specified transmit buffer data triggering number. • When transmit buffer data reset is enabled. • Power-on reset
4, 3	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Function
2	MODF	0	R/(W)*1	Mode Fault Error Flag Indicates the occurrence of a mode fault error. If the MODFEN bit is set to 1 when this module is in slave mode and the SSL pin is negated before the RSPCK cycle necessary for data transfer ends, this module detects a mode fault error. The active level of the SSL signal is determined by the SSL0P bit in the slave select polarity register (SSLP). [Clearing conditions] • SPSR is read when the MODF bit is 1, and then 0 is written to the MODF bit. • Power-on reset 0: No mode fault error occurred 1: A mode fault error occurred Note: This bit is valid only in SPI slave mode.
1	—	0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
0	OVRF	0	R/(W)*1	Overrun Error Flag Indicates the occurrence of an overrun error. If a serial transfer ends when there is not enough space for receiving the specified length of data in the receive buffer (SPRX), this module detects an overrun error, and sets the OVRF bit to 1. [Clearing conditions] • SPSR is read when the OVRF bit is 1, and then 0 is written to the OVRF bit. • Power-on reset 0: No overrun error occurred 1: An overrun error occurred Note: This bit is valid only in SPI slave mode.

Note 1. Only 0 can be written to clear the flag after reading 1.

chanhvinh.com

16.3.5 Data Register (SPDR)

SPDR is a buffer that holds data for transmission and reception.
The transmit buffer (SPTX) and receive buffer (SPRX) are independent and are mapped to SPDR.
SPDR should be read or written to in byte, word, or longword units according to the access width specification bit (SPLW) in the data control register (SPDCR).
The bit length to be used is determined by the data length specification bits (SPB3 to SPB0) in the command register (SPCMD).
The access width set by SPDCR must agree with the data length set by SPCMD.
When data is written to SPDR, the data will be written to the transmit buffer from SPDR if the transmit buffer has a space equal to or more than the SPDR access width. If there is not enough space, data will not be written to the transmit buffer. Even if an attempt is made to write data to the buffer, the data is ignored.
When data is read from SPDR, receive data in the receive buffer will be read. If SPDR is read when there is no receive data in the receive buffer, the read value is undefined.
When SPDR is written to with the longword-, word-, or byte-access width, the transmit data should be written to address 0 irrespective of the access width. If data is written to the other addresses, the data is not guaranteed.
When SPDR is read with the longword-, word-, or byte-access width, the receive data should be read from address 0. If data is read from the other addresses, the data is not guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPD31	SPD30	SPD29	SPD28	SPD27	SPD26	SPD25	SPD24	SPD23	SPD22	SPD21	SPD20	SPD19	SPD18	SPD17	SPD16
Initial value:	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPD15	SPD14	SPD13	SPD12	SPD11	SPD10	SPD9	SPD8	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
Initial value:	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

16.3.6 Sequence Control Register (SPSCR)

SPSCR sets the sequence control method when this module operates in master mode. If the contents of SPSCR are changed while the MSTR and SPE bits in the control register (SPCR) are 1 with the function of this module enabled in master mode, the subsequent operation cannot be guaranteed.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SPSLN1	SPSLN0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Function
7 to 2	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
1 0	SPSLN1 SPSLN0	0 0	R/W R/W	Sequence Length Specification These bits specify a sequence length when this module in master mode performs sequential operations. This module in master mode changes command registers 0 to 3 (SPCMD0 to SPCMD3) to be referenced and the order in which they are referenced according to the sequence length that is set in the SPSLN1 and SPSLN0 bits. The relationship among the setting of bits SPSLN1 and SPSLN0, sequence length, and SPCMD0 to SPCMD3 referenced by this module is shown below. In slave mode, SPCMD0 is always referenced. Sequence Length / Referenced SPCMD # 00: 1 — 0 → 0 → … 01: 2 — 0 → 1 → 0 → … 10: 3 — 0 → 1 → 2 → 0 → … 11: 4 — 0 → 1 → 2 → 3 → 0 → …

chanhvinh.com

16.3.7 Sequence Status Register (SPSSR)

SPSSR indicates the sequence control status when this module operates in master mode.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SPCP1	SPCP0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Function
7 to 2	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
1 0	SPCP1 SPCP0	0 0	R R	Command Pointer During sequence control, these bits indicate one of the command registers 0 to 3 (SPCMD0 to SPCMD3) that is currently pointed to by the pointer. The relationship between the setting of SPCP1 and SPCP0 and SPCMD0 to SPCMD3 is shown below. For the sequence control, see section 16.4.8 (1) (c), Sequence Control. 00: SPCMD0 01: SPCMD1 10: SPCMD2 11: SPCMD3

16.3.8 Bit Rate Register (SPBR)

SPBR sets the bit rate in master mode. If the contents of SPBR are changed while the MSTR and SPE bits in the control register (SPCR) are 1 with the function of this module enabled in master mode, the subsequent operation cannot be guaranteed.

Bit:	7	6	5	4	3	2	1	0
	SPR7	SPR6	SPR5	SPR4	SPR3	SPR2	SPR1	SPR0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When this module is used in slave mode, the bit rate depends on the bit rate of the input clock regardless of the settings of SPBR and BRDV.

The bit rate is determined by combinations of SPBR settings and the bit settings in the BRDV1 and BRDV0 bits in the command registers (SPCMD0 to SPCMD3). The equation for calculating the bit rate is given below. In the equation, n denotes an SPBR setting (0, 1, 2, …, 255), and N denotes a BRDV1 and BRDV0 bit setting (0, 1, 2, 3).

$$\text{Bit rate} = \frac{f(P1\phi)}{2 \times (n + 1) \times 2^{N}}$$

Table 16.3 shows examples of the relationship between the SPBR register and BRDV1 and BRDV0 bit settings.

chanhvinh.com

Table 16.3 Relationship between SPBR and BRDV1 and BRDV0 Settings

			Bit Rate		
SPBR (n)	**BRDV[1:0] (N)**	**Division Ratio**	**Pφ = 50 MHz**	**Pφ = 64 MHz**	**Pφ = 66.67 MHz**
0	0	2*1	25.00 Mbps	32.00 Mbps	33.33 Mbps
1	0	4	12.50 Mbps	16.00 Mbps	16.67 Mbps
2	0	6	8.33 Mbps	10.67 Mbps	11.11 Mbps
3	0	8	6.25 Mbps	8.00 Mbps	8.33 Mbps
4	0	10	5.00 Mbps	6.40 Mbps	6.67 Mbps
5	0	12	4.17 Mbps	5.33 Mbps	5.56 Mbps
5	1	24	2.08 Mbps	2.67 Mbps	2.78 Mbps
5	2	48	1.04 Mbps	1.33 Mbps	1.39 Mbps
5	3	96	520.83 Kbps	666.67 Kbps	694.44 Kbps
255	3	4096	12.21 Kbps	15.63 Kbps	16.28 Kbps

Note 1. Examine the timing specifications to determine the bit rate in the actual system.

16.3.9 Data Control Register (SPDCR)

SPDCR selects the width to access SPDR from longword-, word-, and byte-width, and enables or disables dummy data transmission for the master mode operation.

If the contents of SPDCR are changed while bit TEND in the status register (SPSR) indicates that transmission is not completed, the subsequent operation cannot be guaranteed.

Bit:	7	6	5	4	3	2	1	0
	TXDMY	SPLW1	SPLW0	—	—	—	—	—
Initial value:	0	0	1	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Function
7	TXDMY	0	R/W	Dummy Data Transmission Enable Enables or disables dummy data transmission. When communication is performed with this bit set to 1, dummy data is transmitted from the MOSI pin and a serial communication can be performed even if there is no transmit data in the transmit buffer. Specifically, if there is no transmit data in the transmit buffer and this bit is set to 1, dummy data is transferred to the shift register. Data previously transmitted from the pin is used as dummy data. If this bit is set to 1 after the initialization and a transfer is performed, the transmitted dummy data is undefined. 0: Disables dummy data transmission. 1: Enables dummy data transmission. Note: This bit is valid only in the master mode.
6 5	SPLW1 SPLW0	0 1	R/W R/W	Access Width Specification Specifies the width for accessing the data register (SPDR). If the length of data transferred to SPDR does not agree with these bit settings, operation is not guaranteed.* 00: Setting prohibited 01: SPDR is accessed in bytes (8 bits). 10: SPDR is accessed in words (16 bits). 11: SPDR is accessed in longwords (32 bits).
4 to 0	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.

Note: * The data length is specified by the SPB3 to SPB0 bits in the command register (SPCMD).

See section 16.3.5, Data Register (SPDR).

chanhvinh.com

16.3.10 Clock Delay Register (SPCKD)

SPCKD sets a period from the beginning of SSL signal assertion to RSPCK oscillation (RSPCK delay) when the SCKDEN bit in the command register (SPCMD) is 1. If the contents of SPCKD are changed while the MSTR and SPE bits in the control register (SPCR) are 1 with the function of this module enabled in master mode, the subsequent operation cannot be guaranteed.

When using this module in slave mode, set B'000 to SCKDL2 to SCKDL0.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	SCKDL2	SCKDL1	SCKDL0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Function
7 to 3	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
2	SCKDL2	0	R/W	RSPCK Delay Setting These bits set an RSPCK delay value when the SCKDEN bit in SPCMD is 1. The relationship between the setting of SCKDL2 to SCKDL0 and the RSPCK delay value is shown below. 000: 1 RSPCK 001: 2 RSPCK 010: 3 RSPCK 011: 4 RSPCK 100: 5 RSPCK 101: 6 RSPCK 110: 7 RSPCK 111: 8 RSPCK
1	SCKDL1	0	R/W	
0	SCKDL0	0	R/W	

chanhvinh.com

16.3.11 Slave Select Negation Delay Register (SSLND)

SSLND sets a period (SSL negation delay) from the transmission of a final RSPCK edge to the negation of the SSL signal during a serial transfer by this module in master mode. If the contents of SSLND are changed while the MSTR and SPE bits in the control register (SPCR) are 1 with the function of this module enabled in master mode, the subsequent operation cannot be guaranteed.

When using this module in slave mode, set B'000 to SLNDL2 to SLNDL0.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	SLNDL2	SLNDL1	SLNDL0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Function
7 to 3	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
2	SLNDL2	0	R/W	SSL Negation Delay Setting These bits set an SSL negation delay when the SLNDEN bit in SPCMD is 1. The relationship between the setting of SLNDL2 to SLNDL0 and the SSL negation delay value is shown below. 000: 1 RSPCK 001: 2 RSPCK 010: 3 RSPCK 011: 4 RSPCK 100: 5 RSPCK 101: 6 RSPCK 110: 7 RSPCK 111: 8 RSPCK
1	SLNDL1	0	R/W	
0	SLNDL0	0	R/W	

chanhvinh.com

16.3.12 Next-Access Delay Register (SPND)

SPND sets a non-active period (next-access delay) after termination of a serial transfer when the SPNDEN bit in the command register (SPCMD) is 1. If the contents of SPND are changed while the MSTR and SPE bits in the control register (SPCR) are 1 with the function of this module enabled in master mode, the subsequent operation cannot be guaranteed.

When using this module in slave mode, set B'000 to SPNDL2 to SPNDL0.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	SPNDL2	SPNDL1	SPNDL0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Function
7 to 3	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
2	SPNDL2	0	R/W	Next-Access Delay Setting
1	SPNDL1	0	R/W	These bits set a next-access delay when the SPNDEN bit in SPCMD is 1.
0	SPNDL0	0	R/W	The relationship between the setting of SPNDL2 to SPNDL0 and the next-access delay value is shown below. 000: 1 RSPCK + 2 P1ϕ 001: 2 RSPCK + 2 P1ϕ 010: 3 RSPCK + 2 P1ϕ 011: 4 RSPCK + 2 P1ϕ 100: 5 RSPCK + 2 P1ϕ 101: 6 RSPCK + 2 P1ϕ 110: 7 RSPCK + 2 P1ϕ 111: 8 RSPCK + 2 P1ϕ

chanhvinh.com

16.3.13 Command Register (SPCMD)

Each channel has four command registers (SPCMD0 to SPCMD3). SPCMD0 to SPCMD3 are used to set a transfer format for master mode operation. Some of the bits in SPCMD0 are used to set a transfer mode for slave mode operation. In master mode, this module sequentially references SPCMD0 to SPCMD3 according to the settings in bits SPSLN1 and SPSLN0 in the sequence control register (SPSCR), and executes the serial transfer that is set in the referenced SPCMD. While bit TEND in the status register (SPSR) indicates that transmission is not completed, correct operation of this module cannot be guaranteed if SPCMD is changed that is referred by this module. SPCMD referenced by this module in master mode can be checked by means of bits SPCP1 and SPCP0 in the sequence status register (SPSSR). When the function of this module in slave mode is enabled, operation cannot be guaranteed if the value set in SPCMD0 is changed.

Bit:	15	14	13	12	11	10	9	8
	SCK DEN	SLN DEN	SPN DEN	LSBF	SPB3	SPB2	SPB1	SPB0
Initial value:	0	0	0	0	0	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	SSLKP	—	—	—	BRDV1	BRDV0	CPOL	CPHA
Initial value:	0	0	0	0	1	1	0	1
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Function
15	SCKDEN	0	R/W	RSPCK Delay Setting Enable Sets the period from the point this module in master mode activates the SSL signal until the RSPCK starts oscillation (RSPCK delay). If the SCKDEN bit is 0, this module sets the RSPCK delay to 1 RSPCK. If the SCKDEN bit is 1, this module starts the oscillation of RSPCK at an RSPCK delay in compliance with the clock delay register (SPCKD) settings. To use this module in slave mode, the SCKDEN bit should be set to 0. 0: An RSPCK delay of 1 RSPCK 1: An RSPCK delay equal to SPCKD settings.
14	SLNDEN	0	R/W	SSL Negation Delay Setting Enable Sets the period from the point this module in master mode stops RSPCK oscillation until this module sets the SSL signal inactive (SSL negation delay). If the SLNDEN bit is 0, this module sets the SSL negation delay to 1 RSPCK. If the SLNDEN bit is 1, this module negates the SSL signal at an SSL negation delay in compliance with the slave select negation delay register (SSLND) settings. To use this module in slave mode, the SLNDEN bit should be set to 0. 0: An SSL negation delay of 1 RSPCK 1: An SSL negation delay equal to SSLND settings.
13	SPNDEN	0	R/W	Next-Access Delay Enable Sets the period from the point this module in master mode terminates a serial transfer and sets the SSL signal inactive until this module enables the SSL signal assertion for the next access (next-access delay). If the SPNDEN bit is 0, this module sets the next-access delay to 1 RSPCK + 2P1ϕ. If the SPNDEN bit is 1, this module inserts a next-access delay in compliance with the next-access delay register (SPND) settings. To use this module in slave mode, the SPNDEN bit should be set to 0. 0: A next-access delay of 1 RSPCK + 2 P1ϕ 1: A next-access delay equal to SPND settings.
12	LSBF	0	R/W	LSB First Sets the data format in master mode or slave mode to MSB first or LSB first. 0: MSB first 1: LSB first

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Function
11 10 9 8	SPB3 SPB2 SPB1 SPB0	0 1 1 1	R/W R/W R/W R/W	Data Length Setting These bits set a transfer data length in master mode or slave mode. 0100 to 0111: 8 bits 1111: 16 bits 0010, 0011: 32 bits Others: Setting prohibited
7	SSLKP	0	R/W	SSL Signal Level Keeping When this module in master mode performs a serial transfer, this bit specifies whether the SSL signal level for the current command is to be kept or negated between the SSL negation timing associated with the current command and the SSL assertion timing associated with the next command. To use this module in slave mode, the SSLKP bit should be set to 0. 0: Negates the SSL signal upon completion of transfer. 1: Keeps the SSL signal level from the end of the transfer until the beginning of the next access.
6 to 4	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
3 2	BRDV1 BRDV0	1 1	R/W R/W	Bit Rate Division Setting These bits are used to determine the bit rate. A bit rate is determined by combinations of bits BRDV1 and BRDV 0 and the settings in the bit rate register (SPBR) (see section 16.3.8, Bit Rate Register (SPBR)). The settings in SPBR determine the base bit rate. The settings in bits BRDV1 and BRDV0 are used to select a bit rate which is obtained by dividing the base bit rate by 1, 2, 4, or 8. In the bits SPCMD0 to SPCMD3, different BRDV1 and BRDV0 settings can be specified. This permits the execution of serial transfers at a different bit rate for each command. 00: Select the base bit rate. 01: Select the base bit rate divided by 2. 10: Select the base bit rate divided by 4. 11: Select the base bit rate divided by 8.
1	CPOL	0	R/W	RSPCK Polarity Setting Sets an RSPCK polarity in master or slave mode. When data communication is performed between the Renesas serial peripheral interface module and the other modules, the same RSPCK polarity should be set for both modules. 0: RSPCK = 0 when idle 1: RSPCK = 1 when idle
0	CPHA	1	R/W	RSPCK Phase Setting Sets an RSPCK phase in master or slave mode. When data communication is performed between the Renesas serial peripheral interface module and the other modules, the same RSPCK phase should be set for both modules. 0: Data sampling on odd edge, data variation on even edge 1: Data variation on odd edge, data sampling on even edge

chanhvinh.com

16.3.14 Buffer Control Register (SPBFCR)

SPBFCR resets the number of data units in the transmit buffer (SPTX) or receive buffer (SPRX) and sets the number of triggering data units.

Bit:	7	6	5	4	3	2	1	0
	TXRST	RXRST	TXTRG[1:0]		—	RXTRG[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Function
7	TXRST	0	R/W	Transmit Buffer Data Reset Resets the transmit buffer to an empty state. Transmit data in the transmit buffer becomes invalid when this bit is set to 1. 0: Disables the reset operation*. 1: Enables the reset operation Note: The reset operation is performed after a power-on reset.
6	RXRST	0	R/W	Receive Buffer Data Reset Resets the receive buffer to an empty state. Receive data in the receive buffer becomes invalid when this bit is set to 1. 0: Disables the reset operation*. 1: Enables the reset operation Note: The reset operation is performed after a power-on reset.
5, 4	TXTRG[1:0]	00	R/W	Transmit Buffer Data Triggering Number Specifies the timing at which the transmit buffer empty state is determined, that is when the SPTEF flag in the status register is set. When the number of bytes of data in the transmit buffer (SPTX) is equal to or less than the specified triggering number, the SPTEF flag is set to 1. 00: 7 bytes (1)* 01: 6 bytes (2)* 10: 4 bytes (4)* 11: 0 bytes (8)* Note: The value in the parenthesis shows the number of available bytes in the transmit buffer (SPTX).
3	—	0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
2 to 0	RXTRG[2:0]	000	R/W	Receive Buffer Data Triggering Number Specifies the timing at which the receive buffer full state is determined, that is when the SPRF flag in the status register is set. When the number of bytes of data in the receive buffer (SPRX) is equal to or greater than the specified triggering number, the SPRF flag is set to 1. 000: 1 byte (31)* 001: 2 bytes (30)* 010: 4 bytes (28)* 011: 8 bytes (24)* 100: 16 bytes (16)* 101: 24 bytes (8)* 110: 32 bytes (0)* 111: 5 bytes (27)* Note: * The value in the parenthesis shows the number of available bytes in the receive buffer (SPRX).

chanhvinh.com

16.3.15 Buffer Data Count Setting Register (SPBFDR)

SPBFDR indicates the number of data units stored in the transmit buffer (SPTX) and receive buffer (SPRX). The upper eight bits indicate the number of transmit data units in SPTX and the lower eight bits indicate the number of receive data units in SPRX.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	T[3:0]			
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	R[5:0]					
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Function
15 to 12	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
11 to 8	T[3:0]	0000	R	Indicates the number of bytes of data to be transmitted in SPTX. B'0000 indicates that SPTX is empty. B'1000 indicates that SPTX is full.
7, 6	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
5 to 0	R[5:0]	000000	R	Shows the number of bytes of received data in SPTX. B'000000 indicates that SPRX is empty. B'100000 indicates that SPRX is full.

chanhvinh.com

16.4 Operation

In this section, the serial transfer period means a period from the beginning of driving valid data to the fetching of the final valid data.

16.4.1 Overview of Operations

This module is capable of serial transfers in slave mode and master mode. A particular mode of this module can be selected by using the MSTR bit in the control register (SPCR). Table 16.4 gives the relationship between the modes and SPCR settings, and a description of each mode.

Table 16.4 Relationship between Modes and SPCR and Description of Each Mode

Mode	Slave (SPI Operation)	Master (SPI Operation)
MSTR bit setting	0	1
MODFEN bit setting	0 or 1	0
RSPCK signal	Input	Output
MOSI signal	Input	Output
MISO signal	Output/Hi-Z	Input
SSL signal	Input	Output
SSL polarity modification function	Supported	Supported
Transfer rate	Up to P1ϕ/8	Up to P1ϕ/2
Clock source	RSPCK input	On-chip baud rate generator
Clock polarity	Two	Two
Clock phase	Two	Two
First transfer bit	MSB/LSB	MSB/LSB
Transfer data length	8, 16, or 32 bits	8, 16, or 32 bits
Burst transfer	Possible (CPHA = 1)	Possible (CPHA = 0,1)
RSPCK delay control	Not supported	Supported
SSL negation delay control	Not supported	Supported
Next-access delay control	Not supported	Supported
Transfer activation method	SSL input active or RSPCK oscillation	Transmit buffer is written when SPE = 1
Sequence control	Not supported	Supported
Transmit buffer empty detection	Supported	Supported
Receive buffer full detection	Supported	Supported
Overrun error detection	Supported	Not Supported
Mode fault error detection	Supported (MODFEN = 1)	Not supported

chanhvinh.com

16.4.2 Pin Control

According to the MSTR bit in the control register (SPCR), this module can automatically switch pin directions and output modes. Table 16.5 shows the relationship between pin states and bit settings.

Table 16.5 Relationship between Pin States and Bit Settings

Mode	Pin	Pin State*1
Master mode (SPI operation) (MSTR = 1)	RSPCK	CMOS output
	SSL	CMOS output
	MOSI	CMOS output
	MISO	Input
Slave mode (SPI operation) (MSTR = 0)	RSPCK	Input
	SSL	Input
	MOSI	Input
	MISO*1	CMOS output/Hi-Z

Note 1. When SSL is at the non-active level or the SPE bit in SPCR is cleared to 0, the pin state is Hi-Z.

This module in master mode (SPI operation) determines MOSI signal values during the SSL negation period (including the SSL retention period during a burst transfer) according to MOIFE and MOIFV bit settings in SPPCR, as shown in Table 16.6.

Table 16.6 MOSI Signal Value Determination during SSL Negation Period

MOIFE	MOIFV	MOSI Signal Value during SSL Negation Period
0	0, 1	Last output value from previous transfer (The value is undefined when CPHA is 0)
1	0	Always 0
1	1	Always 1

16.4.3 System Configuration Example

(1) Master/Slave (with This LSI Acting as Master)

Figure 16.2 shows a master/slave system configuration example when this LSI is used as a master. In master/slave configuration, the SSL output of this LSI (master) is not used. The SSL input of the slave is fixed to the low level, and the slave is always maintained in a selected state. In the transfer format corresponding to the case where the CPHA bit in the control register (SPCR) is 0, there are slave devices for which the SSL signal cannot be fixed to the active level. In situations where the SSL signal cannot be fixed, the SSL output of this LSI should be connected to the SSL input of the slave device.

This LSI (master) always drives the RSPCK and MOSI. The slave always drives the MISO.

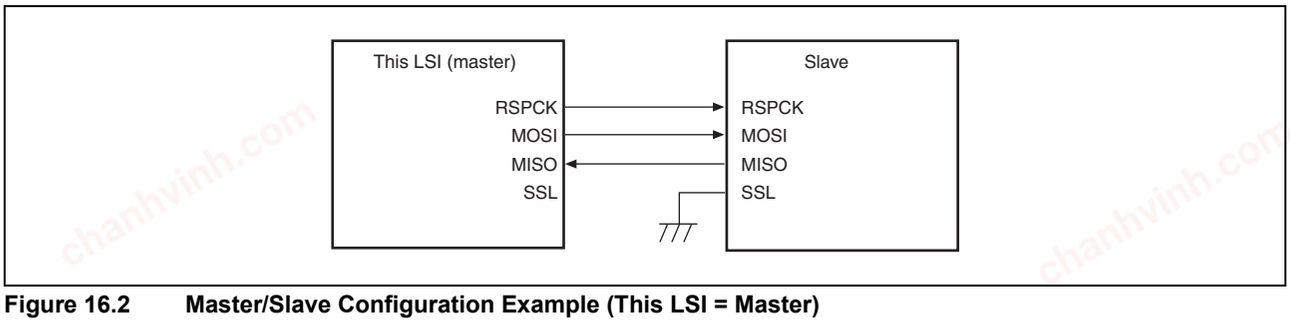


Figure 16.2 Master/Slave Configuration Example (This LSI = Master)

chanhvinh.com

(2) Master/Slave (with This LSI Acting as Slave)

Figure 16.3 shows a master/slave system configuration example when this LSI is used as a slave. When this LSI is to operate as a slave, the SSL pin is used as SSL input. The master always drives the RSPCK and MOSI. This LSI (slave) always drives the MISO. When SSL is at the non-active level, the pin state is Hi-Z.

In the slave configuration in which the CPHA bit in the command register (SPCMD) is set to 1, the SSL input of this LSI (slave) is fixed to the 0 level, this LSI (slave) is always maintained in a selected state, and in this manner it is possible to execute serial transfer (Figure 16.4).

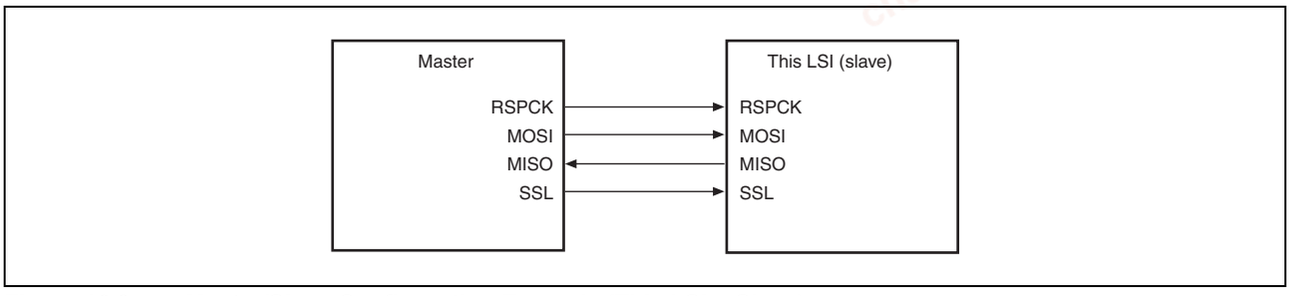


Figure 16.3 Master/Slave Configuration Example (This LSI = Slave)

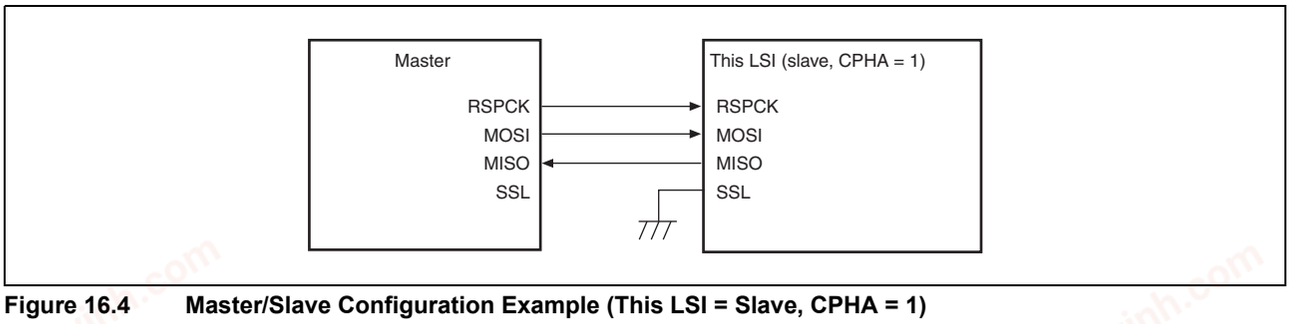


Figure 16.4 Master/Slave Configuration Example (This LSI = Slave, CPHA = 1)

chanhvinh.com

(3) Master/Multi-Slave (with This LSI Acting as Slave)

Figure 16.5 shows a master/multi-slave system configuration example when this LSI is used as a slave. In the example of Figure 16.5, the system is comprised of a master and two LSIs (slave X and slave Y).
The RSPCK and MOSI outputs of the master are connected to the RSPCK and MOSI inputs of the LSIs (slave X and slave Y). The MISO outputs of the LSIs (slave X and slave Y) are all connected to the MISO input of the master. SSLX and SSLY outputs of the master are connected to the SSL inputs of the LSIs (slave X and slave Y), respectively.
The master always drives RSPCK, MOSI, SSLX, and SSLY. Of the LSIs (slave X and slave Y), the slave that receives low level input into the SSL0 input drives MISO.

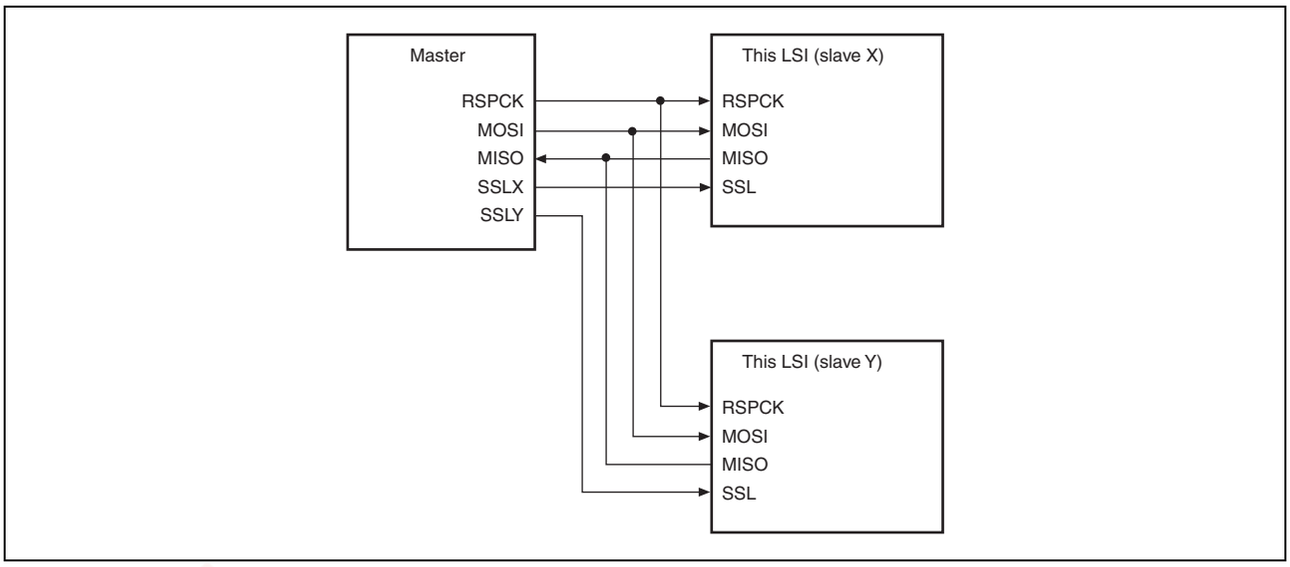


Figure 16.5 Master/Multi-Slave Configuration Example (This LSI = Slave)

chanhvinh.com

16.4.4 Transfer Format

(1) CPHA = 0

Figure 16.6 shows a sample transfer format for the serial transfer of 8-bit data when the CPHA bit in the command register (SPCMD) is 0. In Figure 16.6, RSPCK (CPOL = 0) indicates the RSPCK signal waveform when the CPOL bit in SPCMD is 0; RSPCK (CPOL = 1) indicates the RSPCK signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which this module fetches serial transfer data into the shift register. The input/output directions of the signals depend on the settings of this module. For details, see section 16.4.2, Pin Control.

When the CPHA bit is 0, the driving of valid data to the MOSI and MISO signals commences at an SSL signal assertion timing. The first RSPCK signal change timing that occurs after the SSL signal assertion becomes the first transfer data fetching timing. After this timing, data is sampled at every 1 RSPCK cycle. The change timing for MOSI and MISO signals is always 1/2 RSPCK cycle after the transfer data fetch timing. The settings in the CPOL bit do not affect the RSPCK signal operation timing; they only affect the signal polarity.

t1 denotes a period from an SSL signal assertion to RSPCK oscillation (RSPCK delay). t2 denotes a period from the cessation of RSPCK oscillation to an SSL signal negation (SSL negation delay). t3 denotes a period in which SSL signal assertion is suppressed for the next transfer after the end of serial transfer (next-access delay). t1, t2, and t3 are controlled by a master device running on the system. For a description of t1, t2, and t3 when this module is in master mode, see section 16.4.3 (1), Master/Slave (with This LSI Acting as Master).

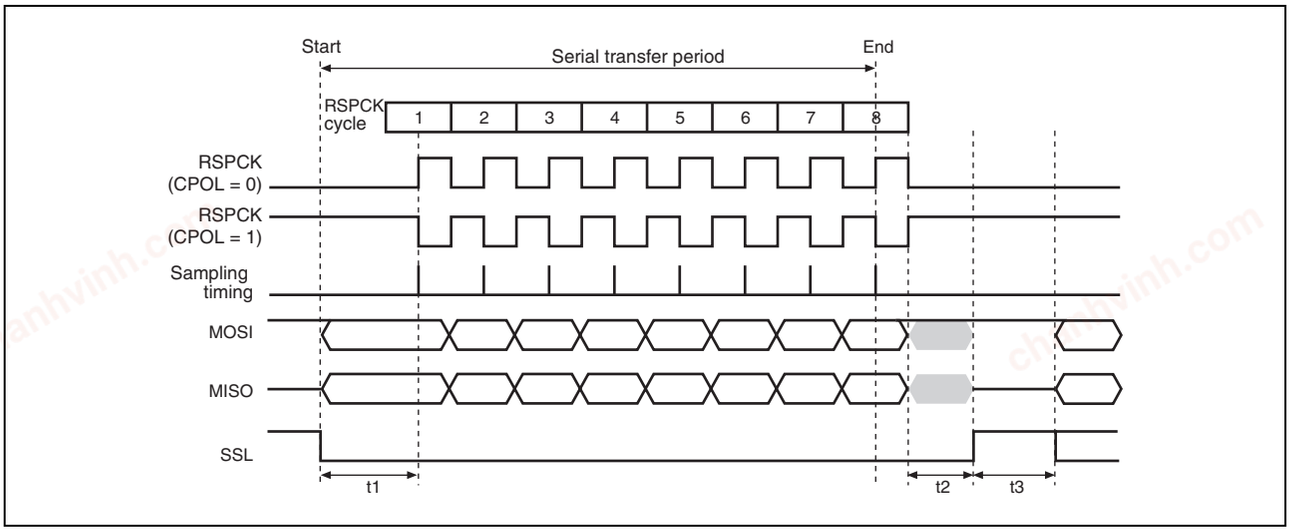


Figure 16.6 Transfer Format (CPHA = 0)

chanhvinh.com

(2) CPHA = 1

Figure 16.7 shows a sample transfer format for the serial transfer of 8-bit data when the CPHA bit in the command register (SPCMD) is 1. In Figure 16.7, RSPCK (CPOL = 0) indicates the RSPCK signal waveform when the CPOL bit in SPCMD is 0; RSPCK (CPOL = 1) indicates the RSPCK signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which this module fetches serial transfer data into the shift register. The input/output directions of the signals depend on the modes (master or slave). For details, see section 16.4.2, Pin Control.

When the CPHA bit is 1, the driving of invalid data to the MOSI and MISO signals commences at an SSL signal assertion timing. The driving of valid data to the MOSI and MISO signals commences at the first RSPCK signal change timing that occurs after the SSL signal assertion. After this timing, data is updated at every 1 RSPCK cycle. The transfer data fetch timing is always 1/2 RSPCK cycle after the data update timing. The settings in the CPOL bit do not affect the RSPCK signal operation timing; they only affect the signal polarity.

t1, t2, and t3 are the same as those in the case of CPHA = 0. For a description of t1, t2, and t3 when this module is in master mode, see section 16.4.3 (1), Master/Slave (with This LSI Acting as Master).

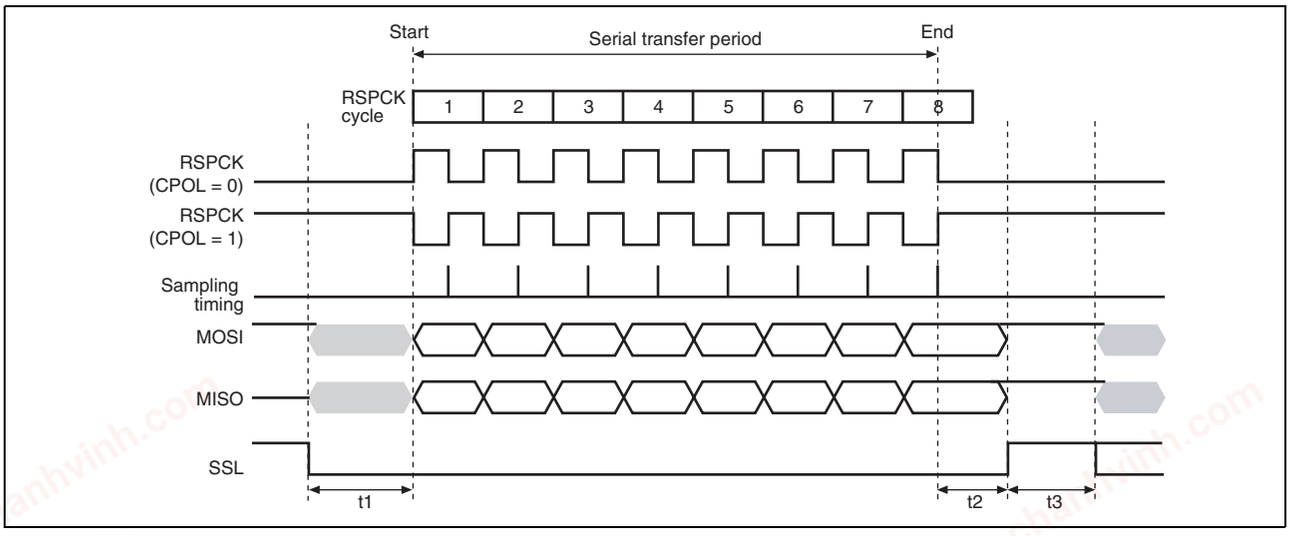


Figure 16.7 Transfer Format (CPHA = 1)

chanhvinh.com

16.4.5 Data Format

The data format depends on the settings in the command register (SPCMD). Irrespective of MSB/LSB first, this module treats the range from the LSB of the data register (SPDR) to the assigned data length as transfer data.

(1) MSB First Transfer (32-Bit Data)

Figure 16.8 shows the operation of the transmit buffer (SPTX) and the shift register when this module performs a 32-bit data length MSB-first data transfer.

The CPU or direct memory access controller writes T31 to T00 to the transmit buffer of SPDR. If the shift register is empty, this module copies the data in the transmit buffer to the shift register, and fully populates the shift register. When serial transfer starts, this module outputs data from the MSB (bit 31) in the shift register, and shifts in the data from the LSB (bit 0) in the shift register. When the RSPCK cycle required for the serial transfer of 32 bits has passed, data R31 to R00 is stored in the shift register. In this state, this module copies the data from the shift register to the receive buffer, and empties the shift register. If the receive buffer does not have a space for the receive data length after the receive data has been copied from the shift register to the receive buffer, another serial transfer will not be started. In order to start another serial transfer, data for the receive data length should be read from the receive buffer to secure the necessary space in the receive buffer.

If another serial transfer is started before the CPU or direct memory access controller writes to the transmit buffer, received data R31 to R00 is shifted out from the shift register.

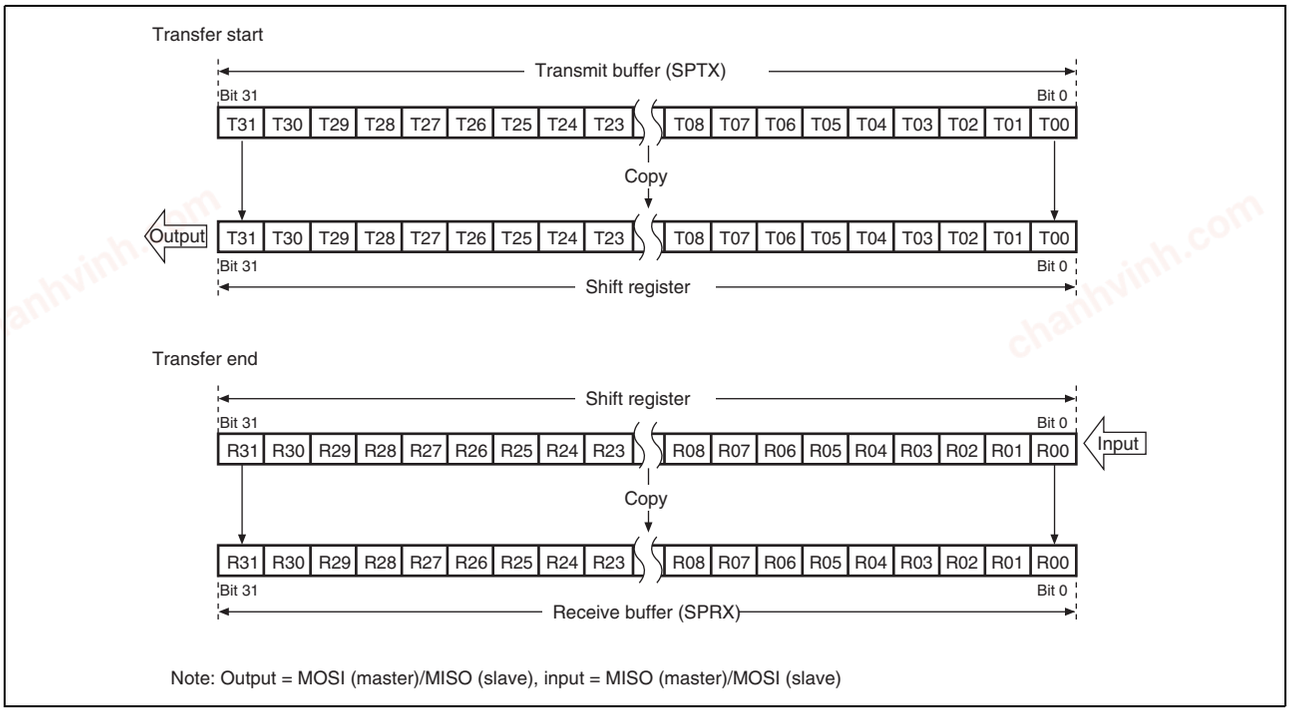


Figure 16.8 MSB First Transfer (32-Bit Data)

chanhvinh.com

(2) MSB First Transfer (16-Bit Data)

Figure 16.9 shows the operation of the transmit buffer (SPTX) and the shift register when this module performs a 16-bit data length MSB-first data transfer.

The CPU or direct memory access controller writes T15 to T00 to the transmit buffer. If the shift register is empty, this module copies the data in the transmit buffer to the shift register, and fully populates the shift register. When serial transfer starts, this module outputs data from bit 15 in the shift register, and shifts in the data from the LSB (bit 0) in the shift register. When the RSPCK cycle required for the serial transfer of 16 bits has passed, received data R15 to R00 is stored in bits 15 to 0 in the shift register. After completion of the serial transfer, data that existed before the transfer is retained in bits 31 to 16 in the shift register. In this state, this module copies the data from the shift register to the receive buffer, and empties the shift register. If the receive buffer does not have a space for the receive data length after receive data has been copied from the shift register to the receive buffer, another serial transfer will not be started. In order to start another serial transfer, data for the receive data length should be read from the receive buffer to secure the necessary space in the receive buffer.

If another serial transfer is started before the CPU or direct memory access controller writes to the transmit buffer, received data R15 to R00 is shifted out from the shift register.

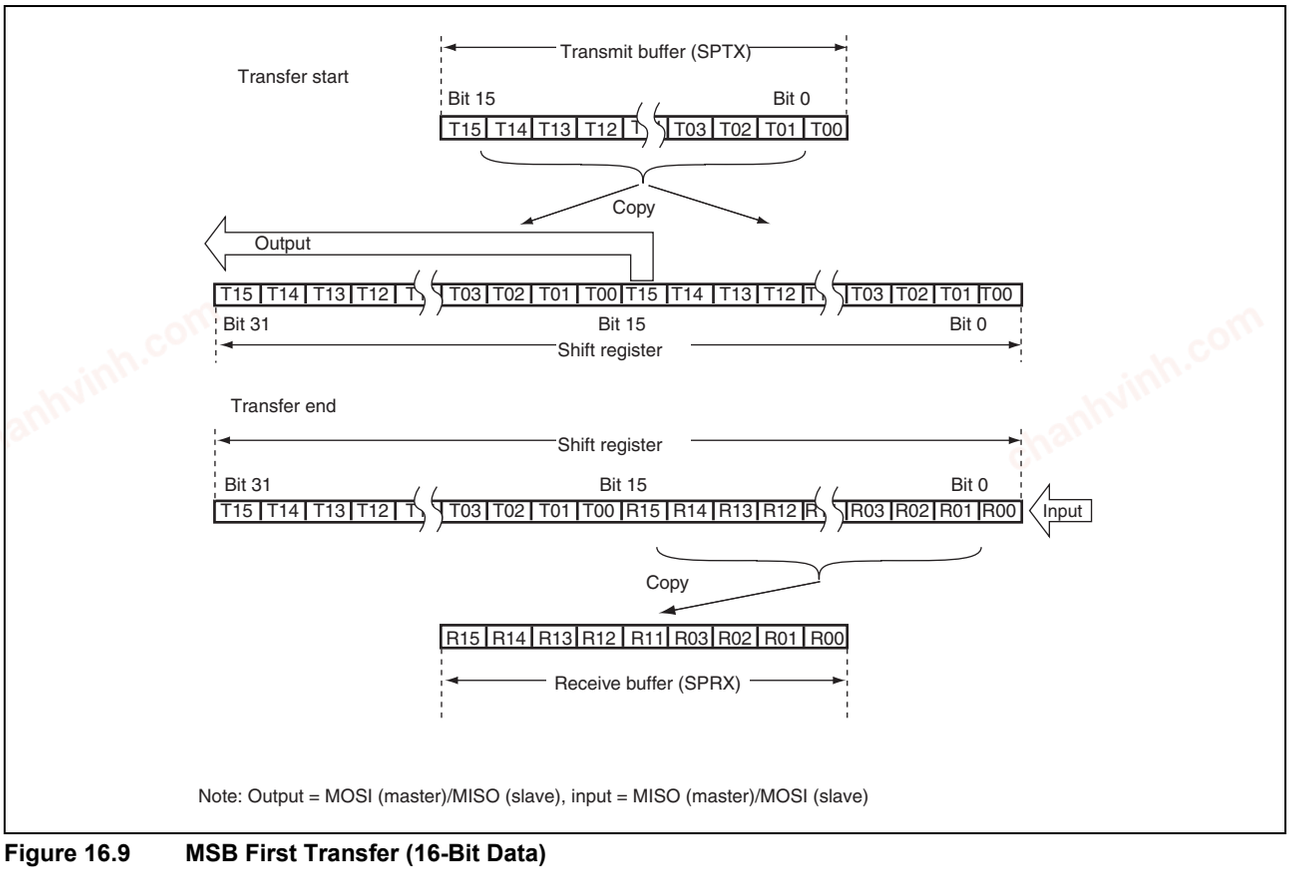


Figure 16.9 MSB First Transfer (16-Bit Data)

chanhvinh.com

(3) MSB First Transfer (8-Bit Data)

Figure 16.10 shows the operation of the transmit buffer (SPTX) and the shift register when this module performs an 8-bit data length MSB-first data transfer.

The CPU or direct memory access controller writes T07 to T00 to the transmit buffer. If the shift register is empty, this module copies the data in the transmit buffer to the shift register, and fully populates the shift register. When serial transfer starts, this module outputs data from bit 7 in the shift register, and shifts in the data from the LSB (bit 0) in the shift register. When the RSPCK cycle required for the serial transfer of 8 bits has passed, received data R07 to R00 is stored in bits 7 to 0 in the shift register. After completion of the serial transfer, data that existed before the transfer is retained in bits 31 to 8 in the shift register. In this state, this module copies the data from the shift register to the receive buffer, and empties the shift register. If the receive buffer does not have a space for the receive data length after receive data has been copied from the shift register to the receive buffer, another serial transfer will not be started. In order to start another serial transfer, data for the receive data length should be read from the receive buffer to secure the necessary area in the receive buffer.

If another serial transfer is started before the CPU or direct memory access controller writes to the transmit buffer, received data R07 to R00 is shifted out from the shift register.

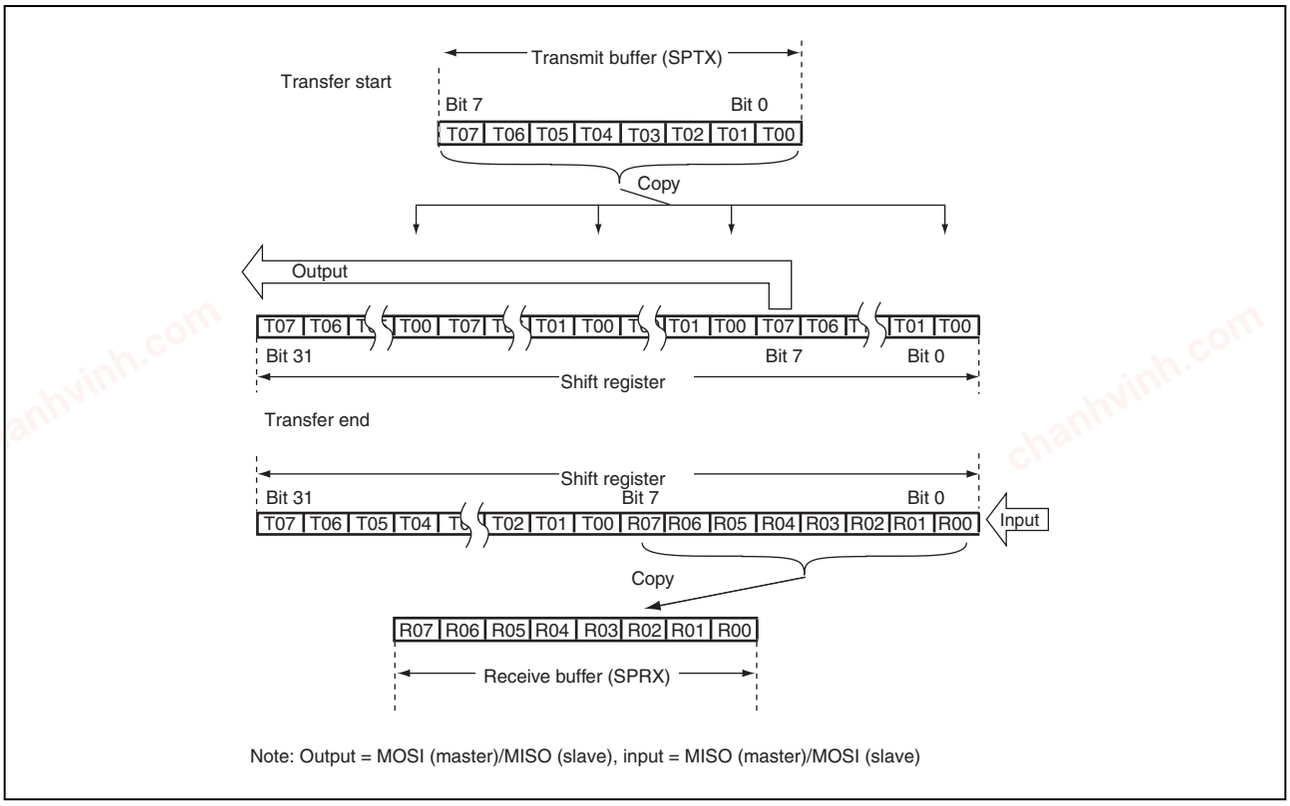


Figure 16.10 MSB First Transfer (8-Bit Data)

chanhvinh.com

(4) LSB First Transfer (32-Bit Data)

Figure 16.11 shows the operation of the transmit buffer (SPTX) and the shift register when this module performs a 32-bit data length LSB-first data transfer.

The CPU or direct memory access controller writes T31 to T00 to the transmit buffer. If the shift register is empty, this module reverses the order of the bits of the data in the transmit buffer, copies it to the shift register, and fully populates the shift register. When serial transfer starts, this module outputs data from the MSB (bit 31) in the shift register, and shifts in the data from the LSB (bit 0) in the shift register. When the RSPCK cycle required for the serial transfer of 32 bits has passed, data R00 to R31 is stored in the shift register. In this state, this module copies the data, in which the order of the bits is reversed, from the shift register to the receive buffer, and empties the shift register. If the receive buffer does not have a space for the receive data length after receive data has been copied from the shift register to the receive buffer, another serial transfer will not be started. In order to start another serial transfer, data for the receive data length should be read from the receive buffer to secure the necessary space in the receive buffer.

If another serial transfer is started before the CPU or direct memory access controller writes to the transmit buffer of the SPDR, received data R00 to R31 is shifted out from the shift register.

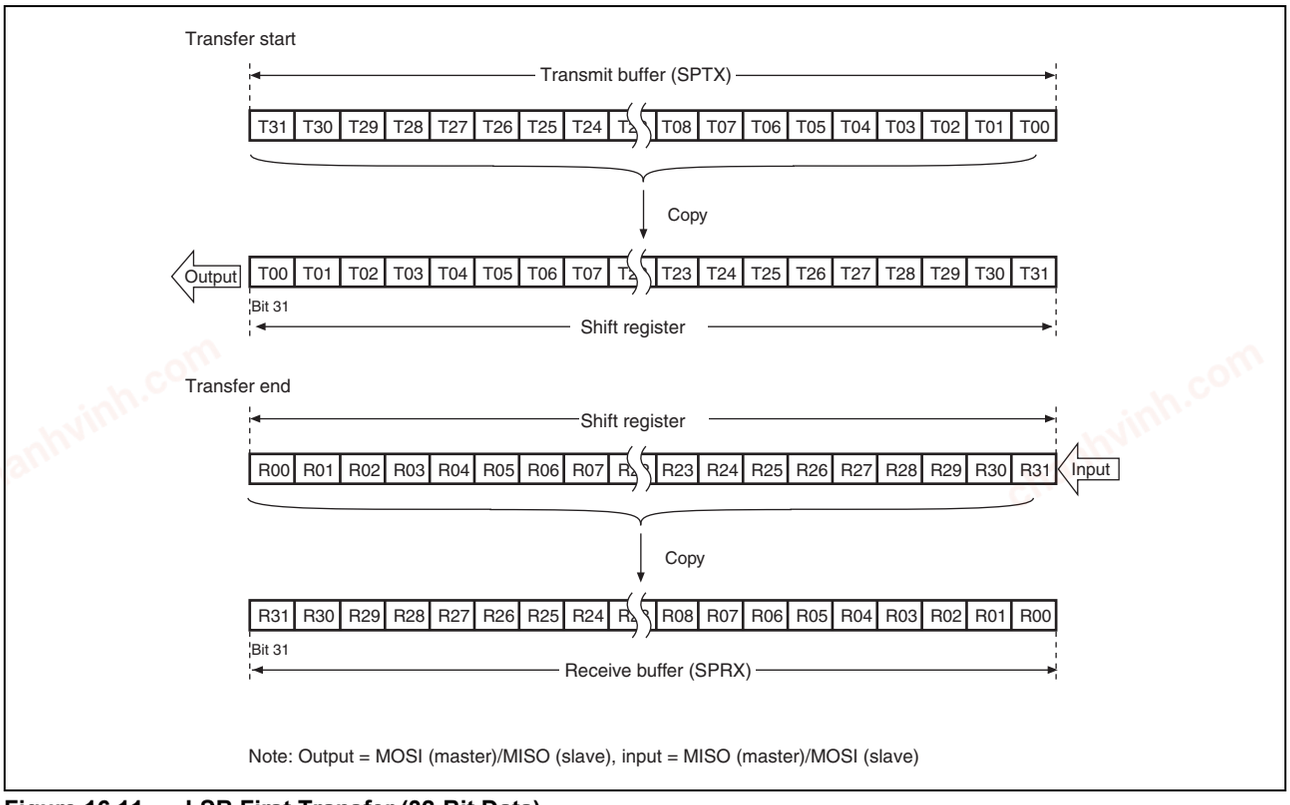


Figure 16.11 LSB First Transfer (32-Bit Data)

chanhvinh.com

(5) LSB First Transfer (16-Bit Data)

Figure 16.12 shows the operation of the transmit buffer (SPTX) and the shift register when this module performs a 16-bit data length LSB-first data transfer.

The CPU or direct memory access controller writes T15 to T00 to the transmit buffer. If the shift register is empty, this module reverses the order of the bits of the data in the transmit buffer, copies it to the shift register, and fully populates the shift register. When serial transfer starts, this module outputs data from the MSB (bit 31) in the shift register, and shifts in the data from bit 16 in the shift register. When the RSPCK cycle required for the serial transfer of 16 bits has passed, received data R00 to R15 is stored in bits 31 to 16 in the shift register. After completion of the serial transfer, data that existed before the transfer is retained in bits 15 to 0 in the shift register. In this state, this module copies the data, in which the order of the bits is reversed, from the shift register to the receive buffer of SPDR, and empties the shift register.

If the receive buffer does not have a space for the receive data length after receive data has been copied from the shift register to the receive buffer, another serial transfer will not be started. In order to start another serial transfer, data for the receive data length should be read from the receive buffer to secure the necessary space in the receive buffer.

If another serial transfer is started before the CPU or direct memory access controller writes to the transmit buffer of SPDR, received data R00 to R15 is shifted out from the shift register.

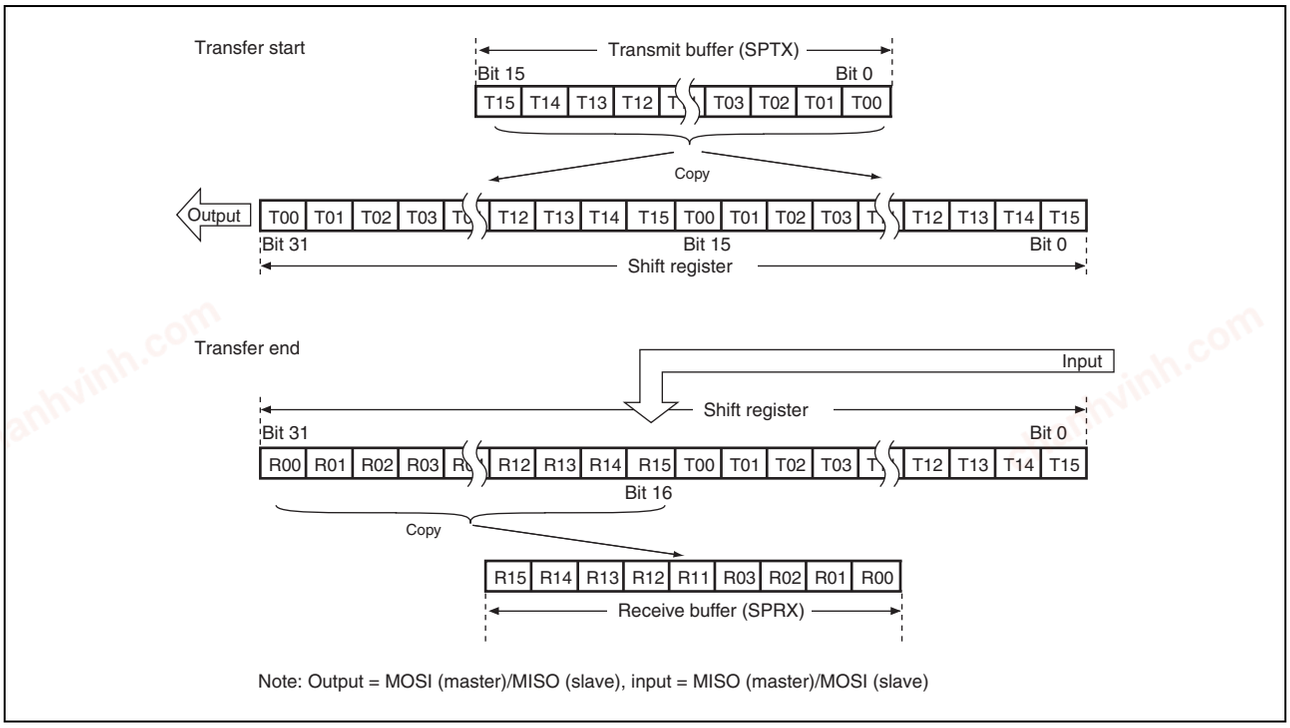


Figure 16.12 LSB First Transfer (16-Bit Data)

chanhvinh.com

(6) LSB First Transfer (8-Bit Data)

Figure 16.13 shows the operation of the transmit buffer (SPTX) and the shift register when this module performs an 8-bit data length LSB-first data transfer.

The CPU or direct memory access controller writes T07 to T00 to the transmit buffer. If the shift register is empty, this module reverses the order of the bits of the data in the transmit buffer, copies it to the shift register, and fully populates the shift register. When serial transfer starts, this module outputs data from the MSB (bit 31) in the shift register, and shifts in the data from bit 24 in the shift register. When the RSPCK cycle required for the serial transfer of 8 bits has passed, received data R00 to R07 is stored in bits 31 to 24 of the shift register. After completion of the serial transfer, data that existed before the transfer is retained in bits 23 to 0 in the shift register. In this state, this module copies the data, in which the order of the bits is reversed, from the shift register to the receive buffer of SPDR, and empties the shift register.

If the receive buffer does not have a space for the receive data length after the receive data has been copied from the shift register to the receive buffer, another serial transfer will not be started. In order to start another serial transfer, data for the receive data length should be read from the receive buffer to secure the necessary space in the receive buffer.

If another serial transfer is started before the CPU or direct memory access controller writes to the transmit buffer of SPDR, received data R00 to R07 is shifted out from the shift register.

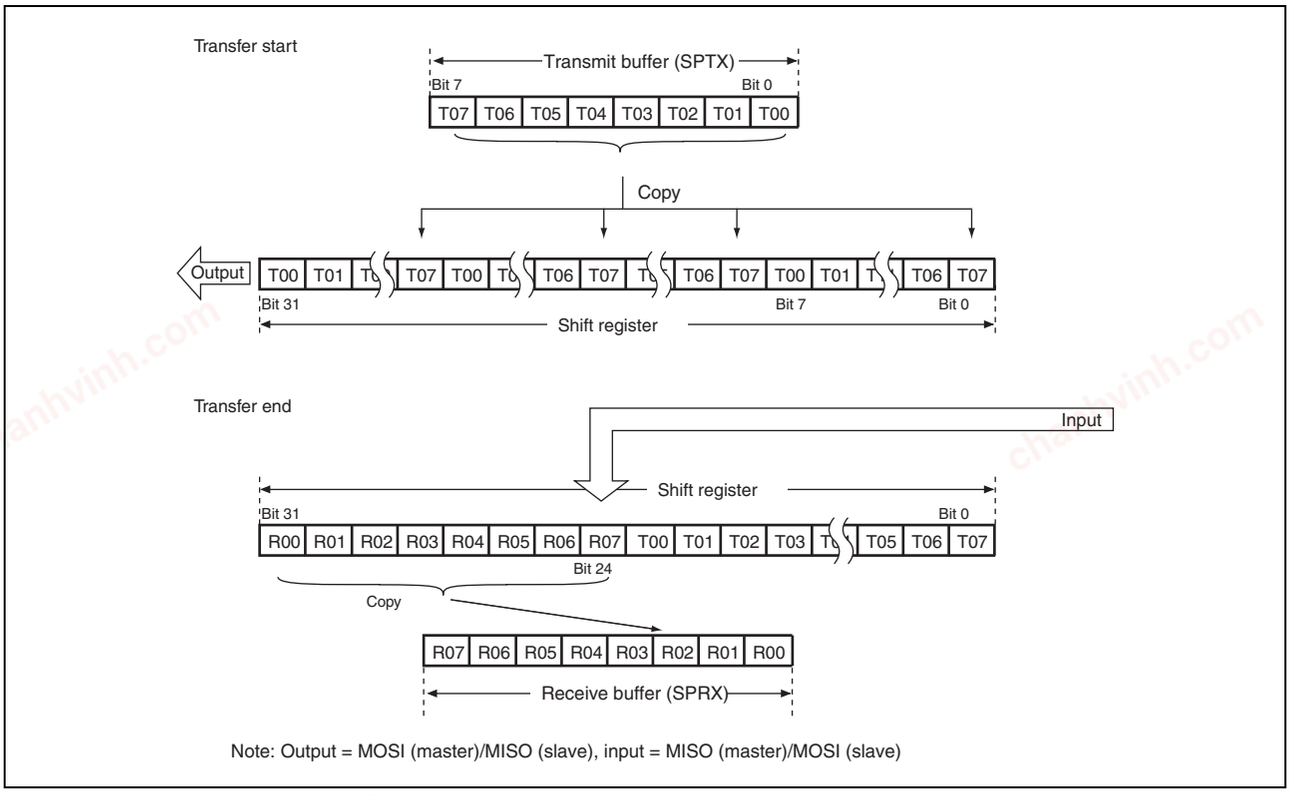


Figure 16.13 LSB First Transfer (8-Bit Data)

chanhvinh.com

16.4.6 Error Detection

In the normal serial transfer, the data written to the transmit buffer of the data register (SPDR) is serially transmitted, and the serially received data can be read from the receive buffer of SPDR. If access is made to SPDR, depending on the status of the transmit buffer/receive buffer or the status at the beginning or end of serial transfer, in some cases non-normal transfers can be executed.

If a non-normal transfer operation occurs, this module detects the event as an overrun error or a mode fault error. Table 16.7 shows the relationship between non-normal transfer operations and the error detection function.

Table 16.7 Relationship between Non-Normal Transfer Operations and Error Detection Function

	Occurrence Condition	Operation	Error Detection
A	SPDR is written when the transmit buffer is full.	Missing write data.	None
B	Serial transfer is started in slave mode when transmit data is still not loaded on the shift register.	Data received in previous serial transfer is serially transmitted.	None
C	SPDR is read when the receive buffer is empty.	The output data is undefined.	None
D	Serial transfer terminates when the receive buffer is full.	Missing serial receive data.	Overrun error (only in slave mode)
E	The SSL input signal is negated during serial transfer in slave mode.	Serial transfer suspended. Missing send/receive data. Operation disabled.	Mode fault error

On operation A shown in Table 16.7, this module does not detect an error. Whether SPDR can be written to or not can be checked using the T[3:0] bits in the buffer data count setting register (SPBFDR).

Likewise, this module does not detect an error on operation B. In a serial transfer that was started before the shift register was updated, this module sends the data that was received in the previous serial transfer, and does not treat the operation indicated in B as an error. Note that the received data from the previous serial transfer is retained in the receive buffer of SPDR, thus it can be correctly read.

Similarly, this module does not detect an error on operation C. To prevent extraneous data from being read, the number of receive data units stored in the receive buffer should be read from the R[5:0] bits in the buffer data count setting register (SPBFDR).

An overrun error shown in D is described in section 16.4.6 (1), Overrun Error. A mode fault error shown in E is described in section 16.4.6 (2), Mode Fault Error.

chanhvinh.com

(1) Overrun Error

If serial transfer ends when the receive buffer of the data register (SPDR) is full, this module detects an overrun error, and sets the OVRF bit in SPSR to 1. When the OVRF bit is 1, this module does not copy data from the shift register to the receive buffer so that the data prior to the occurrence of the error is retained in the receive buffer. To reset the OVRF bit in SPSR to 0, either perform a power-on reset, or write a 0 to the OVRF bit after SPSR has been read with the OVRF bit set to 1.

Figure 16.14 shows an example of operation of the SPRF and OVRF bits in SPSR. The SPSR and SPDR accesses shown in Figure 16.14 indicate the condition of accesses to SPSR and SPDR, respectively, where I denotes an idle cycle, W a write cycle, and R a read cycle. In the example of Figure 16.14, this module performs an 8-bit serial transfer in which the CPHA bit in the command register (SPCMD) is 1, and CPOL is 0. The numbers given under the RSPCK waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

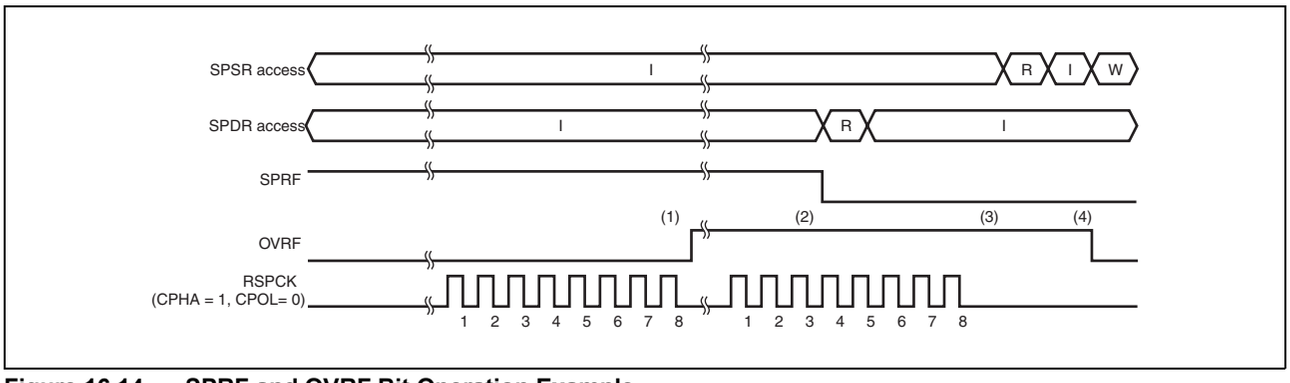


Figure 16.14 SPRF and OVRF Bit Operation Example

The operation of the flags at the timing shown in steps (1) to (4) in the figure is described below.

1. If a serial transfer terminates when the receive buffer does not have a space for the receive data length, this module detects an overrun error, and sets the OVRF bit to 1. This module does not copy the data in the shift register to the receive buffer.
2. The OVFR bit is not cleared even when SPDR is read and thus the number of data bytes in the receive buffer becomes less than the number of the receive buffer data triggering number specified by the RXTRG bits.
3. If the serial transfer terminates in an overrun error state, this module determines that the shift register is empty; in this manner, data transfer is enabled from the transmit buffer to the shift register.
4. If 0 is written to the OVRF bit after SPSR is read with OVRF = 1, this module clears the OVRF bit.

The occurrence of an overrun can be checked either by reading SPSR or by using an error interrupt and reading SPSR. When using an error interrupt, set the SPEIE bit in the control register (SPCR) to 1. When executing a serial transfer without using an error interrupt, measures should be taken to ensure the early detection of overrun errors, such as reading SPSR immediately after SPDR is read.

The OVRF bit is cleared to 0 under the following conditions:

- After SPSR is read in a condition in which the OVRF bit is set to 1, 0 is written to the OVRF bit.
- Power-on reset

Note: • When the receive buffer has area enough to store receive data with an overrun error, this module receives receive data.

chanhvinh.com

(2) Mode Fault Error

When the MSTR bit is 0, this module operates in slave mode. This module detects a mode fault error if the SSL input signal is negated during the serial transfer period (from the time the driving of valid data is started to the time the final valid data is fetched) when the MODFEN bit is 1 in slave mode.
Upon detecting a mode fault error, this module stops driving of the output signals and clears the SPE bit in SPCR to 0. When the SPE bit is cleared to 0, the function of this module is disabled and this module stops driving external signals. For details of disabling the function of this module by clearing the SPE bit to 0, see section 16.4.7, Initialization.
The occurrence of a mode fault error can be checked either by reading SPSR or by using an error interrupt and reading SPSR. When using an error interrupt, set the SPEIE bit in the control register (SPCR) to 1. To detect a mode fault error without using an error interrupt, it is necessary to poll SPSR.
When the MODF bit is 1, writing 1 to the SPE bit is ignored. To enable the function of this module after the detection of a mode fault error, the MODF bit must be set to 0. The MODF bit is cleared to 0 under the following conditions:

- After SPSR is read in a condition where the MODF bit has turned 1, 0 is written to the MODF bit.
- Power-on reset

16.4.7 Initialization

If 0 is written to the SPE bit in the control register (SPCR) or this module clears the SPE bit to 0 because of the detection of a mode fault error, this module disables the module function, and initializes a part of the module function. When a power-on reset is generated, this module initializes all of the module function. An explanation follows of initialization by the clearing of the SPE bit.

(1) Initialization by Clearing SPE Bit

When the SPE bit in SPCR is cleared, this module performs the following initialization:

- Suspending any serial transfer that is being executed
- Stopping the driving of output signals (Hi-Z) in slave mode
- Initializing the internal state
- Initializing the TEND bit in SPSR

Initialization by the clearing of the SPE bit does not initialize the control bits of this module. For this reason, this module can be started in the same transfer mode as prior to the initialization if the SPE bit is re-set to 1.

chanhvinh.com

16.4.8 SPI Operation

(1) Multi-Master Mode Operation

This section explains the operation in multi-master mode.

(a) Starting Serial Transfer

A serial transfer is started when transmit data is copied from the transmit buffer to the shift register, the shift register becomes full, and the receive buffer has a space for the receive data length. If transmit data has already been written to the shift register, data is not copied from the transmit buffer to the shift register.
For details of the transfer format, see section 16.4.4, Transfer Format.

(b) Terminating Serial Transfer

Irrespective of the CPHA bit in the command register (SPCMD), this module terminates the serial transfer after transmitting an RSPCK edge corresponding to the final sampling timing. After the serial transfer is completed, receive data is copied from the shift register to the receive buffer. If the receive buffer does not have a space for the receive data length after receive data is copied from the shift register to the receive buffer, another serial transfer will not be performed. In order to perform another serial transfer, data for the receive data length should be read from the receive buffer to secure the space for the receive data.
It should be noted that the final sampling timing varies depending on the bit length of transfer data. In master mode, the data length depends on the settings in bits SPB3 to SPB0 in SPCMD. For details on the transfer format, see section 16.4.4, Transfer Format.

(c) Sequence Control

The transfer format that is employed in master mode is determined by the sequence control register (SPSCR), command registers 0 to 3 (SPCMD0 to SPCMD3), the bit rate register (SPBR), the clock delay register (SPCKD), the slave select negation delay register (SSLND), and the next-access delay register (SPND).
SPSCR is a register used to determine the sequence configuration for serial transfers that are executed by this module in master mode. The following items are set in command registers SPCMD0 to SPCMD3: SSL output signal value, MSB/LSB first, data length, some of the bit rate settings, RSPCK polarity/phase, whether SPCKD is to be referenced, whether SSLND is to be referenced, and whether SPND is to be referenced. SPBR holds some of the bit rate settings; SPCKD, a clock delay value; SSLND, an SSL negation delay; and SPND, a next-access delay value.
According to the sequence length that is assigned to SPSCR, this module makes up a sequence comprised of a part or all of SPCMD0 to SPCMD3. This module contains a pointer to the SPCMD that makes up the sequence. The value of this pointer can be checked by reading bits SPCP1 and SPCP0 in the sequence status register (SPSSR). When the SPE bit in the control register (SPCR) is set to 1 and the function of this module is enabled, this module loads the pointer to the commands in SPCMD0, and incorporates the SPCMD0 settings into the transfer format at the beginning of serial transfer. This module increments the pointer each time the next-access delay period for a data transfer ends. Upon completion of the serial transfer that corresponds to the final command comprising the sequence, this module sets the pointer in SPCMD0, and in this manner the sequence is executed repeatedly.

chanhvinh.com

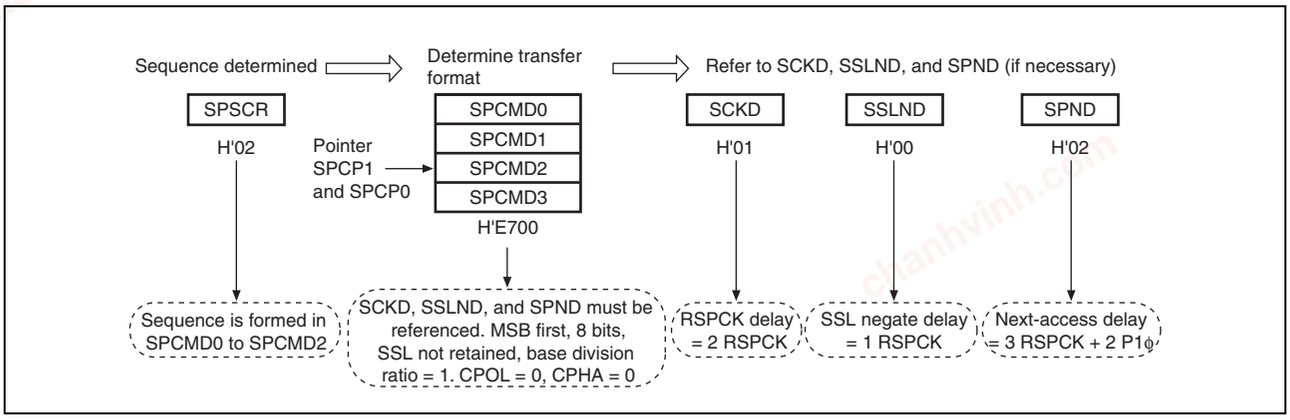


Figure 16.15 Determination Procedure of Serial Transfer Mode in Master Mode

(d) Burst Transfer

If the SSLKP bit in the command register (SPCMD) that this module references during the current serial transfer is 1, this module keeps the SSL signal level during the serial transfer until the beginning of the SSL signal assertion for the next serial transfer. If the SSL signal level for the next serial transfer is the same as the SSL signal level for the current serial transfer, this module can execute continuous serial transfers while keeping the SSL signal assertion status (burst transfer).

Figure 16.16 shows an example of an SSL signal operation for the case where a burst transfer is implemented using SPCMD0 and SPCMD1 settings. The text below explains operations (1) to (7) as depicted in Figure 16.16. It should be noted that the polarity of the SSL output signal depends on the settings in the slave select polarity register (SSLP).

1. Based on SPCMD0, this module asserts the SSL signal and inserts RSPCK delays.
2. Serial transfers are executed according to SPCMD0.
3. SSL negation delays are inserted.
4. Because the SSLKP bit in SPCMD0 is 1, this module keeps the SSL signal value on SPCMD0. This period is sustained, at the shortest, for a period equal to the next-access delay of SPCMD0. If the shift register is empty after the passage of a minimum period, this period is sustained until such time as the transmit data is stored in the shift register for another transfer.
5. Based on SPCMD1, this module asserts the SSL signal and inserts RSPCK delays.
6. Serial transfers are executed according to SPCMD1.
7. Because the SSLKP bit in SPCMD1 is 0, this module negates the SSL signal. In addition, a next-access delay is inserted according to SPCMD1.

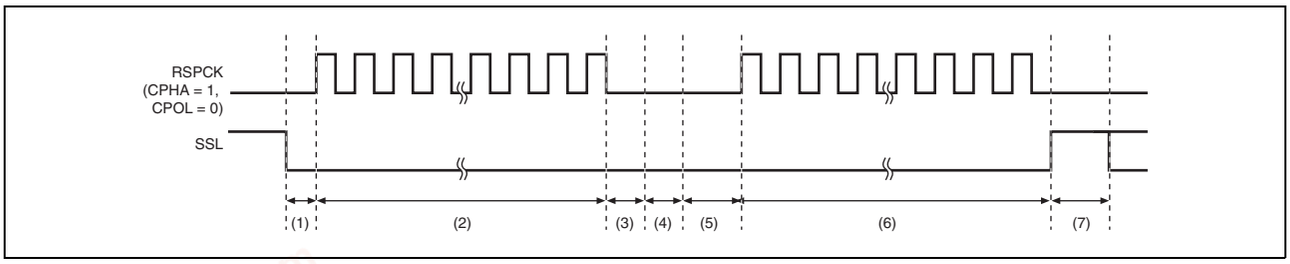


Figure 16.16 Example of Burst Transfer Operation using SSLKP Bit

If the SSL signal settings in the SPCMD in which 1 is assigned to the SSLKP bit are different from the SSL signal output settings in the SPCMD to be used in the next transfer, this module switches the SSL signal status to SSL signal assertion ((5) in Figure 16.16) corresponding to the command for the next transfer. Notice that if such an SSL signal switching occurs, the slaves that drive the MISO signal compete, and the possibility arises of the collision of signal levels.

This module in master mode references within the module the SSL signal operation for the case where the SSLKP bit is

chanhvinh.com

not used. Even when the CPHA bit in SPCMD is 0, this module can accurately start serial transfers by asserting the SSL signal for the next transfer. For this reason, burst transfers in master mode can be executed irrespective of CPHA bit settings (see section 16.4.8 (2), Slave Mode Operation).

(e) RSPCK Delay (t1)

The RSPCK delay value in master mode depends on SCKDEN bit settings in the command register (SPCMD) and on clock delay register (SPCKD) settings. This module determines the SPCMD to be referenced during serial transfer by pointer control, and determines an RSPCK delay value during serial transfer by using the SCKDEN bit in the selected SPCMD and SPCKD, as shown in Table 16.8. For a definition of RSPCK delay, see section 16.4.4, Transfer Format.

Table 16.8 Relationship among SCKDEN and SPCKD Settings and RSPCK Delay Values

SCKDEN	SPCKD	RSPCK Delay Value
0	000 to 111	1 RSPCK
1	000	1 RSPCK
	001	2 RSPCK
	010	3 RSPCK
	011	4 RSPCK
	100	5 RSPCK
	101	6 RSPCK
	110	7 RSPCK
	111	8 RSPCK

(f) SSL Negation Delay (t2)

The SSL negation delay value in master mode depends on SLNDEN bit settings in the command register (SPCMD) and on SSL negation delay register (SSLND) settings. This module determines the SPCMD to be referenced during serial transfer by pointer control, and determines an SSL negation delay value during serial transfer by using the SLNDEN bit in the selected SPCMD and SSLND, as shown in Table 16.9. For a definition of SSL negation delay, see section 16.4.4, Transfer Format.

Table 16.9 Relationship among SLNDEN and SSLND Settings and SSL Negation Delay Values

SLNDEN	SSLND	SSL Negation Delay Value
0	000 to 111	1 RSPCK
1	000	1 RSPCK
	001	2 RSPCK
	010	3 RSPCK
	011	4 RSPCK
	100	5 RSPCK
	101	6 RSPCK
	110	7 RSPCK
	111	8 RSPCK

chanhvinh.com

(g) Next-Access Delay (t3)

The next-access delay value in master mode depends on SPNDEN bit settings in the command register (SPCMD) and on next-access delay register (SPND) settings. This module determines the SPCMD to be referenced during serial transfer by pointer control, and determines a next-access delay value during serial transfer by using the SPNDEN bit in the selected SPCMD and SPND, as shown in Table 16.10. For a definition of next-access delay, see section 16.4.4, Transfer Format.

Table 16.10 Relationship among SPNDEN and SPND Settings and Next-Access Delay Values

SPNDEN	SPND	Next-Access Delay Value
0	000 to 111	1 RSPCK + 2 P1ϕ
1	000	1 RSPCK + 2 P1ϕ
	001	2 RSPCK + 2 P1ϕ
	010	3 RSPCK + 2 P1ϕ
	011	4 RSPCK + 2 P1ϕ
	100	5 RSPCK + 2 P1ϕ
	101	6 RSPCK + 2 P1ϕ
	110	7 RSPCK + 2 P1ϕ
	111	8 RSPCK + 2 P1ϕ

chanhvinh.com

(h) Initialization Flowchart

Figure 16.17 is a flowchart illustrating an example of initialization in SPI operation when this module is used in master mode. For a description of how to set up the interrupt controller, direct memory access controller, and input/output ports, see the descriptions given in the individual blocks.

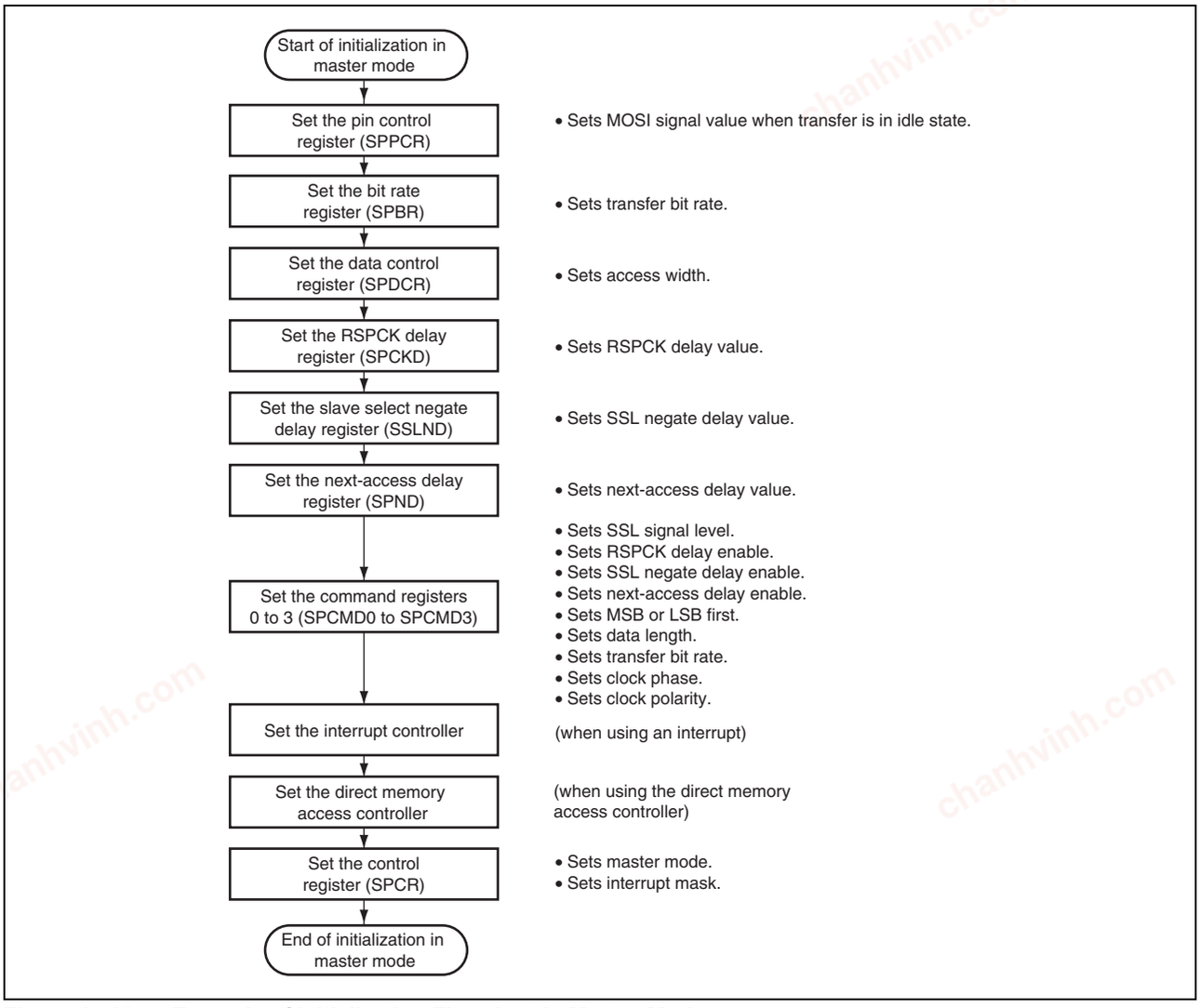


Figure 16.17 Example of Initialization Flowchart in Master Mode

chanhvinh.com

(i) Transfer Operation Flowchart

Figure 16.18 is a flowchart illustrating a transfer in SPI operation when this module is used in master mode.

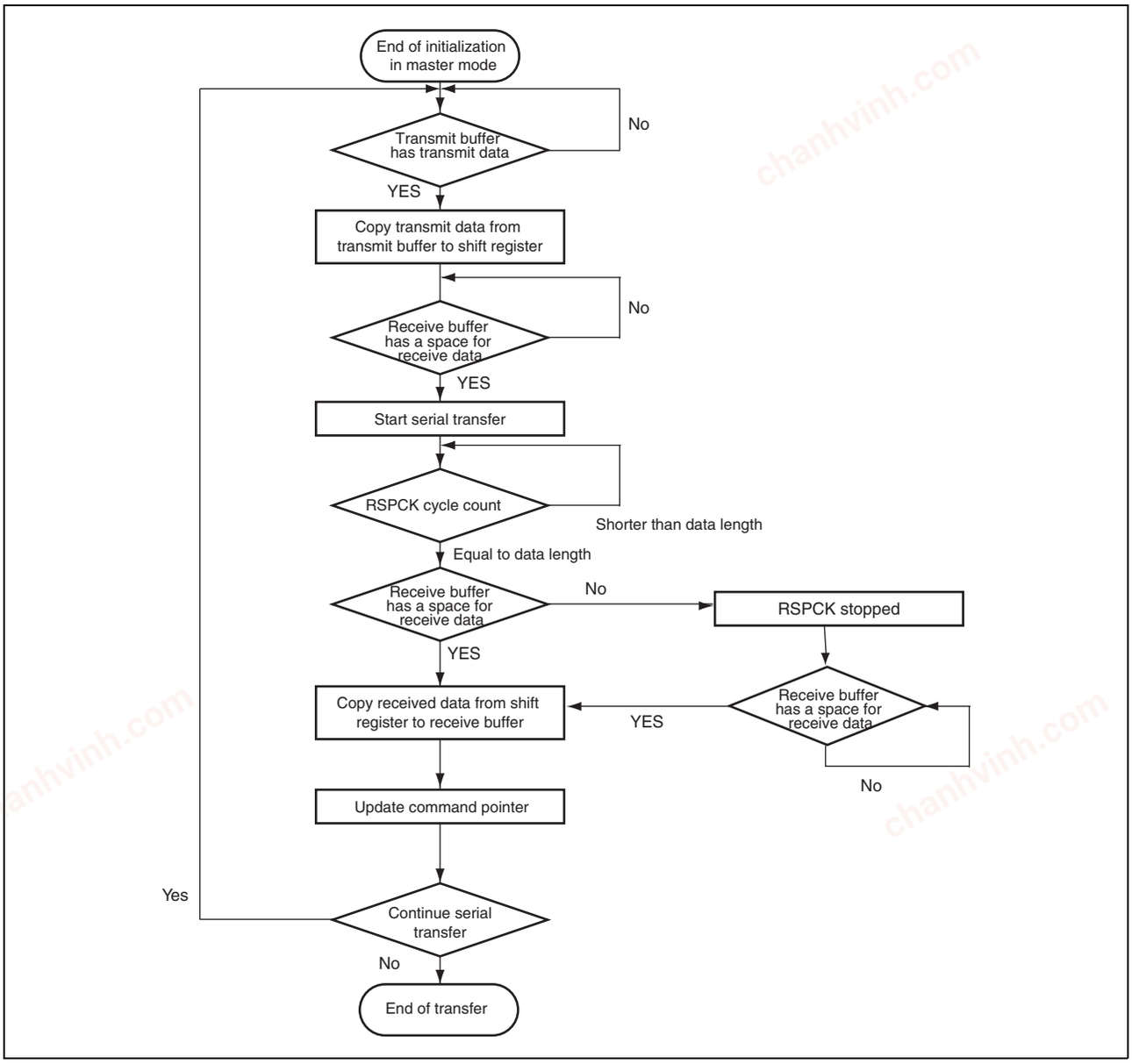


Figure 16.18 Transfer Operation Flowchart in Master Mode

chanhvinh.com

(2) Slave Mode Operation

(a) Starting Serial Transfer

If this module detects an SSL input signal assertion when the CPHA bit in the command register 0 (SPCMD0) is 0, this module is required to start driving valid data to the MISO output signal. For this reason, when the CPHA bit is 0, the asserting of the SSL input signal triggers the start of a serial transfer.
If this module detects the first RSPCK edge in an SSL signal asserted condition when the CPHA bit is 1, this module is required to start driving valid data to the MISO output signal. For this reason, when the CPHA bit is 1, the first RSPCK edge in an SSL signal asserted condition triggers the start of a serial transfer.
When detecting the start of a serial transfer in a condition in which the shift register is empty, this module changes the status of the shift register to "full", so that data cannot be copied from the transmit buffer to the shift register when serial transfer is in progress. If the shift register was full before the serial transfer started, this module leaves the status of the shift register intact, in the full state.
Irrespective of CPHA bit settings, this module starts driving MISO output signals at the SSL signal assertion timing. Whether the data output from this module is valid or invalid differs depending on CPHA bit settings.
For details on the transfer format, see section 16.4.4, Transfer Format. The polarity of the SSL input signal depends on the setting of the SSL0P bit in the slave select polarity register (SSLP).

(b) Terminating Serial Transfer

Irrespective of the CPHA bit in the command register 0 (SPCMD0), this module terminates the serial transfer after detecting an RSPCK edge corresponding to the final sampling timing. When the receive buffer has an enough space for receive data, this module copies received data from the shift register to the receive buffer of the data register (SPDR) upon termination of the serial transfer. Irrespective of the value of the SPRF bit, this module changes the status of the shift register to "empty" upon termination of the serial transfer. If this module detects an SSL input signal negation from the beginning of serial transfer to the end of serial transfer, a mode fault error occurs (see section 16.4.6, Error Detection).
The final sampling timing changes depending on the bit length of the transfer data. In slave mode, the data length depends on the settings in bits SPB3 to SPB0 bits in SPCMD0. The polarity of the SSL input signal depends on the setting in the SSL0P bit in the slave select polarity register (SSLP). For details on the transfer format, see section 16.4.4, Transfer Format.

(c) Notes on Slave Operations

If the CPHA bit in the command register 0(SPCMD0) is 0, this module starts serial transfers when it detects the assertion edge for an SSL input signal. In the type of configuration shown in Figure 16.4 as an example, if this module is used in single-slave mode, the SSL signal is always fixed at active state. Therefore, when the CPHA bit is set to 0, this module cannot correctly start a serial transfer. To correctly execute send/receive operation in a configuration in which the SSL input signal is fixed at active state, the CPHA bit should be set to 1. When it is necessary to set the CPHA bit to 0, the SSL input signal should not be fixed.

(d) Burst Transfer

If the CPHA bit in the command register 0 (SPCMD0) is 1, continuous serial transfer (burst transfer) can be executed while retaining the assertion state for the SSL input signal. If the CPHA bit is 1, the period from the first RSPCK edge to the sampling timing for the reception of the final bit in an SSL signal active state corresponds to a serial transfer period. Even when the SSL input signal remains at the active level, this module can accommodate burst transfers because it can detect the start of access.
If the CPHA bit is 0, for the reason given in section 16.4.8 (2) (c), Notes on Slave Operations, second and subsequent serial transfers during the burst transfer cannot be executed correctly.

chanhvinh.com

(e) Initialization Flowchart

Figure 16.19 is a flowchart illustrating an example of initialization in SPI operation when this module is used in slave mode. For a description of how to set up the interrupt controller, direct memory access controller, and input/output ports, see the descriptions given in the individual blocks.

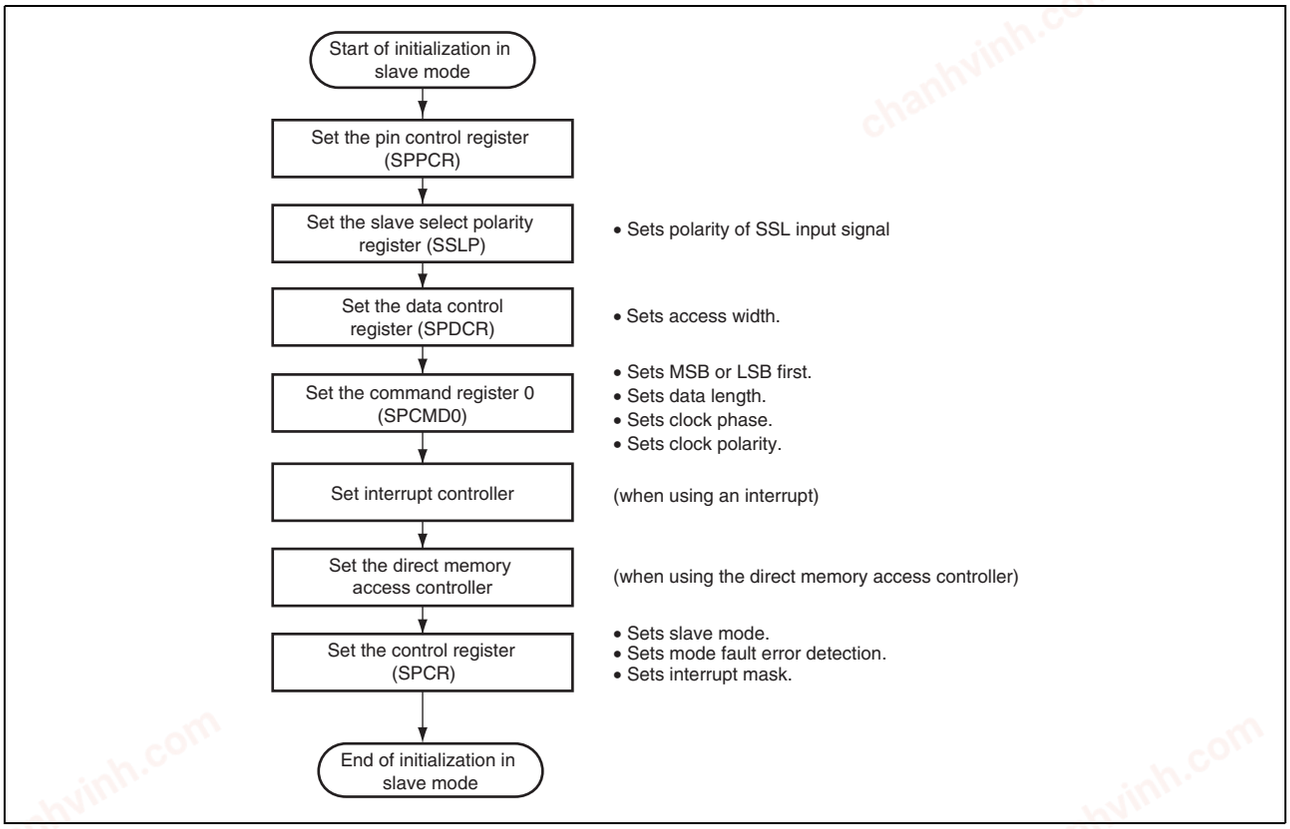


Figure 16.19 Example of Initialization Flowchart in Slave Mode

chanhvinh.com

(f) Transfer Operation Flowchart (CPHA = 0)

Figure 16.20 is a flowchart illustrating a transfer in SPI operation when this module is used in slave mode with the CPHA bit in the command register 0 (SPCMD0) set to 0.

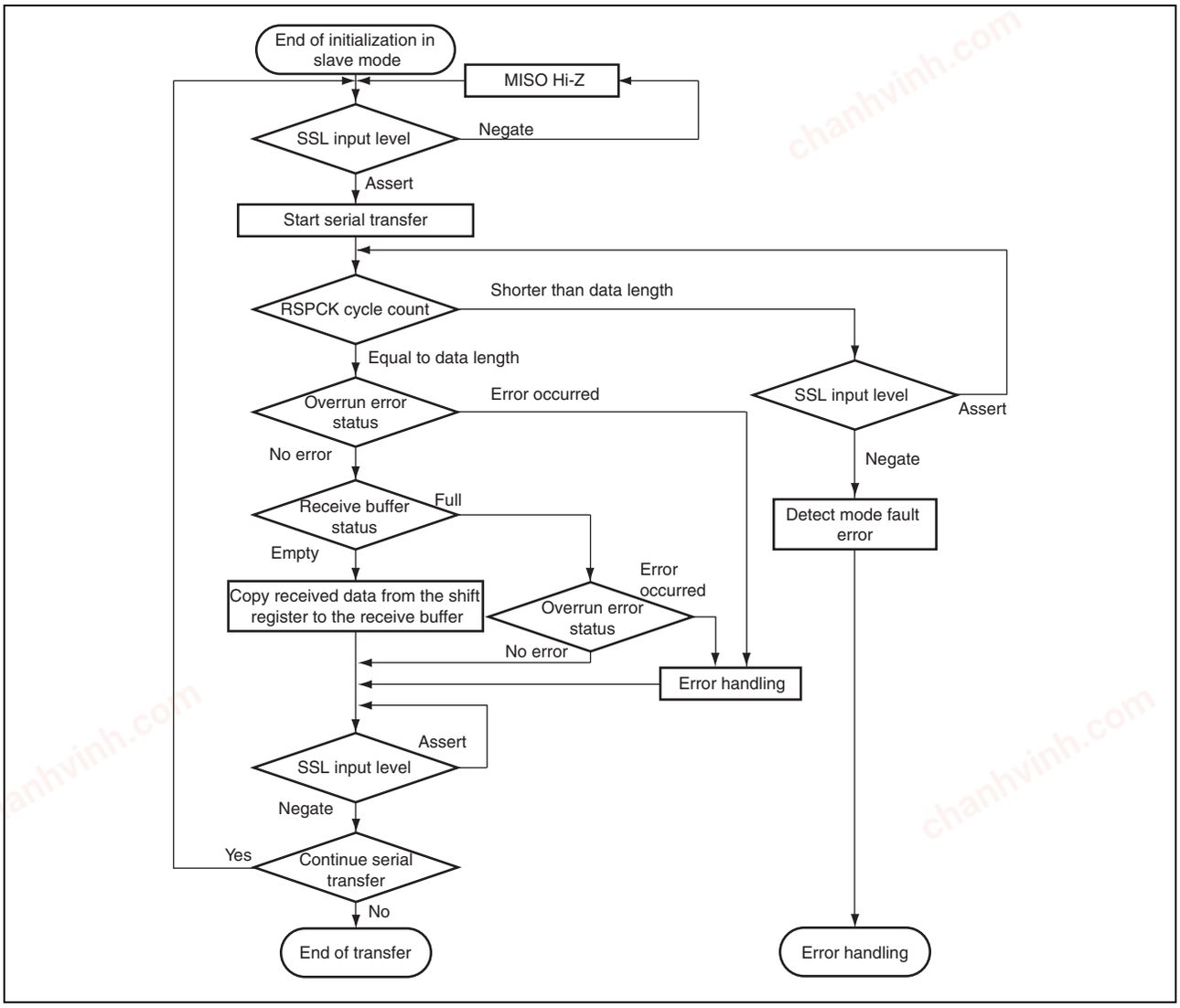


Figure 16.20 Transfer Operation Flowchart in Slave Mode (CPHA = 0)

chanhvinh.com

(g) Transfer Operation Flowchart (CPHA = 1)

Figure 16.21 is a flowchart illustrating a transfer in SPI operation when this module is used in slave mode with the CPHA bit in the command register 0 (SPCMD0) and the MODFEN bit in the control register (SPCR) set to 1, respectively. The subsequent operation is not guaranteed when the serial transfer is started with the MODFEN bit set to 0 and the SSL input level is negated with the number of RSPCK cycles shorter than the data length.

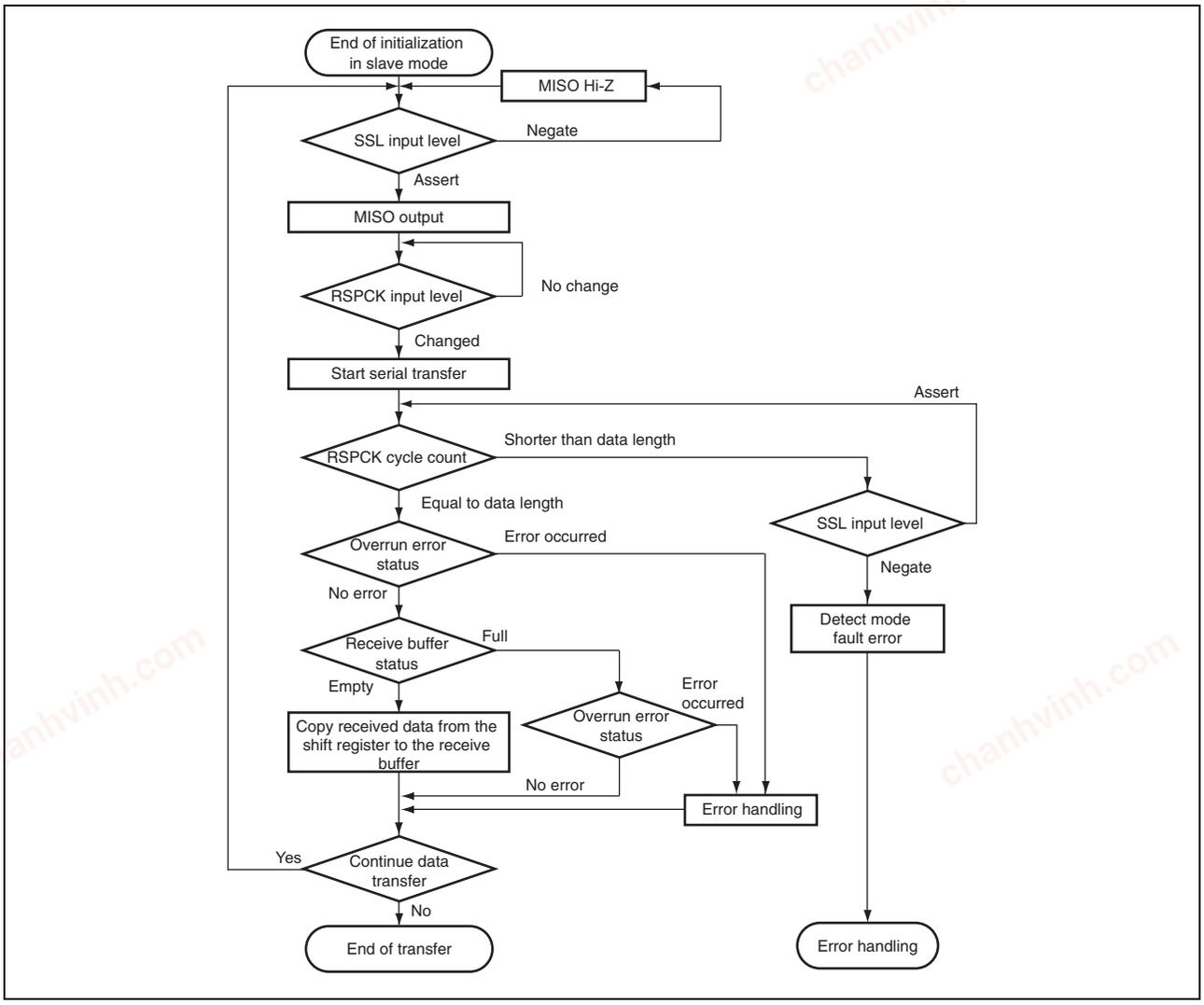


Figure 16.21 Transfer Operation Flowchart in Slave Mode (CPHA = 1)

chanhvinh.com

16.4.9 Error Handling

Figure 16.22 and Figure 16.23 show the error handling. The following error handling is used to return from the error state after an error in master or slave mode.

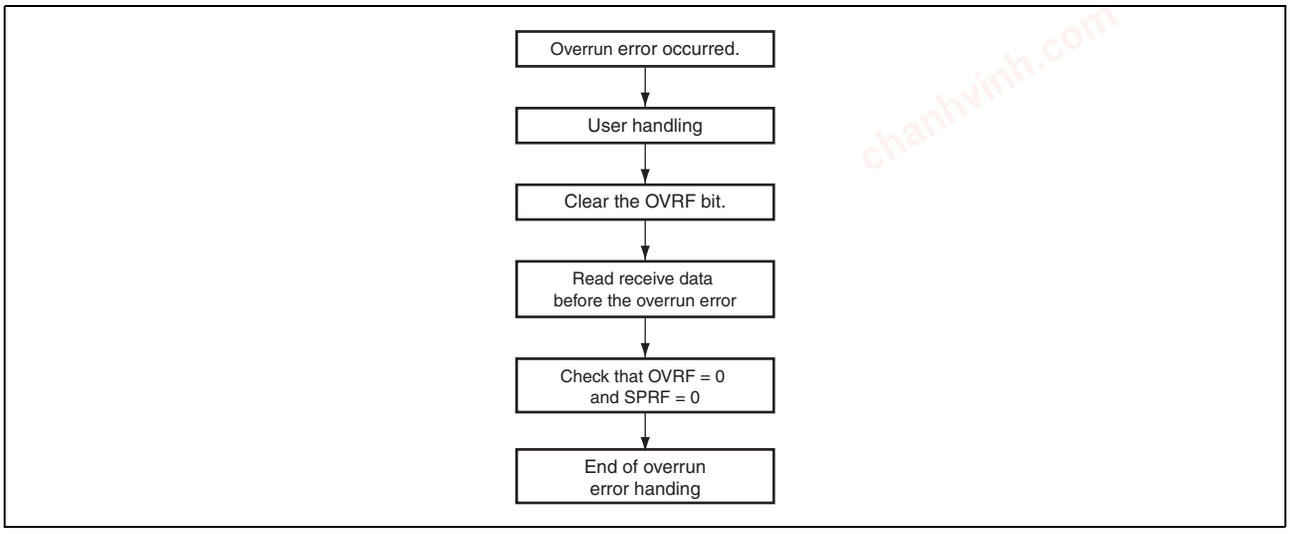


Figure 16.22 Error Handling (Overrun Error)

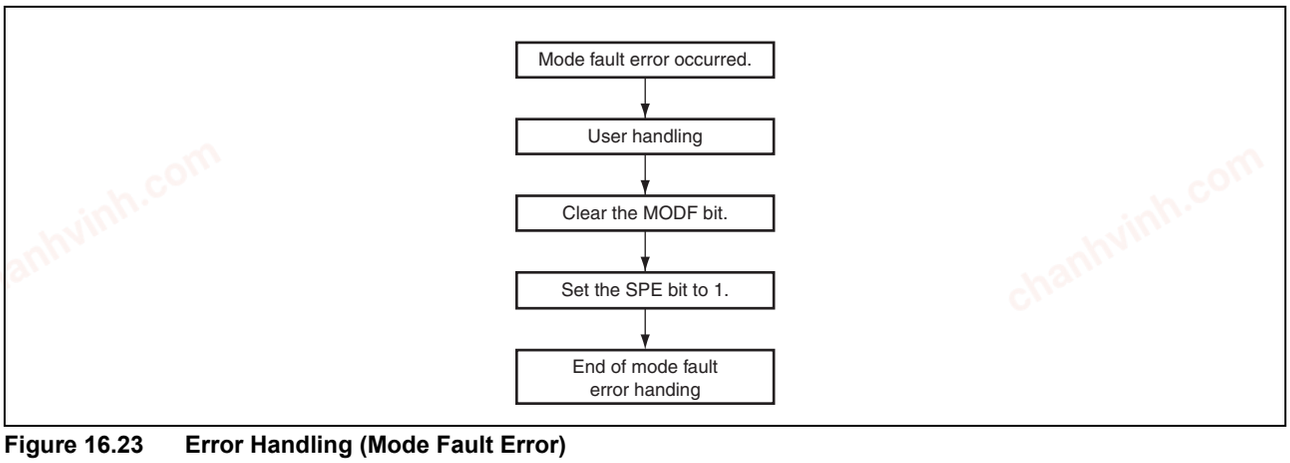


Figure 16.23 Error Handling (Mode Fault Error)

chanhvinh.com

16.4.10 Loopback Mode

When 1 is written to the SPLP bit in the pin control register (SPPCR), this module shuts off the path between the MISO pin and the shift register, and between the MOSI pin and the shift register, and connects the input path and the output path (reversed) of the shift register. This is called loopback mode. When a serial transfer is executed in loopback mode, the transmit data becomes the received data. Figure 16.24 shows the configuration of the shift register input/output paths for the case where this module in master mode is set in loopback mode.

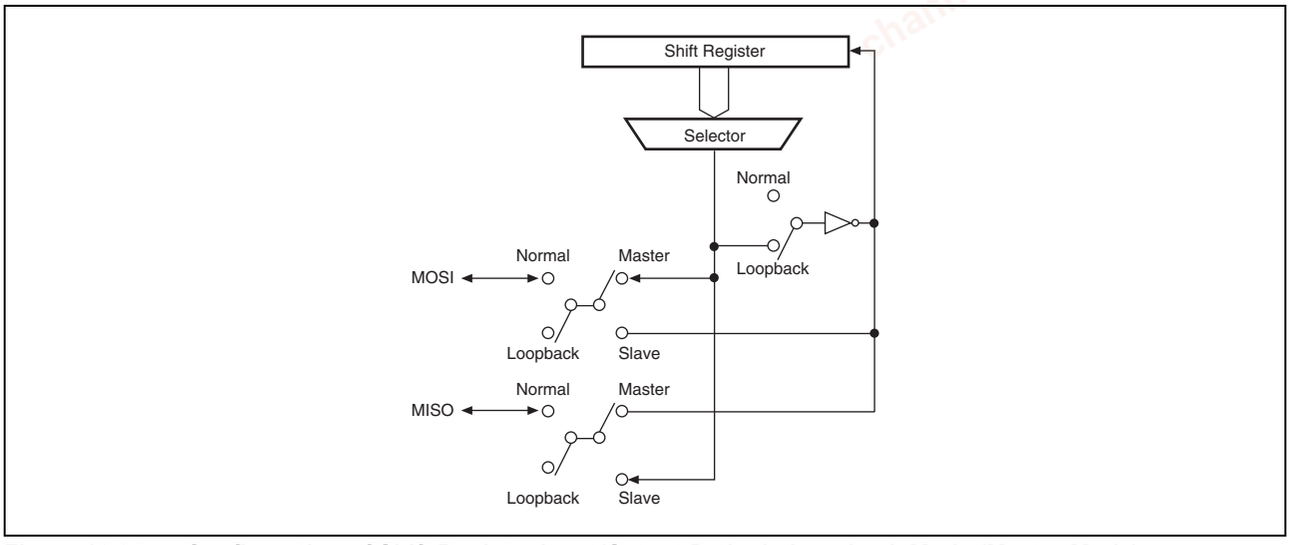


Figure 16.24 Configuration of Shift Register Input/Output Paths in Loopback Mode (Master Mode)

16.4.11 Interrupt Sources

This module has interrupt sources of receive buffer full, transmit buffer empty, mode fault, and overrun. In addition, the direct memory access controller can be activated by the receive buffer full or transmit buffer empty interrupt for data transfer.

Table 16.11 shows the interrupt sources.

When any of the interrupt conditions in Table 16.11 is met, an interrupt is generated. The interrupt sources should be cleared with data transfer by the CPU or direct memory access controller.

Table 16.11 Interrupt Sources

Name	Interrupt Source	Abbreviation	Interrupt Condition	Activation of Direct Memory Access Controller
SPRI	Receive buffer full	RXI	(SPRIE = 1) • (SPRF = 1)	Possible
SPTI	Transmit buffer empty	TXI	(SPTIE = 1) • (SPTEF = 1)	Possible
SPEI	Mode fault	MOI	(SPEIE = 1) • (MODF = 1)	—
	Overrun	OVI	(SPEIE = 1) • (OVRF = 1)	—

chanhvinh.com

17. SPI Multi I/O Bus Controller

The SPI multi I/O bus controller outputs control signals to the serial flash memory connected to the SPI multi I/O bus space, thus enabling direct connection of the serial flash memory.
This LSI incorporates an independent SPI multi I/O bus controller channel.

17.1 Features

This module allows the connected serial flash memory to be accessed by directly reading the SPI multi I/O bus space, or using SPI operating mode to transmit and receive data.

- Serial Flash Memory Interface
  Up to two serial flash memories per channel can be connected.
  A data bus size of 1 bit, 2 bits, or 4 bits can be selected for one serial flash memory device.
  Serial flash memory for DDR transfer can be directly connected (RZ/A1LU only).
- External Address Space Read Mode
  A maximum of 8-Gbyte address space is supported (when two serial flash memories are connected)
  The SPBSSL pin can be automatically controlled through access address monitoring
  Efficient data reception due to built-in read cache (64-bit line × 16 entries)
- SPI Operating Mode
  Desired read/write access to serial flash memory possible
- Bit rate
  SPBCLK is generated by frequency division of Bϕ by internal baud rate generator
  SPBCLK frequency division ratio can be set from 2 to 4080
- SPBSSL Pin Control
  Delay from SPBSSL signal assertion to SPBCLK operation (clock delay) can be set
  Range: 1 to 8 SPBCLK cycles (set in SPBCLK-cycle units)
  Delay from SPBCLK stop to SPBSSL output negation (SPBSSL negation delay) can be set
  Range: 1.5 to 8.5 SPBCLK cycles (set in SPBCLK-cycle units)
  SPBSSL output assertion wait before next access (next access delay) can be set
  Range: 1 to 8 SPBCLK cycles (set in SPBCLK-cycle units)
  SPBSSL polarity can be changed

chanhvinh.com

17.2 Block Diagram

Figure 17.1 shows a block diagram of this module for one channel.

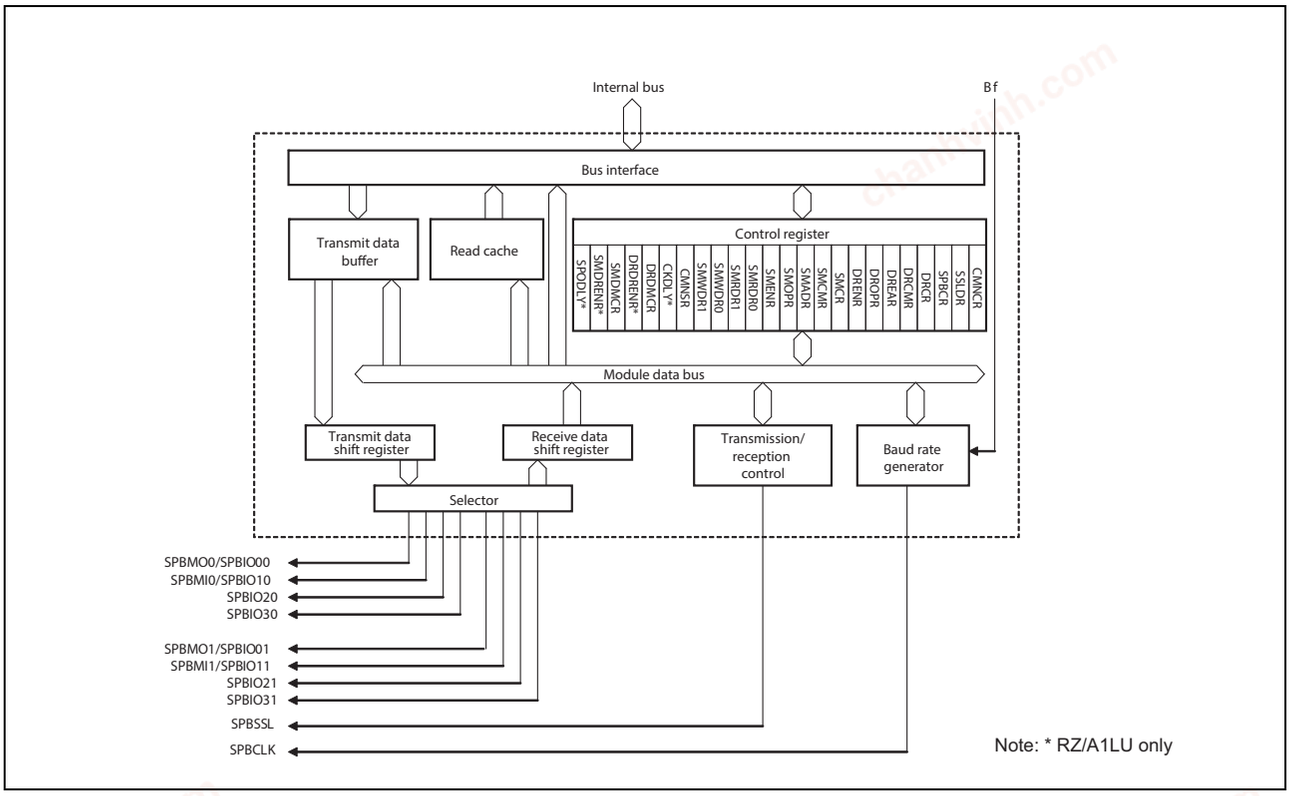


Figure 17.1 Block Diagram

chanhvinh.com

17.3 Input/Output Pins

Table 17.1 shows the pin configuration for one channel.

Table 17.1 Pin Configuration

Port	Pin Name	Symbol	I/O	Function
Common	Clock pin	SPBCLK_n	Output	Clock output
	Slave select pin	SPBSSL_n	Output	Slave selection
0	Data 0 pin	SPBMO0_n/ SPBIO00_n	I/O	Port 0 master transmit data/data 0
	Port 0 data 1 pin	SPBMI0_n/ SPBIO10_n	I/O	Port 0 master input data/ data 1
	Port 0 data 2 pin	SPBIO20_n	I/O	Port 0 data 2
	Port 0 data 3 pin	SPBIO30_n	I/O	Port 0 data 3
1	Port 1 data 0 pin	SPBMO1_n/ SPBIO01_n	I/O	Port 1 master transmit data/data 0
	Port 1 data 1 pin	SPBMI1_n/ SPBIO11_n	I/O	Port 1 master input data/ data 1
	Port 1 data 2 pin	SPBIO21_n	I/O	Port 1 data 2
	Port 1 data 3 pin	SPBIO31_n	I/O	Port 1 data 3

Note: n represents a channel number (0). In the text, the channel number is omitted.

chanhvinh.com

17.4 Register Descriptions

Table 17.2 shows the register configuration.

Table 17.2 Register Configuration

Channel	Register Name	Abbreviation	Initial Value	R/W	Address	Access Size
0	Common control register_0	CMNCR_0	H'01AA4000	R/W	H'3FEFA000	32
	SSL delay register_0	SSLDR_0	H'00070707	R/W	H'3FEFA004	32
	Bit rate register_0	SPBCR_0	H'00000003	R/W	H'3FEFA008	32
	Data read control register_0	DRCR_0	H'00000000	R/W	H'3FEFA00C	32
	Data read command setting register_0	DRCMR_0	H'00030000	R/W	H'3FEFA010	32
	Data read extended address setting register_0	DREAR_0	H'00000000	R/W	H'3FEFA014	32
	Data read option setting register_0	DROPR_0	H'00000000	R/W	H'3FEFA018	32
	Data read enable setting register_0	DRENR_0	H'00004700	R/W	H'3FEFA01C	32
	SPI mode control register_0	SMCR_0	H'00000000	R/W	H'3FEFA020	32
	SPI mode command setting register_0	SMCMR_0	H'00000000	R/W	H'3FEFA024	32
	SPI mode address setting register_0	SMADR_0	H'00000000	R/W	H'3FEFA028	32
	SPI mode option setting register_0	SMOPR_0	H'00000000	R/W	H'3FEFA02C	32
	SPI mode enable setting register_0	SMENR_0	H'00004000	R/W	H'3FEFA030	32
	SPI mode read data register 0_0	SMRDR0_0	Undefined	R	H'3FEFA038	8, 16, 32
	SPI mode read data register 1_0	SMRDR1_0	Undefined	R	H'3FEFA03C	8, 16, 32
	SPI mode write data register 0_0	SMWDR0_0	H'00000000	R/W	H'3FEFA040	8, 16, 32
	SPI mode write data register 1_0	SMWDR1_0	H'00000000	R/W	H'3FEFA044	8, 16, 32
	Common status register_0	CMNSR_0	H'00000001	R	H'3FEFA048	32
	SPI AC input characteristics adjustment register_0 (RZ/A1LU only)	CKDLY_0	H'00000004	R/W	H'3FEFA050	32
	Data read dummy cycle setting register_0	DRDMCR_0	H'00000000	R/W	H'3FEFA058	32
	Data read DDR enable register_0 (RZ/A1LU only)	DRDRENR_0	H'00000000	R/W	H'3FEFA05C	32
	SPI mode dummy cycle setting register_0	SMDMCR_0	H'00000000	R/W	H'3FEFA060	32
	SPI mode DDR enable register_0 (RZ/A1LU only)	SMDRENR_0	H'00000000	R/W	H'3FEFA064	32
	SPI AC output characteristics adjustment register_0 (RZ/A1LU only)	SPODLY_0	H'00000000	R/W	H'3FEFA068	32

Note: In the text, the channel number is omitted.

chanhvinh.com

17.4.1 Common Control Register (CMNCR)

CMNCR is a 32-bit register that controls the SPI multi I/O bus controller. The settings of this register are reflected both in external address space read mode and SPI operating mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MD	-	-	-	-	-	-	SFDE	MOIIO3[1:0]		MOIIO2[1:0]		MOIIO1[1:0]		MOIIO0[1:0]	
Initial value:	0	0	0	0	0	0	0	1	1	0	1	0	1	0	1	0
R/W:	R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IO3FV[1:0]		IO2FV[1:0]		-	-	IO0FV[1:0]		-	CPHAT	CPHAR	SSLP	CPOL	-	BSZ[1:0]	
Initial value:	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	MD	0	R/W	Operating Mode Switch Switches the operating modes. 0: External address space read mode 1: SPI operating mode
30 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	SFDE	1	R/W	Data Swap Setting for Serial Flash Memory Specifies whether or not swapping of data in serial flash memory is performed. 0: Swapping is not performed. 1: Swapping is performed in 8-bit units. For details, see section 17.5.4, Data Alignment.
23, 22	MOIIO3[1:0]	10	R/W	SPBSSL Output Idle Value Fix SPBIO30, SPBIO31 Fixes output values of SPBIO30 and SPBIO31 in SPBSSL negation period. 00: Output value 0 01: Output value 1 10: Output value is the value of the immediately previous bit (or the pin is Hi-Z, if Hi-Z was the state in the immediately previous bit period). 11: Output value Hi-Z
21, 20	MOIIO2[1:0]	10	R/W	SPBSSL Output Idle Value Fix SPBIO20, SPBIO21 Fixes output values of SPBIO20 and SPBIO21 in SPBSSL negation period. 00: Output value 0 01: Output value 1 10: Output value is the value of the immediately previous bit (or the pin is Hi-Z, if Hi-Z was the state in the immediately previous bit period). 11: Output value Hi-Z
19, 18	MOIIO1[1:0]	10	R/W	SPBSSL Output Idle Value Fix SPBIO10, SPBIO11 Fixes output values of SPBIO10 and SPBIO11 in SPBSSL negation period. 00: Output value 0 01: Output value 1 10: Output value is the value of the immediately previous bit (or the pin is Hi-Z, if Hi-Z was the state in the immediately previous bit period). 11: Output value Hi-Z

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
17, 16	MOIIO0[1:0]	10	R/W	SPBSSL Output Idle Value Fix SPBIO00, SPBIO01 Fixes output values of SPBIO00 and SPBIO01 in SPBSSL negation period. 00: Output value 0 01: Output value 1 10: Output value is the value of the immediately previous bit (or the pin is Hi-Z, if Hi-Z was the state in the immediately previous bit period). 11: Output value Hi-Z
15, 14	IO3FV[1:0]	01	R/W	SPBIO30, SPBIO31 Fixed Value for 1-bit/2-bit Size Fixes the output value of SPBIO30 and SPBIO31 pins for 1-bit/2-bit size. 00: Output value 0 01: Output value 1 10: Output value is the value of the immediately previous bit (or the pin is Hi-Z, if Hi-Z was the state in the immediately previous bit period). 11: Output value Hi-Z
13, 12	IO2FV[1:0]	00	R/W	SPBIO20, SPBIO21 Fixed Value for 1-bit/2-bit Size Fixes the output value of SPBIO20 and SPBIO21 pins for 1-bit/2-bit size. 00: Output value 0 01: Output value 1 10: Output value is the value of the immediately previous bit (or the pin is Hi-Z, if Hi-Z was the state in the immediately previous bit period). 11: Output value Hi-Z
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	IO0FV[1:0]	00	R/W	SPBIO00, SPBIO01 Fixed Value for 1-bit Size Input Fixes the output value of SPBIO00 and SPBIO01 pins for 1-bit size input. 00: Output value 0 01: Output value 1 10: Output value is the value of the immediately previous bit (or the pin is Hi-Z, if Hi-Z was the state in the immediately previous bit period). 11: Output value Hi-Z
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	CPHAT	0	R/W	Output Shift Sets the SPBCLK edge of the output data. CPHAT and CPHAR should be set according to the table in the description of CPHAR. 0: Data transmission at even edge during SDR transfer Data transmission starts at even edge during DDR transfer (RZ/A1LU only). 1: Data transmission at odd edge during SDR transfer Data transmission starts at odd edge during DDR transfer (RZ/A1LU only).
5	CPHAR	0	R/W	Input Latch Sets the SPBCLK edge of the reception data. CPHAT and CPHAR should be set according to the following table. 0: Data reception at odd edge during SDR transfer Data reception starts at odd edge during DDR transfer (RZ/A1LU only). 1: Data reception at even edge during SDR transfer Data reception starts at even edge during DDR transfer (RZ/A1LU only).

CPHAT	CPHAR	
0	0	Setting enabled
0	1	Setting enabled
1	0	Setting prohibited
1	1	Setting enabled

Note: To set DDR transfer, set both the CPHAT and CPHAR bits to 0 or 1 (RZ/A1LU only).

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
4	SSLP	0	R/W	SPBSSL Signal Polarity Sets the polarity of SPBSSL signal. 0: Active low SPBSSL signal 1: Active high SPBSSL signal
3	CPOL	0	R/W	SPBSSL Negation Period SPBCLK Output Direction Sets the SPBCLK output direction during SPBSSL negation period. 0: SPBCLK output is 0 during SPBSSL negation period. 1: SPBCLK output is 1 during SPBSSL negation period.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1, 0	BSZ[1:0]	00	R/W	Data Bus Size Specifies the number of serial flash memories to be connected. 00: 1 memory 01: 2 memories 1X: Setting prohibited Note: After changing (the value of) this bit, all the entries in the read cache must be cleared by setting the RCF bit in DRCR to 1.

chanhvinh.com

17.4.2 SSL Delay Register (SSLDR)

SSLDR is a 32-bit register that adjusts the timing between the SPBSSL signal and the SPBCLK signal.
The settings of this register are reflected both in external address space read mode and SPI operating mode.
The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	SPNDL[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	SLNDL[2:0]			-	-	-	-	-	SCKDL[2:0]		
Initial value:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 19	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
18 to 16	SPNDL[2:0]	111	R/W	Next Access Delay Sets the period from transfer end to next transfer start (next access). 000: 1 SPBCLK cycle 001: 2 SPBCLK cycles 010: 3 SPBCLK cycles 011: 4 SPBCLK cycles 100: 5 SPBCLK cycles 101: 6 SPBCLK cycles 110: 7 SPBCLK cycles 111: 8 SPBCLK cycles
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 8	SLNDL[2:0]	111	R/W	SPBSSL Negation Delay Sets the period from the time the last SPBCLK edge is sent of a transfer to SPBSSL pin negation (SPBSSL negation delay). 000: 1.5 SPBCLK cycles 001: 2.5 SPBCLK cycles 010: 3.5 SPBCLK cycles 011: 4.5 SPBCLK cycles 100: 5.5 SPBCLK cycles 101: 6.5 SPBCLK cycles 110: 7.5 SPBCLK cycles 111: 8.5 SPBCLK cycles
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	SCKDL[2:0]	111	R/W	Clock Delay Sets the period from SPBSSL pin assertion to SPBCLK oscillation (clock delay). 000: 1 SPBCLK cycle 001: 2 SPBCLK cycles 010: 3 SPBCLK cycles 011: 4 SPBCLK cycles 100: 5 SPBCLK cycles 101: 6 SPBCLK cycles 110: 7 SPBCLK cycles 111: 8 SPBCLK cycles

chanhvinh.com

17.4.3 Bit Rate Register (SPBCR)

SPBCR is a 32-bit register that sets the bit rate.

The settings of this register are reflected both in external address space read mode and SPI operating mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPBR[7:0]								-	-	-	-	-	-	BRDV[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 8	SPBR[7:0]	All 0	R/W	Bit Rate Sets the bit rate. The bit rate is determined by a combination of these bits with the BRDV[1:0] bits. For details, see Table 17.3, Relationship between SPBR[7:0] and BRDV[1:0] Settings.
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	BRDV[1:0]	11	R/W	Bit Rate Frequency Division Sets the bit rate. The bit rate is determined by a combination of these bits with the SPBR[7:0] bits. The SPBR value is used to set the base bit rate. The BRDV value is used to select a division ratio of the base bit rate from among no division, 2, 4, and 8. 00: Base bit rate 01: Base bit rate divided by 2 10: Base bit rate divided by 4 11: Base bit rate divided by 8

chanhvinh.com

(1) Bit Rate

SPBR[7:0] and BRDV[1:0] are used for setting the bit rate.

The following formula is used to calculate the bit rate when SPBR[7:0] ≠ 0.

Bit rate = Bϕ / (2 × n × 2^N)
n: SPBR[7:0] setting (1, …, 255)
N: BRDV[1:0] setting (0 to 3)

Also the following formula is used to calculate the bit rate when SPBR[7:0] = 0.

Bit rate = Bϕ /2^N
N: BRDV[1:0] setting (0 to 3)

Table 17.3 Relationship between SPBR[7:0] and BRDV[1:0] Settings

			Bit Rate		
SPBR[7:0] (n)	**BRDV[1:0] (N)**	**Division Ratio**	**Bϕ = 100 MHz**	**Bϕ = 128 MHz**	**Bϕ = 133.33 MHz**
0	0	1	Setting prohibited		
1	0	2	50 Mbps	64 Mbps	66.67 Mbps
2	0	4	25 Mbps	32 Mbps	33.33 Mbps
3	0	6	16.67 Mbps	21.33 Mbps	22.22 Mbps
4	0	8	12.5 Mbps	16 Mbps	16.67 Mbps
5	0	10	10 Mbps	12.8 Mbps	13.33 Mbps
6	0	12	8.33 Mbps	10.67 Mbps	11.11 Mbps
6	1	24	4.17 Mbps	5.33 Mbps	5.56 Mbps
6	2	48	2.08 Mbps	2.67 Mbps	2.78 Mbps
6	3	96	1.04 Mbps	1.33 Mbps	1.39 Mbps
255	3	4080	24.51 Kbps	31.37 Kbps	32.68 Kbps

Note: The bit rate should be set so that it will satisfy the AC characteristics of this module.

chanhvinh.com

17.4.4 Data Read Control Register (DRCR)

DRCR is a 32-bit register that sets the operation in external address space read mode.
The bits except the SSLN bit should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	SSLN	-	-	-	-	RBURST[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	W	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	RCF	RBE	-	-	-	-	-	-	-	SSLE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	W	R/W	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	SSLN	0	W	SPBSSL Negation Asserted SPBSSL can be negated by writing 1 to this bit when both the RBE and SSLE bits are 1. This bit is always read as 0. Note: To start next access after SPBSSL negation using this bit, read SSLF in CMNSR = 0 to confirm that the SPBSSL has been negated.
23 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19 to 16	RBURST [3:0]	0000	R/W	Read Data Burst Length Sets the burst length (data unit count) when reading. This bit is enabled when the RBE bit is set to 1. 0000: 1 data unit 0001: 2 continuous data units : 1110: 15 continuous data units 1111: 16 continuous data units One data unit is 64 bits long.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	RCF	0	W	Read Cache Flush When 1 is written to this bit, all the entries in the read cache are cleared. This bit is always read as 0. Note: After flushing the read cache by writing 1 to the RCF bit, read the DRCR before proceeding to read from the external address space.
8	RBE	0	R/W	Read Burst Turns burst ON or OFF when reading. 0: Data is read according to the access size. 1: Read cache is enabled, and as many data units as the burst count specified in RBURST[3:0] bits is read.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SSLE	0	R/W	SPBSSL Negation Sets the conditions for SPBSSL negation during read burst. SPBSSL is negated for each access during normal read. 0: SPBSSL is negated after transfer of data set in burst length. 1: SPBSSL is negated when the accessed address is not continuous with the previously transferred address.

chanhvinh.com

17.4.5 Data Read Command Setting Register (DRCMR)

DRCMR is a 32-bit register that sets the commands issued in external address space read mode.
The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	CMD[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	OCMD[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	CMD[7:0]	H'03	R/W	Command Sets the command.
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	OCMD[7:0]	H'00	R/W	Optional Command Sets the optional command.

chanhvinh.com

17.4.6 Data Read Extended Address Setting Register (DREAR)

DREAR is a 32-bit register that sets the address when the serial flash address is output in 32-bit mode.
The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	EAV[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

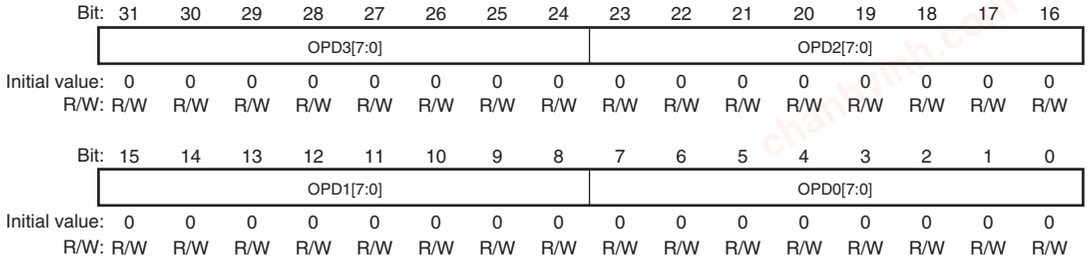
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	EAC[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	EAV[7:0]	H'00	R/W	32-Bit Extended Upper Address Fixed Value Sets the upper address bit values of the external address specified by the EAC[2:0] bits when the serial flash address is output in 32-bit mode. Bit 0 corresponds to the serial flash address bit [25], and bit 7 corresponds to the bit [32]. This setting is valid when the ADE[3] bit in DRENR is 1. When EAC[2:0] are 000, serial flash address [32:25] fixed values should be set to EAV[7:0]. When EAC[2:0] are 001, serial flash address [32:26] fixed values should set to EAV[7:1]. (1) When BSZ[1:0] in CMNCR = 00 (one serial flash memory connected) Serial flash addresses [31:0] are used for accessing. (2) When BSZ[1:0] in CMNCR = 01 (two serial flash memories connected) Serial flash addresses [32:1] are used for accessing.
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	EAC[2:0]	000	R/W	32-Bit Extended External Address Valid Range Sets the range of the external address to be used as serial flash address when the serial flash address is output in 32-bit mode. This setting is valid when the ADE[3] bit in DRENR is 1. 000: External address bits [24:0] enabled 001: External address bits [25:0] enabled Other than above: Setting prohibited

chanhvinh.com

17.4.7 Data Read Option Setting Register (DROPR)

DROPR is a 32-bit register that sets the option data in external address space read mode.
The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.



Bit	Bit Name	Initial Value	R/W	Description
31 to 24	OPD3[7:0]	H'00	R/W	Option Data 3 Sets the option data 3.
23 to 16	OPD2[7:0]	H'00	R/W	Option Data 2 Sets the option data 2.
15 to 8	OPD1[7:0]	H'00	R/W	Option Data 1 Sets the option data 1.
7 to 0	OPD0[7:0]	H'00	R/W	Option Data 0 Sets the option data 0.

Note: OPD3, OPD2, OPD1, and OPD0 are output in this order.

chanhvinh.com

17.4.8 Data Read Enable Setting Register (DRENR)

DRENR is a 32-bit register that sets the bit size of the command, optional command, address, option data, and read data in external address space read mode and enables outputting them other than read data.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CDB[1:0]		OCDB[1:0]		-	-	ADB[1:0]		-	-	OPDB[1:0]		-	-	DRDB[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DME	CDE	-	OCDE	ADE[3:0]				OPDE[3:0]				-	-	-	-
Initial value:	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	CDB[1:0]	00	R/W	Command Bit Size Sets the command size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
29, 28	OCDB[1:0]	00	R/W	Optional Command Bit Size Sets the optional command size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
27, 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25, 24	ADB[1:0]	00	R/W	Address Bit Size Sets the address size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
23, 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21, 20	OPDB[1:0]	00	R/W	Option Data Bit Size Sets the option data size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	DRDB[1:0]	00	R/W	Data Read Bit Size Sets the data read size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
15	DME	0	R/W	Dummy Cycle Enable Enables insertion of the dummy cycle before the read data. Note: A setting is prohibited for a transfer starting with a dummy cycle. 0: Dummy cycle insertion disabled 1: Dummy cycle insertion enabled

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
14	CDE	1	R/W	Command Enable Sets the command to be output. 0: Command output disabled 1: Command output enabled
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	OCDE	0	R/W	Optional Command Enable Sets the optional command to be output. 0: Optional command output disabled 1: Optional command output enabled
11 to 8	ADE[3:0]	0111	R/W	Address Enable Sets the address to be output. Be sure to use the following setting; otherwise, the operation is not guaranteed. (1) BSZ[1:0] in CMNCR = 00 (one serial flash memory connected) 0000: Output disabled 0111: Address[23:0] 1111: Address[31:0] Other than above: Setting prohibited (2) BSZ[1:0] in CMNCR = 01 (two serial flash memories connected) 0000: Output disabled 0111: Address[24:1] 1111: Address[32:1] Other than above: Setting prohibited
7 to 4	OPDE[3:0]	0000	R/W	Option Data Enable Sets the option data to be output. Use only the settings given below. Otherwise, the operation cannot be guaranteed. 0000: Output disabled 1000: OPD3 1100: OPD3, OPD2 1110: OPD3, OPD2, OPD1 1111: OPD3, OPD2, OPD1, OPD0 Other than above: Setting prohibited
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

17.4.9 SPI Mode Control Register (SMCR)

SMCR is a 32-bit register that sets the operation in SPI operating mode.
The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	SSLKP	-	-	-	-	-	SPIRE	SPIWE	SPIE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R/W	R/W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	SSLKP	0	R/W	SPBSSL Signal Level Determines the SPBSSL status after the end of transfer. 0: SPBSSL signal is negated at the end of transfer. 1: SPBSSL signal level is maintained from the end of transfer to the start of next access.
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	SPIRE	0	R/W	Data Read Enable Enables reading in SPI operating mode. 0: Data reading disabled 1: Data reading enabled Note: When the transfer data bit size is set to 2 bits or 4 bits with the SPIDB[1:0] bits, the SPIRE and SPIWE bits should not be set to 1 at the same time.
1	SPIWE	0	R/W	Data Write Enable Enables writing in SPI operating mode. 0: Data writing disabled 1: Data writing enabled Note: When the transfer data bit size is set to 2 bits or 4 bits with the SPIDB[1:0] bits, the SPIRE and SPIWE bits should not be set to 1 at the same time.
0	SPIE	0	W	SPI Data Transfer Enable Data is transferred by setting this bit to 1. This bit is enabled only when the TEND bit in CMNSR is set to 1. The operation cannot be guaranteed when this bit is set to 1 with the TEND bit set to 0. This bit is always read as 0. Note: When the SPBSSL pin is de-asserted, the command, optional command, address, and option data that are output enabled are output even if the SPIRE and SPIWE bits are set to 0. When the SPBSSL pin is asserted, follow the notes described in section 17.6.2, Notes on Starting Transfer from the SPBSSL Retained State in SPI Operating Mode.

chanhvinh.com

17.4.10 SPI Mode Command Setting Register (SMCMR)

SMCMR is a 32-bit register that sets the commands issued in SPI operating mode.
The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	CMD[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	OCMD[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	CMD[7:0]	H'00	R/W	Command Sets the command.
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	OCMD[7:0]	H'00	R/W	Optional Command Sets the optional command.

17.4.11 SPI Mode Address Setting Register (SMADR)

SMADR is a 32-bit register that sets the addresses in SPI operating mode.
The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADR[31:24]								ADR[23:16]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADR[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	ADR[31:24]	H'00	R/W	Address Sets the value of bits 31 to 24 when the serial flash address is output in 32-bit units. This setting is valid when ADE[3] in SMENR is 1.
23 to 0	ADR[23:0]	H'000000	R/W	Address Sets the address.

chanhvinh.com

17.4.12 SPI Mode Option Setting Register (SMOPR)

SMOPR is a 32-bit register that sets the option data in SPI operating mode.
The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OPD3[7:0]								OPD2[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OPD1[7:0]								OPD0[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	OPD3[7:0]	H'00	R/W	Option Data 3 Sets the option data 3.
23 to 16	OPD2[7:0]	H'00	R/W	Option Data 2 Sets the option data 2.
15 to 8	OPD1[7:0]	H'00	R/W	Option Data 1 Sets the option data 1.
7 to 0	OPD0[7:0]	H'00	R/W	Option Data 0 Sets the option data 0.

Note: OPD3, OPD2, OPD1, and OPD0 are output in this order.

chanhvinh.com

17.4.13 SPI Mode Enable Setting Register (SMENR)

SMENR is a 32-bit register that sets the bit size of the command, optional command, address, option data, and transfer data in SPI operating mode and enables their output. SMENR also enables dummy cycle insertion. Disabling all of the command, optional command, address, option data, dummy cycle, and transfer data is prohibited. At least one of them except dummy cycle must be enabled.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CDB[1:0]		OCDB[1:0]		-	-	ADB[1:0]		-	-	OPDB[1:0]		-	-	SPIDB[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DME	CDE	-	OCDE	ADE[3:0]				OPDE[3:0]				SPIDE[3:0]			
Initial value:	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	CDB[1:0]	00	R/W	Command Bit Size Sets the command size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
29, 28	OCDB[1:0]	00	R/W	Optional Command Bit Size Sets the optional command size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
27, 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25, 24	ADB[1:0]	00	R/W	Address Bit Size Sets the address size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
23, 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21, 20	OPDB[1:0]	00	R/W	Option Data Bit Size Sets the option data size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	SPIDB[1:0]	00	R/W	Transfer Data Bit Size Sets the transfer data size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
15	DME	0	R/W	Dummy Cycle Enable Enables insertion of the dummy cycle before the read data. Note: Dummy cycle insertion is prohibited for write in SPI operating mode including the case in which a transfer ends with a dummy cycle. Note: A setting is prohibited for a transfer starting with a dummy cycle. 0: Dummy cycle insertion disabled 1: Dummy cycle insertion enabled
14	CDE	1	R/W	Command Enable Sets the command to be output. 0: Command output disabled 1: Command output enabled
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	OCDE	0	R/W	Optional Command Enable Sets the optional command to be output. 0: Optional command output disabled 1: Optional command output enabled
11 to 8	ADE[3:0]	0000	R/W	Address Enable Sets the address to be output. Use only the settings given below. Otherwise, the operation cannot be guaranteed. 0000: Output disabled 0100: ADR[23:16] 0110: ADR[23:8] 0111: ADR[23:0] 1111: ADR[31:0] Other than above: Setting prohibited
7 to 4	OPDE[3:0]	0000	R/W	Option Data Enable Sets the option data to be output. Use only the settings given below. Otherwise, the operation cannot be guaranteed. 0000: Output disabled 1000: OPD3 1100: OPD3, OPD2 1110: OPD3, OPD2, OPD1 1111: OPD3, OPD2, OPD1, OPD0 Other than above: Setting prohibited
3 to 0	SPIDE[3:0]	0000	R/W	Transfer Data Enable Sets valid transfer data. Valid data differs depending on the BSZ[1:0] bit setting in CMNCR. The following settings must be used. Otherwise, the operation is not guaranteed. (1) BSZ[1:0] bits in CMNCR = 00 (one serial flash memory connected) 0000: Not transferred 1000: 8 bits transferred (enables data at address 0 of the SPI mode read/write data registers 0) 1100: 16 bits transferred (enables data at addresses 0 and 1 of the SPI mode read/write data registers 0) 1111: 32 bits transferred (enables data at addresses 0 to 3 of the SPI mode read/write data registers 0) Other than above: Setting prohibited (2) BSZ[1:0] bits in CMNCR = 01 (two serial flash memories connected) 0000: Not transferred 1000: 16 bits transferred (enables data at addresses 0 and 1 of the SPI mode read/write data registers 0) 1100: 32 bits transferred (enables data at addresses 0 to 3 of the SPI mode read/write data registers 0) 1111: 64 bits transferred (enables data at addresses 0 to 3 of the SPI mode read/write data registers 0 and data at addresses 0 to 3 of the SPI mode read/write data registers 1) Other than above: Setting prohibited

chanhvinh.com

17.4.14 SPI Mode Read Data Register 0 (SMRDR0)

SMRDR0 is a 32-bit register that stores the read data in SPI operating mode.
Access to this register should be performed in the same size as the transfer size specified in the SPIDE[3:0] bits in the SPI mode enable setting register (SMENR). Be sure to access from address 0.
The settings of this register should be read when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDATA0[31:16]															
Initial value:	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDATA0[15:0]															
Initial value:	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RDATA0 [31:0]	Undefined	R	Read Data Holds the data read in SPI operating mode. Data bits differ depending on the settings of SFDE and BSZ[1:0] bits in CMNCR and SPIDE[3:0] bits in SMENR. BSZ[1:0] = 01, SPIDE[3:0] = 1111, SFDE = 1: Read data[63:32]. BSZ[1:0] = 01, SPIDE[3:0] = 1111, SFDE = 0: Read data[31:0]. Other than the above: Read data[31:0].

Note: The contents of this register and SMRDR1 are modified upon completion of reception in SPI operating mode. Be sure to read data when reception in SPI operating mode is completed.

17.4.15 SPI Mode Read Data Register 1 (SMRDR1)

SMRDR1 is a 32-bit register that stores the read data in SPI operating mode.
This register is enabled when the BSZ[1:0] bits in CMNCR are set to 01 (two serial flash memories connected) and disabled when the BSZ[1:0] bits in CMNCR are set to 00 (one serial flash memory connected).
Access to this register should be performed in the same size as the transfer size specified in the SPIDE[3:0] bits in the SPI mode enable setting register (SMENR). Be sure to access from address 0.
The settings of this register should be read when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDATA1[31:16]															
Initial value:	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDATA1[15:0]															
Initial value:	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RDATA1 [31:0]	Undefined	R	Read Data Holds the data read in SPI operating mode. Data bits differ depending on the settings of SFDE and BSZ[1:0] bits in CMNCR and SPIDE[3:0] bits in SMENR. BSZ[1:0] = 01, SPIDE[3:0] = 1111, SFDE = 1: Read data[31:0]. BSZ[1:0] = 01, SPIDE[3:0] = 1111, SFDE = 0: Read data[63:32]. Other than the above: Bits in this register are disabled.

chanhvinh.com

17.4.16 SPI Mode Write Data Register 0 (SMWDR0)

SMWDR0 is a 32-bit register that sets the write data in SPI operating mode.
Access to this register should be performed in the same size as the transfer size specified in the SPIDE[3:0] bits in the SPI mode enable setting register (SMENR). Be sure to access from address 0.
The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WDATA0[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDATA0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	WDATA0 [31:0]	All 0	R/W	Write Data Holds the data to be written in SPI operating mode. Data bits differ depending on the settings of SFDE and BSZ[1:0] bits in CMNCR and SPIDE[3:0] bits in SMENR. BSZ[1:0] = 01, SPIDE[3:0] = 1111, SFDE = 1: Write data[63:32]. BSZ[1:0] = 01, SPIDE[3:0] = 1111, SFDE = 0: Write data[31:0]. Other than the above: Write data[31:0].

17.4.17 SPI Mode Write Data Register 1 (SMWDR1)

SMWDR1 is a 32-bit register that sets the write data in SPI operating mode.
This register is enabled when the BSZ[1:0] bits in CMNCR are set to 01 (two serial flash memories connected) and disabled when the BSZ[1:0] bits in CMNCR are set to 00 (one serial flash memory connected).
Access to this register should be performed in the same size as the transfer size specified in the SPIDE[3:0] bits in the SPI mode enable setting register (SMENR). Be sure to access from address 0.
The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WDATA1[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDATA1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	WDATA1 [31:0]	All 0	R/W	Write Data Holds the data to be written in SPI operating mode. Data bits differ depending on the settings of SFDE and BSZ[1:0] bits in CMNCR and SPIDE[3:0] bits in SMENR. BSZ[1:0] = 01, SPIDE[3:0] = 1111, SFDE = 1: Write data[31:0]. BSZ[1:0] = 01, SPIDE[3:0] = 1111, SFDE = 0: Write data[63:32]. Other than the above: Bits in this register are disabled.

chanhvinh.com

17.4.18 Common Status Register (CMNSR)

CMNSR is a 32-bit register that holds flags indicating the operating state.

The settings of this register are reflected both in external address space read mode and SPI operating mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SSLF	TEND
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	SSLF	0	R	SPBSSL Pin Monitor 0: SPBSSL pin is negated 1: SPBSSL pin is asserted
0	TEND	1	R	Transfer End Flag Indicates whether the data transfer has ended. 0: Indicates that data transfer is in progress 1: Indicates that data transfer has ended

chanhvinh.com

17.4.19 SPI AC Input Characteristics Adjustment Register (CKDLY) (RZ/A1LU only)

CKDLY is used to adjust the timing of the setup and hold times for data input. The timing should be adjusted to suit the AC characteristics of the serial flash memory to be connected. Settings of this register should be changed while the SSLF flag in CMNSR is 0; otherwise, the operation cannot be guaranteed. When writing, write to the register as a 32-bit unit with bits 15 to 8 set to H'A5.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GB[7:0]								-	-	-	-	CKDLY[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W:	W	W	W	W	W	W	W	W	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 8	GB[7:0]	All 0	W	Guard When writing, write to the register as a 32-bit unit with these bits set to H'A5. These bits are always read as 0.
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	CKDLY[3:0]	0100	R/W	Input Characteristics Adjustment Switches the relative timing of the setup and hold times for data input. The two values below are specifiable. 0100: Initial value 1010: Makes the data input setup time shorter and the data input hold time longer. Other than the above: Settings prohibited

chanhvinh.com

17.4.20 Data Read Dummy Cycle Setting Register (DRDMCR)

DRDMCR is a 32-bit register that sets the size and number of dummy cycles to be inserted in external address space read mode.

The settings of this register are enabled when the DME bit in the data read enable setting register (DRENR) is 1.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DMDB[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	DMCYC[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	DMDB [1:0]	00	R/W	Dummy Cycle Bit Size Sets the dummy cycle size in bit units. The setting of these bits is combined with the setting of the IO0FV, IO2FV, and IO3FV bits in the common control register (CMNCR) to determine the state of the unused pins during the dummy cycles. The state of the used pins is Hi-Z. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	DMCYC [2:0]	000	R/W	Number of Dummy Cycles Sets the number of dummy cycles to be inserted when the DME bit in the data read enable setting register (DRENR) is 1. 000: 1 cycle 001: 2 cycles 010: 3 cycles 011: 4 cycles 100: 5 cycles 101: 6 cycles 110: 7 cycles 111: 8 cycles

chanhvinh.com

17.4.21 Data Read DDR Enable Register (DRDRENR) (RZ/A1LU only)

DRDRENR is a 32-bit register that sets SDR or DDR transfer of the address, option data, and read data in external address space read mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	ADDRE	-	-	-	OPDRE	-	-	-	DRDRD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	ADDRE	0	R/W	Address DDR Enable Sets SDR or DDR transfer of the address. 0: SDR transfer 1: DDR transfer
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	OPDRE	0	R/W	Option Data DDR Enable Sets SDR or DDR transfer of the option data. 0: SDR transfer 1: DDR transfer
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	DRDRE	0	R/W	Data Read DDR Enable Sets SDR or DDR transfer of the read data. 0: SDR transfer 1: DDR transfer

chanhvinh.com

17.4.22 SPI Mode Dummy Cycle Setting Register (SMDMCR)

SMDMCR is a 32-bit register that sets the size and number of dummy cycles to be inserted in SPI operating mode.
The settings of this register are enabled when the DME bit in the SPI mode enable setting register (SMENR) is 1.
The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DMDB[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	DMCYC[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17 and 16	DMDB [1:0]	00	R/W	Dummy Cycle Bit Size Sets the dummy cycle size in bit units. The setting of these bits is combined with the setting of the IO0FV, IO2FV, and IO3FV bits in the common control register (CMNCR) to determine the state of the unused pins during the dummy cycles. The state of the used pins is Hi-Z. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	DMCYC[2:0]	000	R/W	Number of Dummy Cycles Sets the number of dummy cycles to be inserted when the DME bit in the SPI mode enable setting register (SMENR) is 1. 000: 1 cycle 001: 2 cycles 010: 3 cycles 011: 4 cycles 100: 5 cycles 101: 6 cycles 110: 7 cycles 111: 8 cycles

chanhvinh.com

17.4.23 SPI Mode DDR Enable Register (SMDRENR) (RZ/A1LU only)

SMDRENR is a 32-bit register that sets SDR or DDR transfer of the address, option data, and transfer data in SPI operating mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	ADDRE	-	-	-	OPDRE	-	-	-	SPIDRE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	ADDRE	0	R/W	Address DDR Enable Sets SDR or DDR transfer of the address. 0: SDR transfer 1: DDR transfer
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	OPDRE	0	R/W	Option Data DDR Enable Sets SDR or DDR transfer of the option data. 0: SDR transfer 1: DDR transfer
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SPIDRE	0	R/W	Transfer Data DDR Enable Sets SDR or DDR transfer of the transfer data. 0: SDR transfer 1: DDR transfer

chanhvinh.com

17.4.24 SPI AC Output Characteristics Adjustment Register (SPODLY) (RZ/A1LU only)

SPODLY is used to adjust the timing of the delay, hold, buffer on and buffer off times for data output. The timing should be adjusted to suit the AC characteristics of the serial flash memory to be connected. Settings of this register should be changed while the SSLF flag in CMNSR is 0; otherwise, the operation cannot be guaranteed. When writing, write to the register as a 32-bit unit with bits 31 to 24 set to H'A5.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GB[7:0]								-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPODLY[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GB[7:0]	All 0	W	Guard When writing, write to the register as a 32-bit unit with these bits set to H'A5. These bits are always read as 0.
23 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	SPODLY[15:0]	All 0	R/W	Output Characteristics Adjustment Switches the timing of the delay, hold, buffer on and buffer off times for data output. The two values below are specifiable. H'0000: Initial value H'1111: The delay, hold, buffer on and buffer off times for data output are lengthened. Other than the above: Settings prohibited

chanhvinh.com

17.5 Operation

17.5.1 System Configuration

With this module, one or two serial flash memories can be directly connected per channel (data size of 1, 2, and 4 bits). The number of connected memories can be selected using the BSZ[1:0] bits in CMNCR.

Examples of system configuration with one serial flash memory connected and two serial flash memories connected are shown in Figure 17.2 and Figure 17.3, respectively.

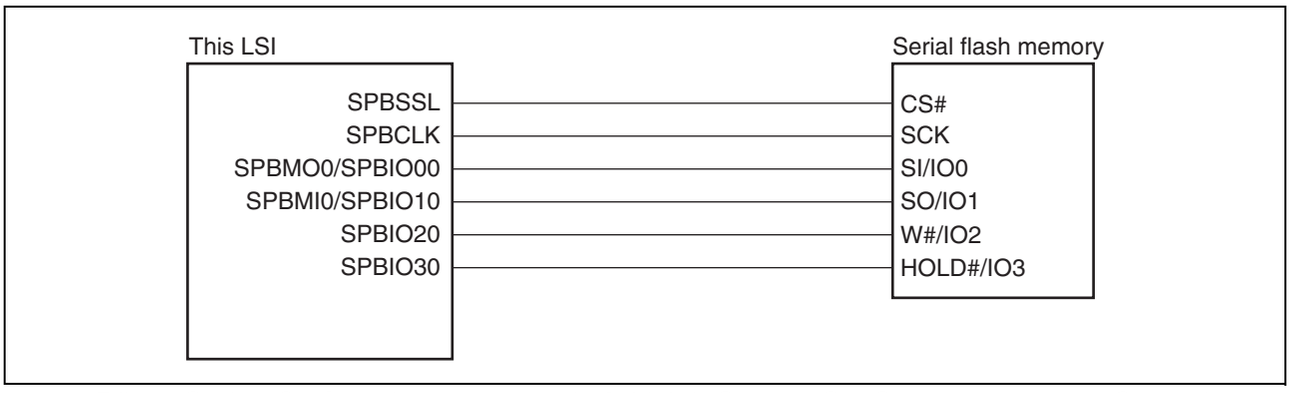


Figure 17.2 System Configuration Example with 4-Bit Data Size and One Serial Flash Memory Connected (BSZ[1:0] Bits in CMNCR = 00)

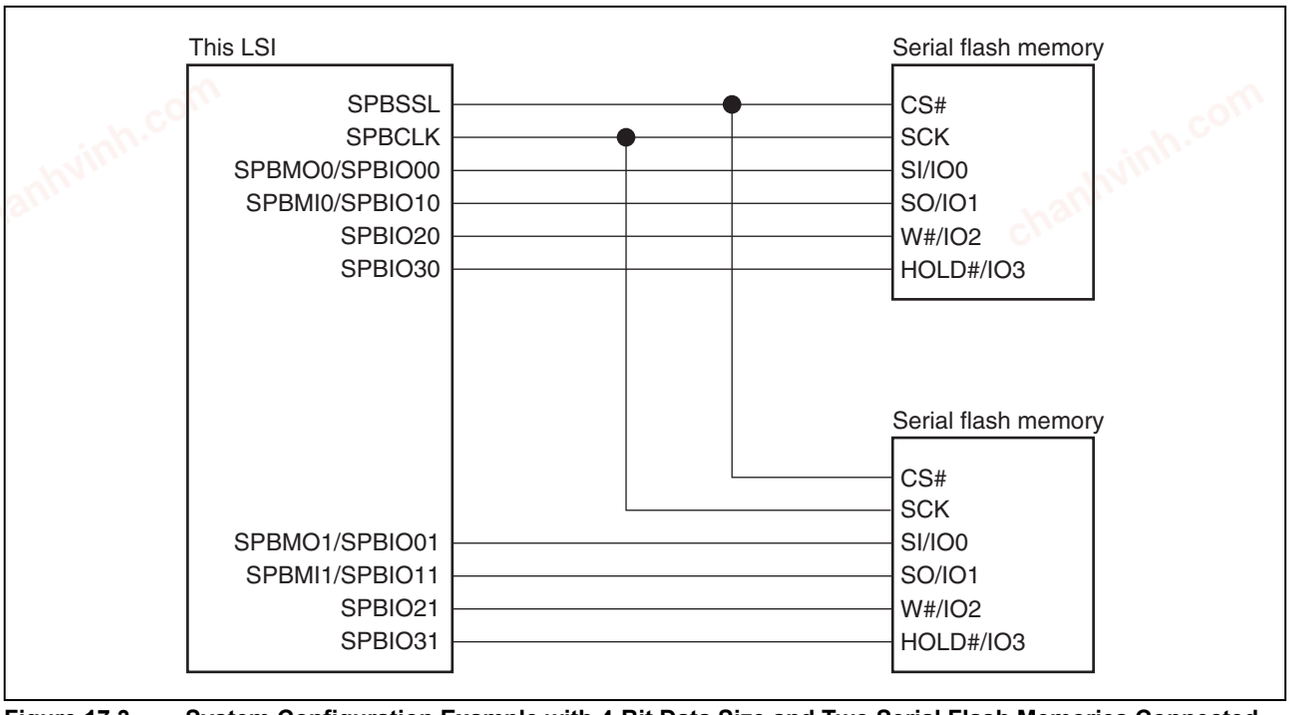


Figure 17.3 System Configuration Example with 4-Bit Data Size and Two Serial Flash Memories Connected (BSZ[1:0] Bits in CMNCR = 01)

chanhvinh.com

17.5.2 Address Map

In external address space read mode, the serial flash connected is assigned in the SPI multi I/O bus space. A maximum accessible address space differs depending on the number of serial flash memories connected. In combination with DREAR, a maximum of 4 Gbytes can be accessed when one serial flash memory is connected, and a maximum of 8 Gbytes can be accessed when two memories are connected.

Table 17.4 Address Map

Channel	Number of Serial Flash Memories Connected	Internal Address	Max. Access Area
0	1	H'18000000 to H'1BFFFFFF	4 Gbytes
		H'58000000 to H'5BFFFFFF (mirror area)	
	2	H'18000000 to H'1BFFFFFF	8 Gbytes
		H'58000000 to H'5BFFFFFF (mirror area)	

17.5.3 32-bit Serial Flash Addresses

Since the SPI multi I/O bus space is 64 Mbytes, only a part of the 32-bit serial flash address area can be directly accessed. Here, the fixed value set in the pertinent register is used as the upper bit value of a 32-bit address.
To output serial flash addresses in 32 bits, set the ADE[3] bit in DRENR to 1, set the range of the external addresses used as the serial flash addresses to the EAC[2:0] bits in DREAR, and set the upper bit value of the 32-bit address as the fixed value to the EAV[7:0] bits in DREAR.

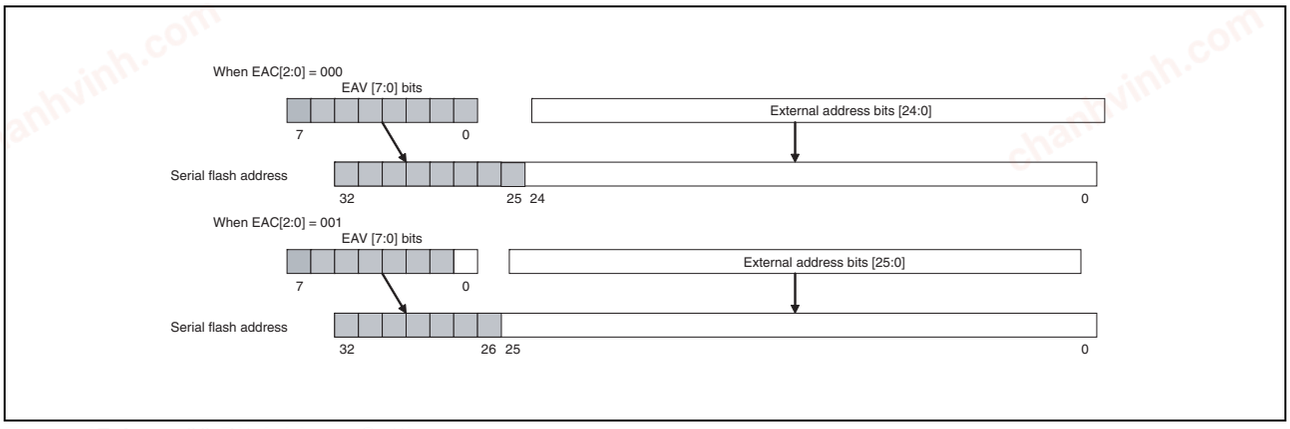


Figure 17.4 32-Bit Address Setting

Setting the ADE[3] bit in DRENR to 1 allows the serial flash address to be output using [31:0] bits. When EAC[2:0] = 000, external address bits [24:0] are valid; set the value for [32:25] bits to EAV[7:0]. When EAC[2:0] = 001, external address bits [25:0] are valid; set the value for [32:26] bits to EAV[7:1].
The address bits actually used for access depend on the number of serial flash memories connected. When one serial flash memory is connected, address bits [31:0] are used and when two memories are connected, address bits [32:1] are used.

Note: When the capacity of the serial flash memory used is smaller than 4 Gbytes, keep the following point in mind. If an access spreads over the last address of the serial flash in burst mode (RBE bit in DRCR = 1), the access address does not agree with the internal address of the serial flash. To prevent this, software should appropriately manage the accessible address areas for the serial flash memory used according to the memory capacity.

chanhvinh.com

17.5.4 Data Alignment

Data alignment can be set by using the SFDE bit in the common control register (CMNCR). Data alignment in data read mode and in SPI mode are shown in Figure 17.5 and Figure 17.6, respectively.

When two serial flash memories are connected, the serial flash memory connected to the pin SPBIO30-SPBIO00 has the address 2n and the serial flash memory connected to the pin SPBIO31-SPBIO01 has the address 2n + 1. The data should be accessed in word or larger units. It cannot be accessed in byte units.

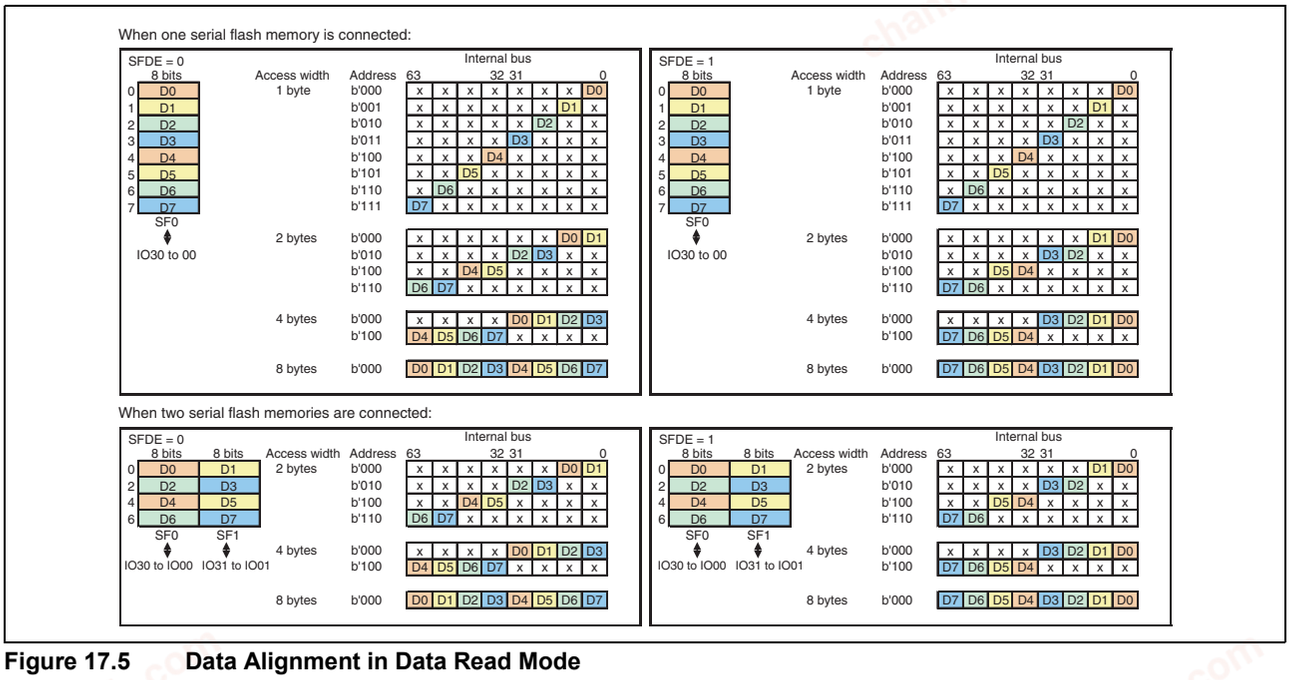


Figure 17.5 Data Alignment in Data Read Mode

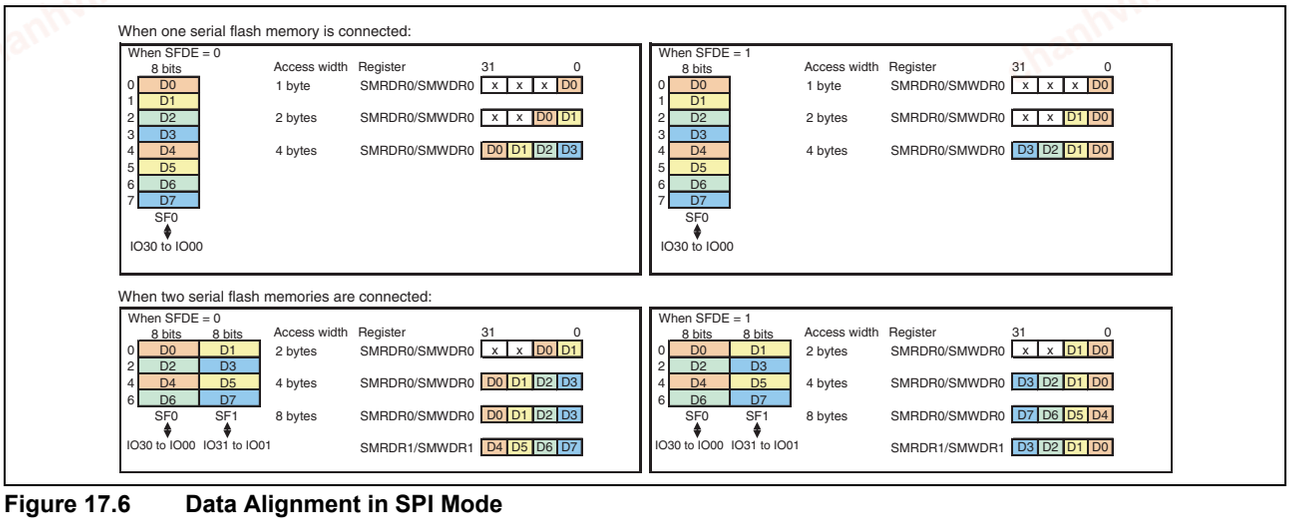


Figure 17.6 Data Alignment in SPI Mode

chanhvinh.com

17.5.5 Operating Modes

This module has two operating modes: external address space read mode and SPI operating mode.
In external address space read mode, a read access to the SPI multi I/O bus space is converted into SPI communication and data is received. After data acquisition, data is returned to the bus master that is the issuing source. For details, see section 17.5.6, External Address Space Read Mode.
In SPI operating mode, arbitrary SPI communication is carried out using register settings. For details, see section 17.5.8, SPI Operating Mode.

17.5.6 External Address Space Read Mode

A read access to the SPI multi I/O bus space can be converted into SPI communication in external address space read mode. Further, the commands, optional commands, option data, and dummy cycle issued for reading can be modified using registers.
For the address, option data, and read data, either SDR or DDR transfer can be selected using the appropriate register when the SPBCLK frequency division ratio is two or larger (RZ/A1LU only).
In external address space read mode, either normal read operation or burst read operation can be selected. The transfer format is determined based on the following registers.
- Common control register (CMNCR)
- SSL delay register (SSLDR)
- Bit rate setting register (SPBCR)
- Data read control register (DRCR)
- Data read command setting register (DRCMR)
- Data read extended address setting register (DREAR)
- Data read option setting register (DROPR)
- Data read enable setting register (DRENR)
- Data read dummy cycle setting register (DRDMCR).
- Data read DDR enable register (DRDRENR)*

Note: * RZ/A1LU only

(1) Normal Read Operation

When the RBE bit in DRCR is set to 0, normal read operation is performed.
In the normal read operation, the data of 8 bits, 16 bits, 32 bits, and 64 bits are read for respectively a byte, a word, and a longword read access. Here, a byte access is enabled only when one serial flash memory is connected. After reading, the SPBSSL pin is negated.
The normal read operation timing is shown in Figure 17.7.
t1 is the time period from SPBSSL pin assertion to SPBCLK oscillation (clock delay), t2 is the time period from transmission of the last SPBCLK edge of a transfer to SPBSSL pin negation (SPBSSL negation delay), and t3 is the time period from one transfer end to the next transfer start (next access). For details of t1, t2, and t3, see section 17.5.9, Transfer Format.

chanhvinh.com

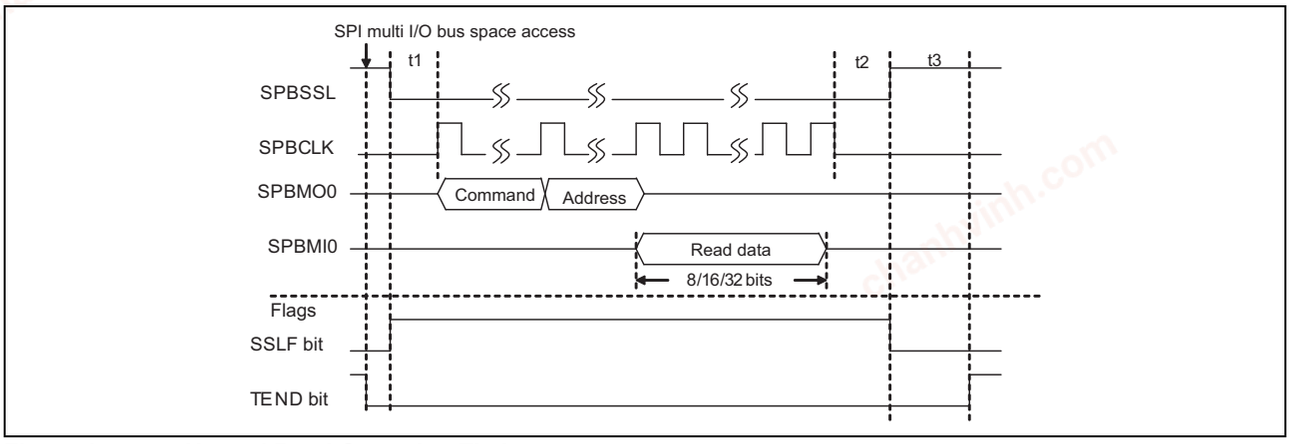
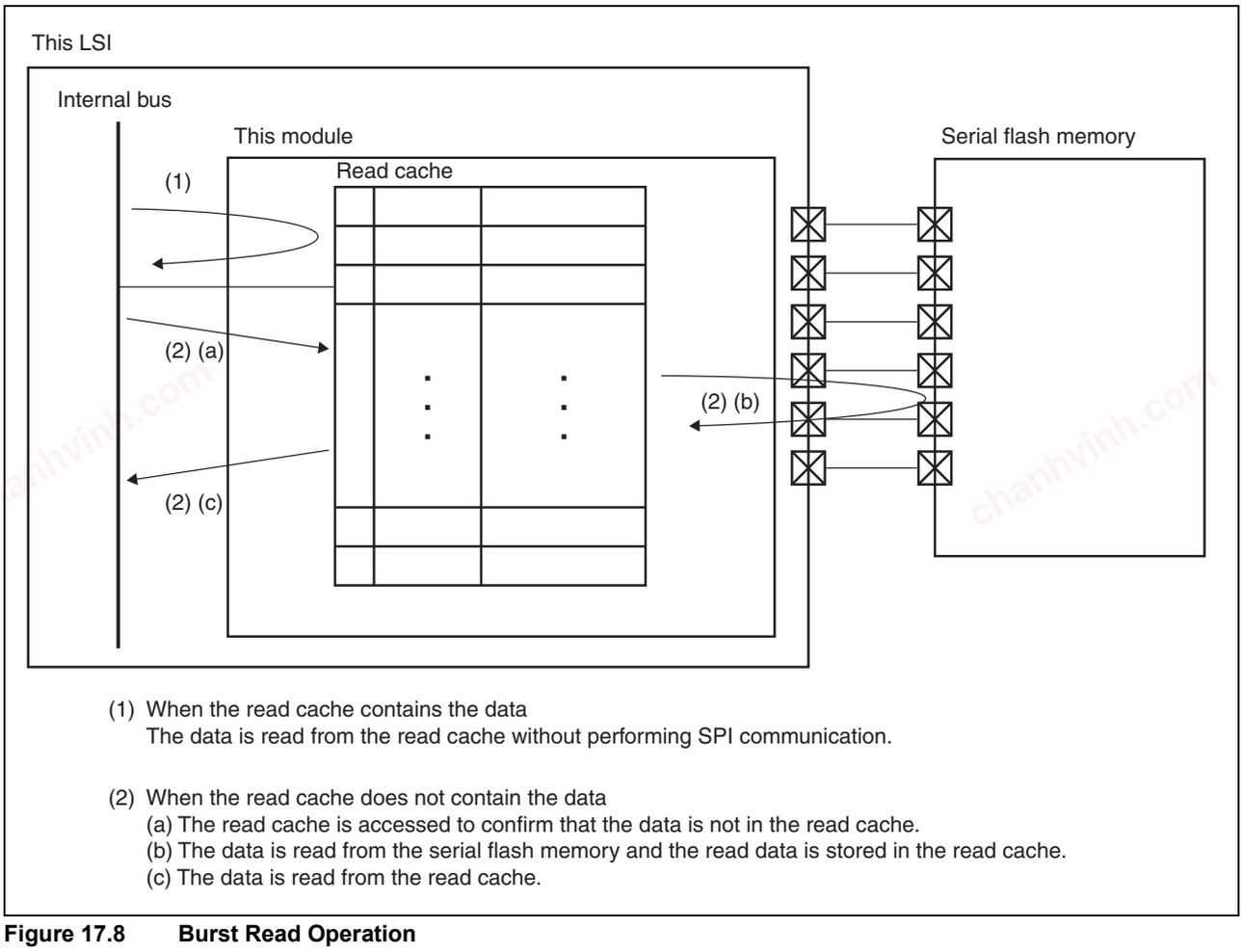


Figure 17.7 Normal Read Operation Timing

chanhvinh.com

(2) Burst Read Operation

When the RBE bit in DRCR is set to 1, burst read operation is performed.
Read cache is enabled in the burst read operation. For read cache operation, see section 17.5.7, Read Cache.
For reading bytes, words, or longwords, the read cache is first referred to for the data. When the read cache contains the data, the data is read from the read cache without accessing the serial flash memory. When the read cache does not contain the data, burst read operation is performed in the serial flash memory and the read data is stored in the read cache. The data transfer length at that time is 64 bits × RBURST[3:0] bits and the data is always read from the 64-bit boundary.
The SPBSSL pin status after data transfer can be selected by using the SSLE bit in DRCR. When the SSLE bit is set to 0, the SPBSSL pin is negated after data transfer. For an operation performed when the SSLE bit is set to 1, see (3) Burst Read Operation with Automatic SPBSSL Negation, just below.
A pattern diagram of this operation and a burst read operation timing diagram when SSLE bit is set to 0 are shown in Figure 17.8 and Figure 17.9.



(1) When the read cache contains the data
The data is read from the read cache without performing SPI communication.

(2) When the read cache does not contain the data
(a) The read cache is accessed to confirm that the data is not in the read cache.
(b) The data is read from the serial flash memory and the read data is stored in the read cache.
(c) The data is read from the read cache.

Figure 17.8 Burst Read Operation

chanhvinh.com

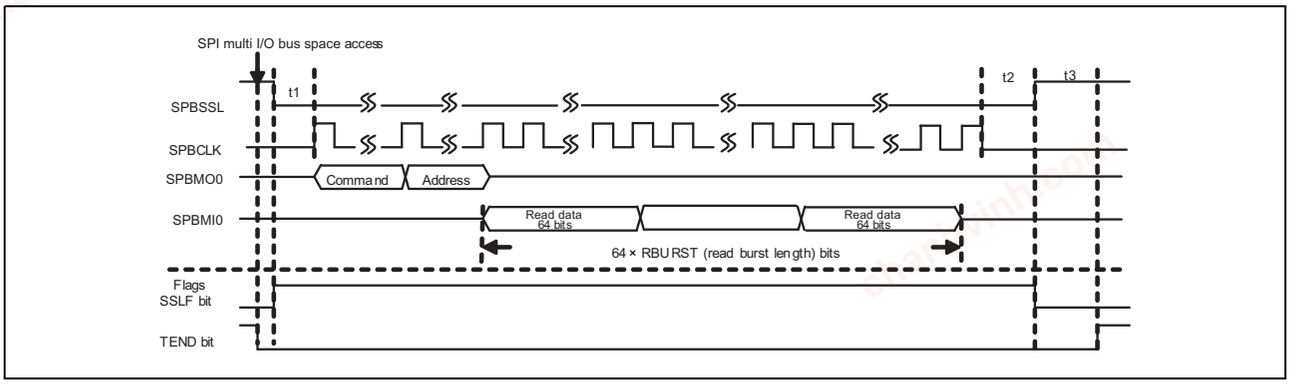


Figure 17.9 Burst Read Operation Timing (SSLE Bit = 0)

(3) Burst Read Operation with Automatic SPBSSL Negation

When SSLE bit in DRCR is set to 1, this module does not negate the SPBSSL pin after the burst read transfer. When accessing the next time, if the address is continuous with the previous read address, the burst read operation is performed without issuing the command, optional command, address, option data, or dummy cycle. If the address is not continuous with the previous read address, the SPBSSL pin is once negated and the burst read operation is performed after issuing the command, optional command, address, option data, or dummy cycle.

Burst read timing diagrams for continuous address and non-continuous address are shown in Figure 17.10 and Figure 17.11.

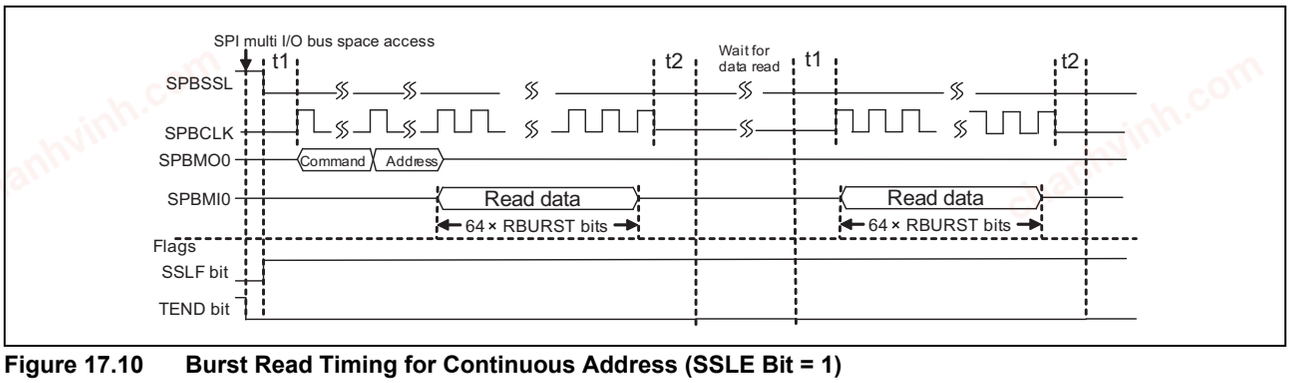


Figure 17.10 Burst Read Timing for Continuous Address (SSLE Bit = 1)

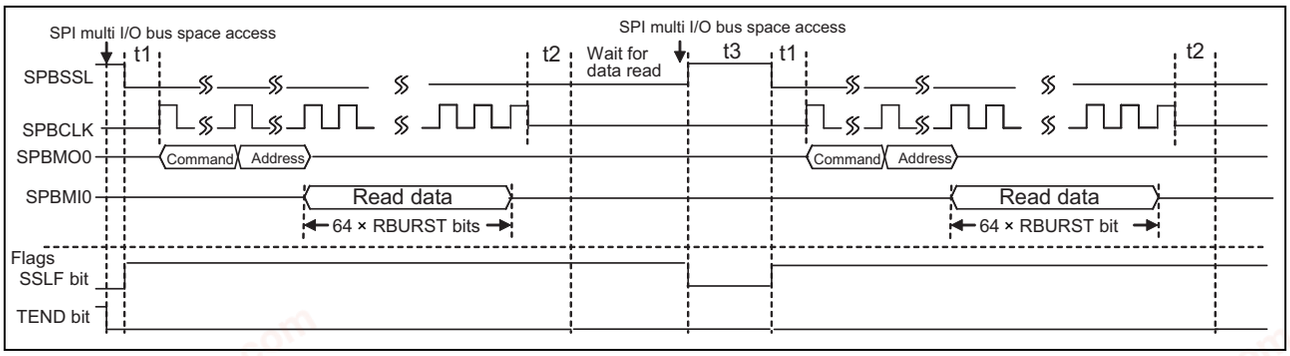


Figure 17.11 Burst Read Timing for Non-Continuous Address (SSLE Bit = 1)

For the next access after negation of the SPBSSL with the SSLN bit in DRCR with this operation, read SSLF = 0 in CMNSR to confirm that the SPBSSL has been negated.

chanhvinh.com

(4) Initial Setting Flow

An example of an initial setting flow in external address space read mode is shown in Figure 17.12.

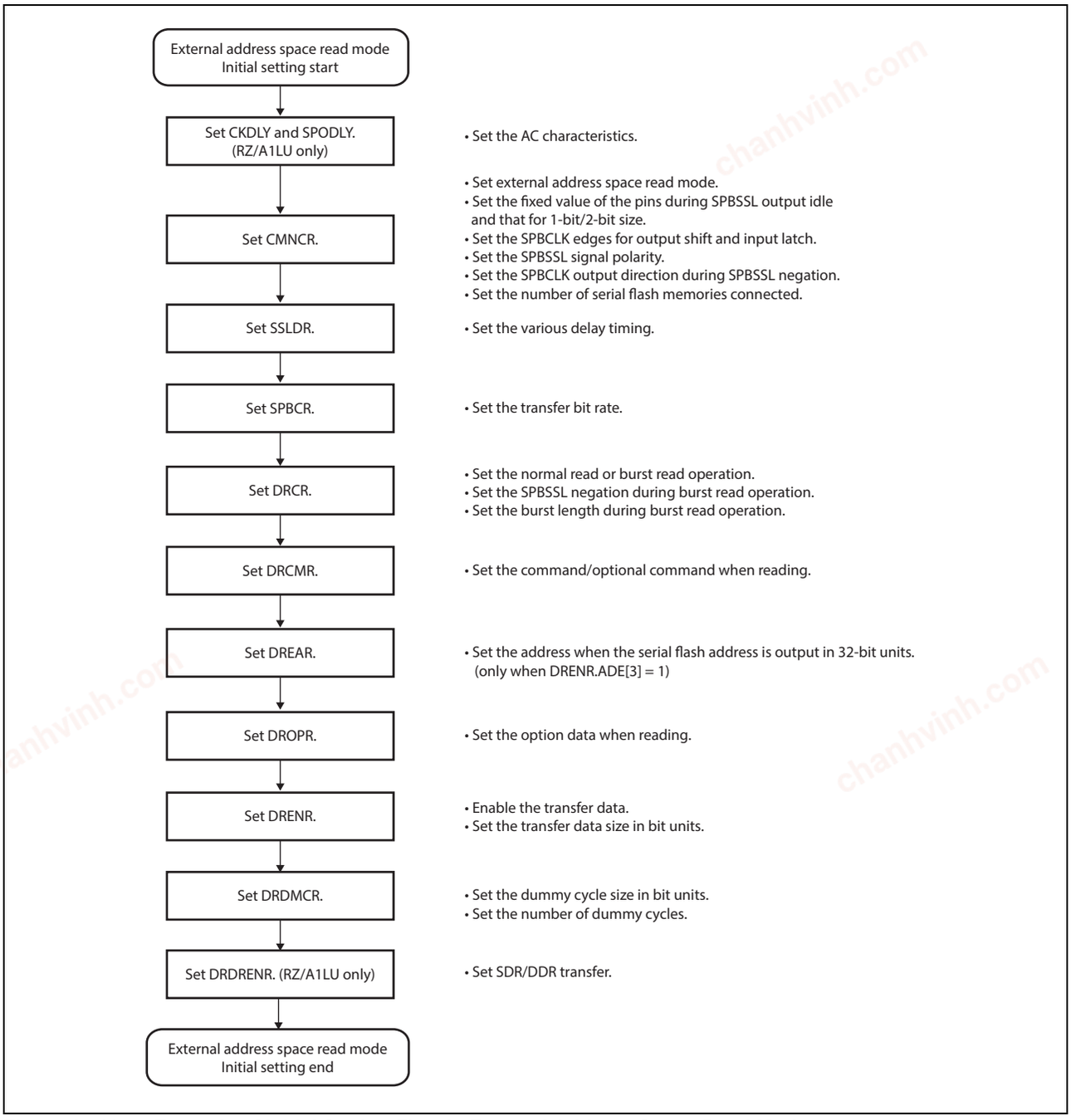


Figure 17.12 Example of Initial Setting Flow in External Address Space Read Mode

chanhvinh.com

17.5.7 Read Cache

This module has a simple built-in read cache. The read cache can be used during external address space read mode and burst read operation. The read cache is configured with a line size of 64 bits and 16 entries.
Read cache configuration is shown in Figure 17.13.

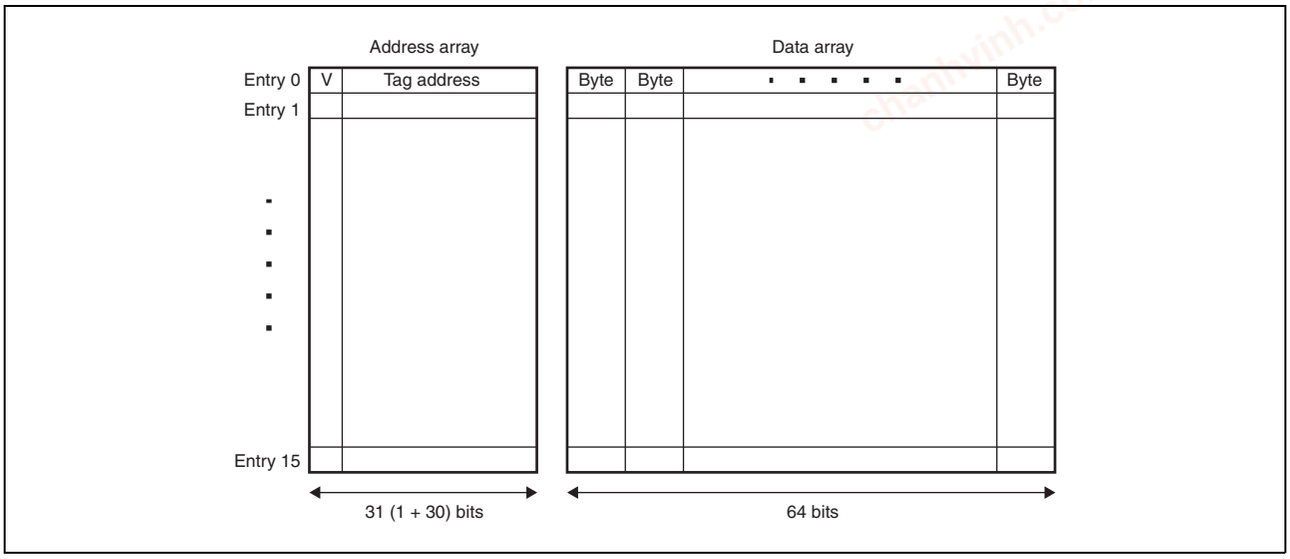


Figure 17.13 Read Cache Configuration

(1) Address Array

The V bit indicates whether the entry data is valid. When the V bit is 1, the data is valid and when V bit is 0, the data is invalid.
The tag address bits hold the address used for the serial flash memory. Address bits 32 to 3 are used for the purpose.
Address bits 23 to 3 are enabled when address output is 24 bits and one serial flash memory is connected; and address bits 24 to 3 are enabled when two serial flash memories are connected.
Address bits 31 to 3 are enabled when address output is 32 bits and one serial flash memory is connected; and address bits 32 to 3 are enabled when two serial flash memories are connected.

(2) Data Array

It retains the 64-bit read data. Registration in the read cache is performed in line units.

(3) Read Operation

In case of read-hit, data is read from the read cache. In case of read-miss, after the 64 × RBURST (read burst length) data is read from the serial flash memory and the read cache is updated, the data is returned to the bus master.

(4) Data Replacement

The write pointer is used to update data. In case of read-miss, the RBURST (read burst length) portion data is replaced starting at the entry specified by the write pointer. In other words, the data is replaced in the storage order of the data.
Whether data is referred to or not will not affect the replacement order of data.

chanhvinh.com

17.5.8 SPI Operating Mode

This module can carry out an arbitrary SPI operation by using the register settings.
The transfer format is determined based on the following registers.
- Common control register (CMNCR)
- SSL delay register (SSLDR)
- Bit rate setting register (SPBCR)
- SPI mode control register (SMCR)
- SPI mode command setting register (SMCMR)
- SPI mode address setting register (SMADR)
- SPI mode option setting register (SMOPR)
- SPI mode enable setting register (SMENR)
- SPI mode read data register (SMRDR)
- SPI mode write data register (SMWDR)
- SPI mode dummy cycle setting register (SMDMCR)
- SPI mode DDR enable register (SMDRENR)*

For the address, option data, and transfer data, either SDR or DDR transfer can be selected using the appropriate register when the SPBCLK frequency division ratio is two or larger (RZ/A1LU only).
SPI operating mode can be used for reading the status of the serial flash memory and writing to the serial flash memory. In this mode, one transfer refers to the operation from when the SPIE bit in SMCR is set to 1 to when the TEND bit is set to 1.
Note: * RZ/A1LU only

(1) Transfer Start

The transfer of data is started in the set transfer format by setting the SPIE bit in SMCR to 1. When write operation is enabled, the SPI mode write data register is transmitted to the serial flash memory. When read operation is enabled, data read from the serial flash memory is stored into the SPI mode read data register.
The SPI operation timing is shown in Figure 17.14.

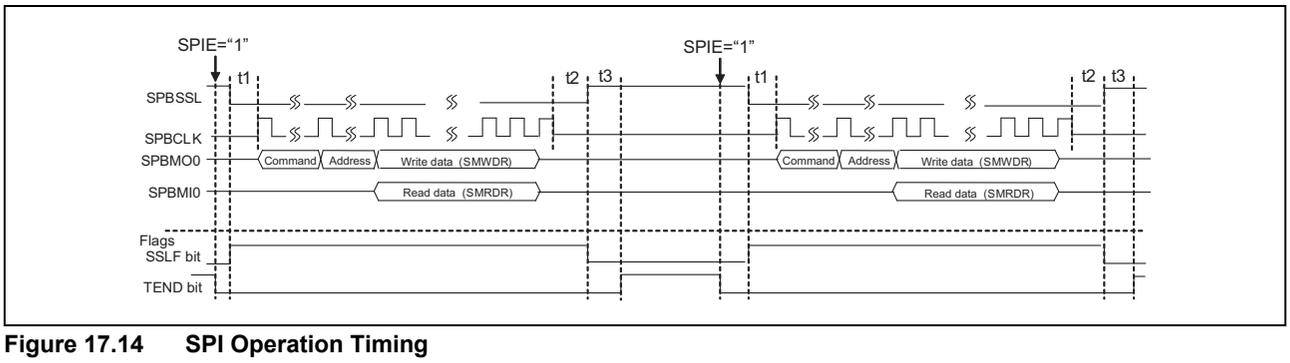


Figure 17.14 SPI Operation Timing

(2) Read/Write Enable

- Read operation: Data can be read by setting the SPIRE bit in SMCR to 1. The read data is stored into SMRDR.
- Write operation: Data can be written by setting the SPIWE bit in SMCR to 1. The data stored in SMWDR is output.

When the data size is set to 1 bit using the SPIDB[1:0] bits in SMENR, data can be transmitted and received by setting the SPIRE and SPIWE bits to 1. However, when the data size is set to 2 or 4 bits by using the SPIDB[1:0] bits, only one of the SPIRE and SPIWE bits should be enabled. The operation is not guaranteed if both the bits are enabled.

chanhvinh.com

(3) Retention of SPBSSL Pin Assertion

By setting the SSLKP bit in SMCR to 1, assertion of the SPBSSL pin can be continued till the next transfer. With this function, the transfer can be carried out continuously with the SPBSSL kept in the asserted state.
The data transfer timing using the SSLKP bit is shown in Figure 17.15.

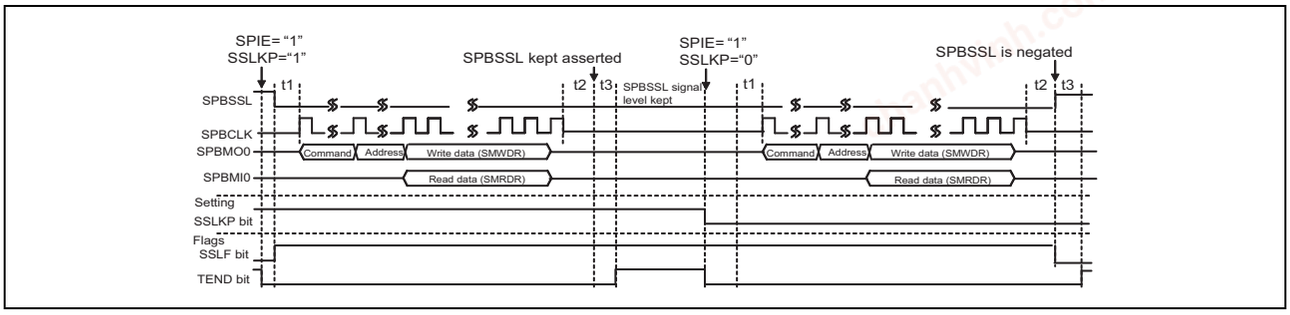


Figure 17.15 Data Transfer Timing using the SSLKP Bit

(4) Initial Setting Flow

An example of an initial setting flow in SPI operating mode is shown in Figure 17.16.

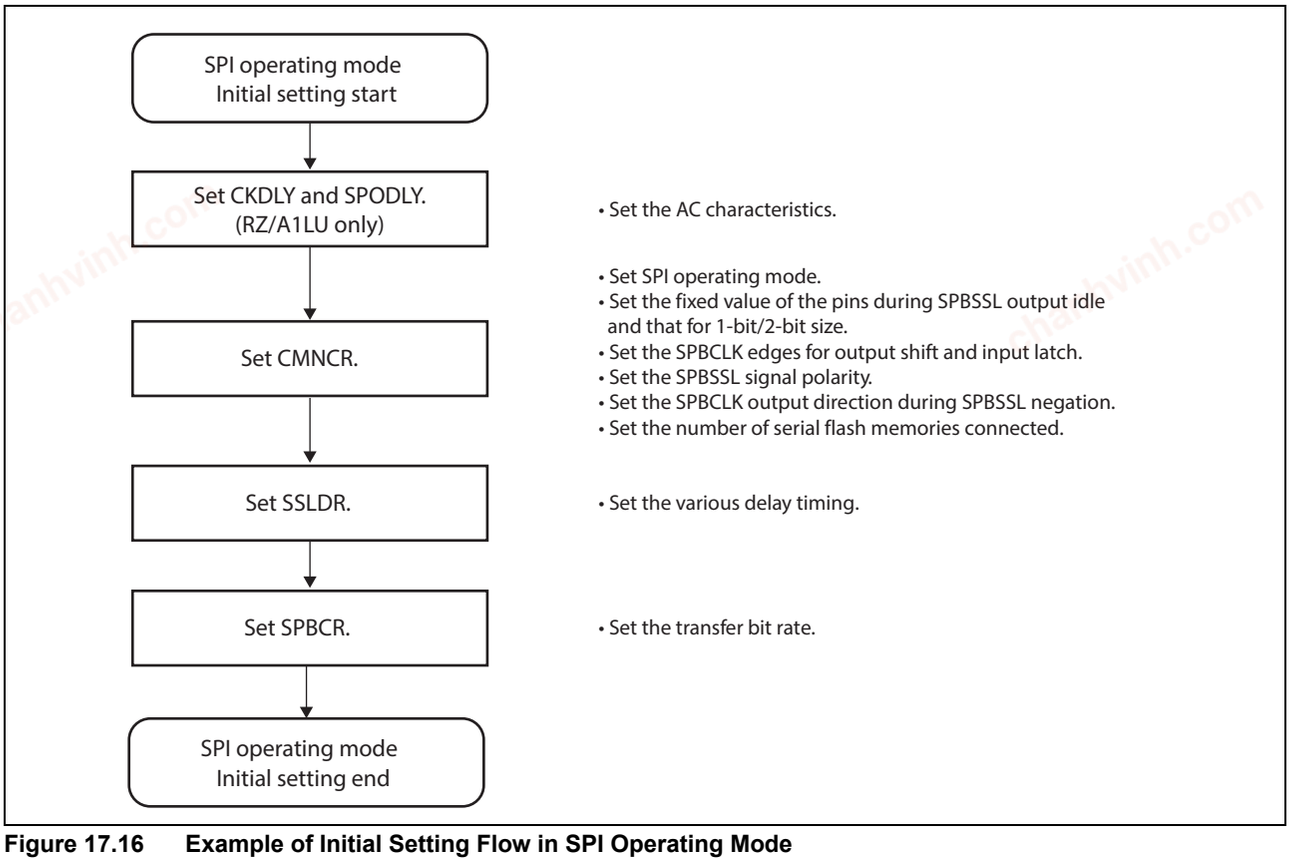


Figure 17.16 Example of Initial Setting Flow in SPI Operating Mode

chanhvinh.com

(5) Data Transfer Setting Flow

An example of a data transfer setting flow in SPI operating mode is shown in Figure 17.17.

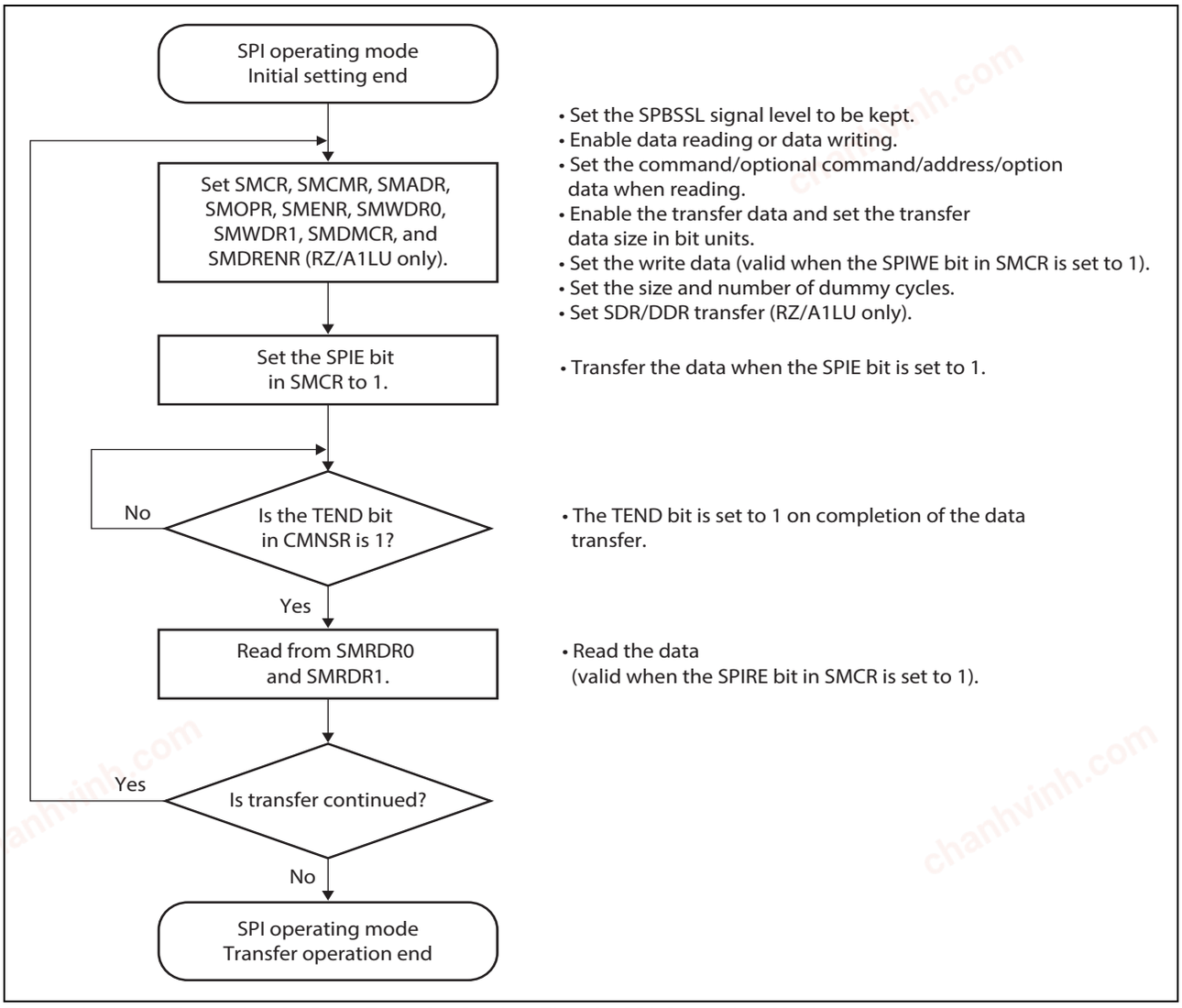


Figure 17.17 Example of a Data Transfer Setting Flow in SPI Operating Mode

chanhvinh.com

17.5.9 Transfer Format

(1) SPBSSL Pin Enable Polarity Control

The enable polarity of the SPBSSL pin can be changed with the SSLP bit in CMNCR.

(2) SPBCLK Output

The SPBCLK output direction during SPBSSL negation can be set with the CPOL bit in CMNCR.

(3) Data Transmission and Reception Timing

Data transmission and reception timing is different between SDR transfer and DDR transfer (RZ/A1LU only).
During SDR transfer, data is transmitted and received at either the odd or even edges. The data transmission timing can be set to the odd or even edge with the CPHAT bit in CMNCR. Similarly, the data reception timing can be set to the odd or even edge with the CPHAR bit in CMNCR.
During DDR transfer (RZ/A1LU only), data is transmitted and received at both the odd and even edges. The first data transmission timing can be set to the odd or even edge with the CPHAT bit in CMNCR. Similarly, the first data reception timing can be set to the odd or even edge with the CPHAR bit in CMNCR.

(4) Delay Settings

t1 is the time period from SPBSSL pin assertion to SPBCLK oscillation (clock delay). It can be set with the SCKDL[2:0] bits in SSLDR. t2 is the time period till the SPBSSL signal negation after the SPBCLK oscillation is stopped (SPBSSL negation delay). It can be set with the SLNDL[2:0] bits in SSLDR. t3 is the time period required to prevent SPBSSL signal assertion for the next transfer after the end of the previous transfer (next access delay). It can be set with the SPNDL[2:0] bits in SSLDR.

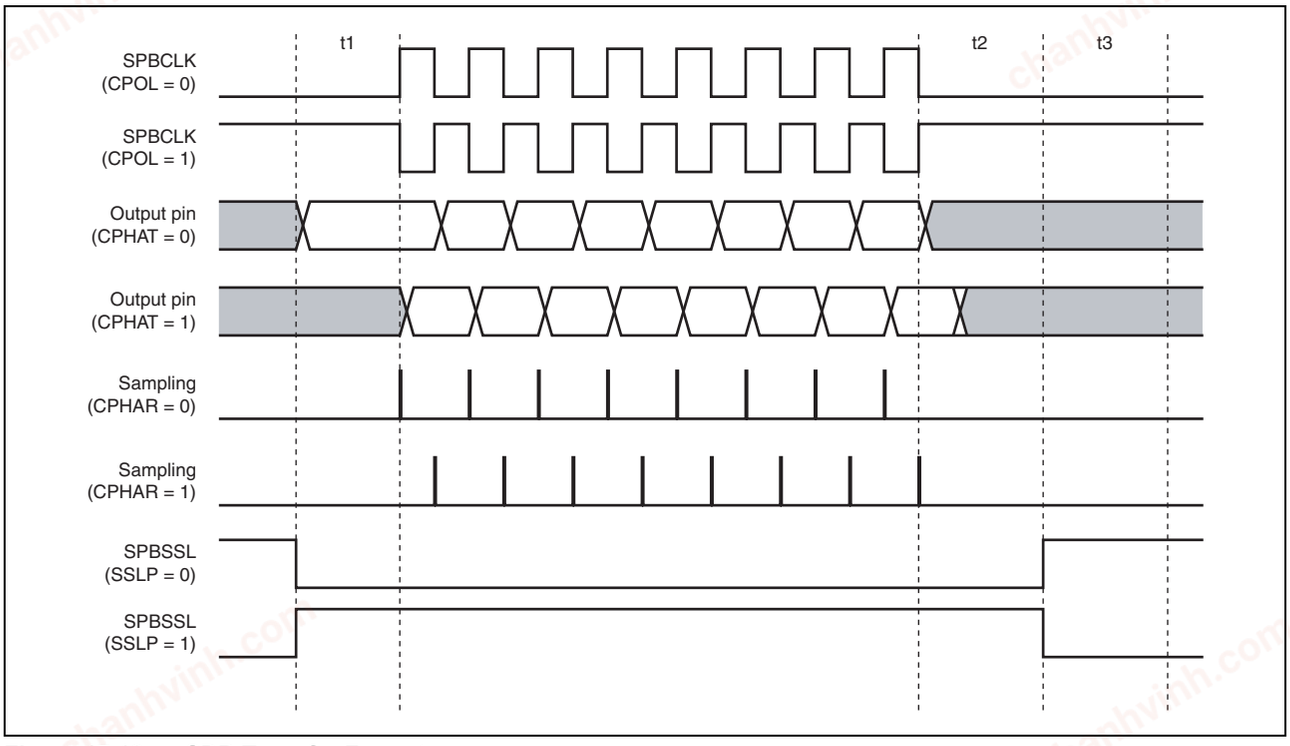


Figure 17.18 SDR Transfer Format

chanhvinh.com

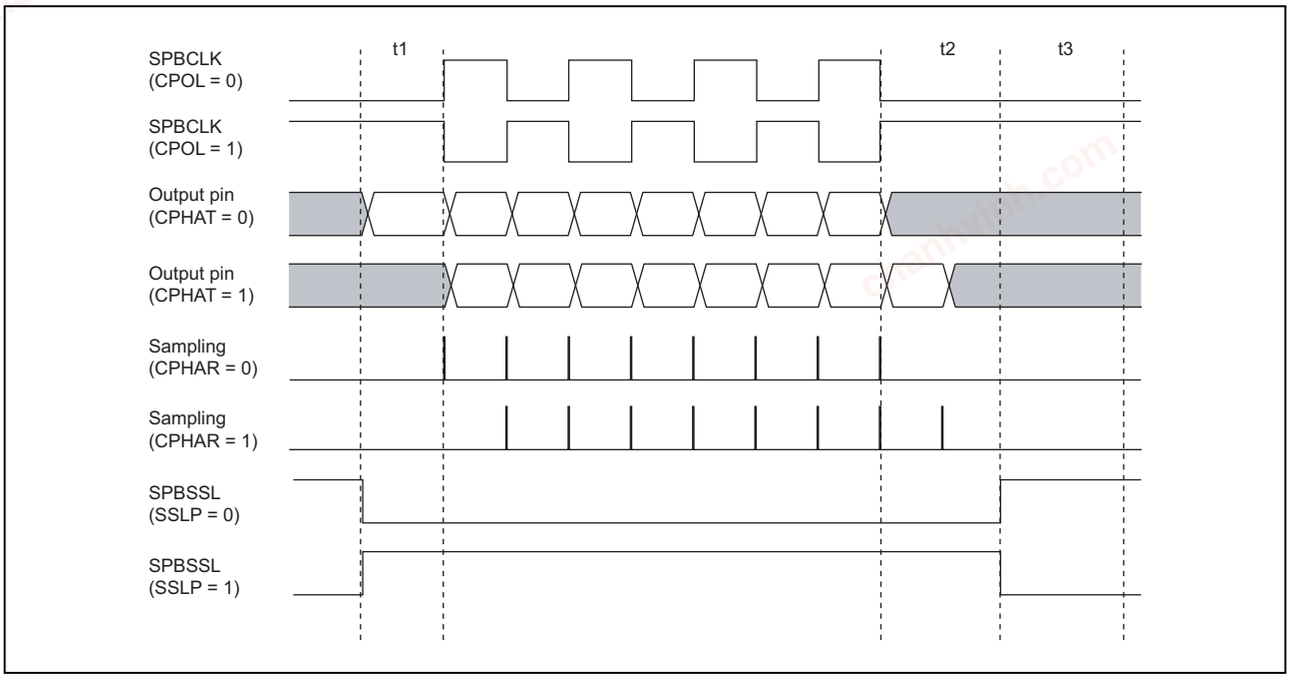


Figure 17.19 DDR Transfer Format (RZ/A1LU only)

Note: In DDR reception when CPHAR = 1, the sampling timing of the last bit is based on the frequency-divided clock in this module. When CPHAT = 1 in DDR transfer, the serial flash memory cannot provide the sampling timing of the last bit; therefore, a transfer ending with a DDR transfer is not performed correctly when CPHAT = 1.

chanhvinh.com

17.5.10 Data Format

This module can input and output data in the order of command, optional command, address, option data, dummy cycle, and data.

(1) Data Registers

Table 17.5 shows the input and output data.

Table 17.5 Data Registers

Data		External Address Space Read Operation	SPI Operation
Command (8 bits)		CMD[7:0] bits in DRCMR	CMD[7:0] bits in SMCMR
Optional command (8 bits)		OCMD[7:0] bits in DRCMR	OCMD[7:0] bits in SMCMR
Address (32/24 bits)	BSZ[1:0] = 00 (one flash memory connected)	32 bits: DREAR.EAV[6:1 to 0] bits + lower [25 to 24:0] bits of the read address. 24 bits: Lower [23:0] bits of the read address	32 bits: ADR[31:0] bits in SMADR 24 bits: ADR[23:0] bits in SMADR
	BSZ[1:0] = 01 (two flash memories connected)	32 bits: DREAR.EAV[7:1 to 0] bits + lower [25 to 24:1] bits of the read address. 24 bits: Lower [24:1] bits of the read address	
Option data (8 bits × 4)		DROPR	SMOPR
Dummy cycle (1 to 8 cycles)		DRDMCR	SMDMCR (only when read)
Transfer data		Normal read: 8, 16, and 32 bits Burst read: 64 × RBURST bits	Read: SMRDR0, SMRDR1 Write: SMWDR0, SMWDR1

(2) Data Enable

In external address space read mode, transfer enable or disable of the command, optional command, address, option data, and dummy cycle can be controlled with the CDE, OCDE, ADE[3:0], OPDE[3:0], and DME bits in DRENR, respectively. The size and number of dummy cycles can be controlled with the data read dummy cycle setting register (DRDMCR).

When the SPBCLK frequency division ratio is two or larger, either SDR or DDR transfer can be selected for the address, option data, and read data, using the ADDRE, OPDRE, and DRDRE bits in the data read DDR enable register (DRDRENR) (RZ/A1LU only).

Similarly, in SPI operating mode, enable or disable of the command, optional command, address, option data, dummy cycle, and transfer data can be controlled with the CDE, OCDE, ADE[3:0], OPDE[3:0], DME, and SPIDE[3:0] bits in SMENR, respectively. However, disabling all the above parameters is prohibited in SPI operating mode. At least one of them except dummy cycle must be enabled. The size and number of dummy cycles can be controlled with the SPI mode dummy cycle setting register (SMDMCR).

When the SPBCLK frequency division ratio is two or larger, either SDR or DDR transfer can be selected for the address, option data, and transfer data, using the ADDRE, OPDRE, and SPIDRE bits in the SPI mode DDR enable register (SMDRENR) (RZ/A1LU only).

For the address and option data in external address space read mode; and the address, option data, and transfer data in SPI operating mode, the enable bit setting allowed is determined according to the transfer data size. For the allowed setting combinations of the enable bits and transfer data size, refer to the description of the pertinent register.

If data is disabled, that data is skipped, and input and output of the next data is carried out. The command, optional command, address, and option data are always output. During dummy cycles, the state of the used pins is Hi-Z. In external address space read mode, data is always input; and in SPI operating mode, input and output of data is determined based on the settings of the SPIRE and SPIWE bits in SMCR.

There are some restrictions on dummy cycle insertion; refer to the description of the DME bits in DRENR and SMENR for details.

chanhvinh.com

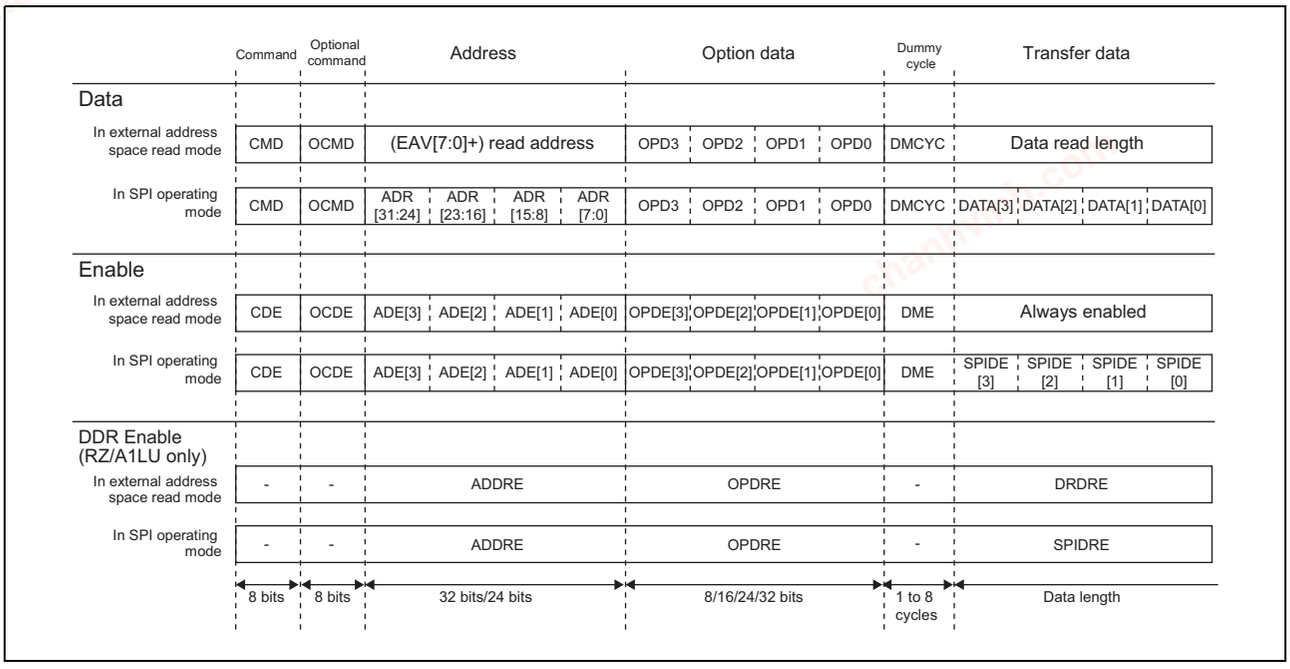


Figure 17.20 Data and Enable

chanhvinh.com

(3) Bit Size

In external address space read mode, the size of the command, optional command, address, option data, and the read data in bit units is respectively controlled with the CDB[1:0], OCDB[1:0], ADB[1:0], OPDB[1:0], and DRDB[1:0] bits in DRENR. The size of the dummy cycle in bit units is also controlled with the DMDB[1:0] bits in DRDMCR.
Similarly, in SPI operating mode, the size of the command, optional command, address, option data, and read write data in bit units is controlled with the CDB[1:0], OCDB[1:0], ADB[1:0], OPDB[1:0], and SPIDB[1:0] bits in SMENR. The size of the dummy cycle in bit units is also controlled with the DMDB[1:0] bits in SMDMCR.

(a) 1-bit Size

When the size is set to 1 bit, SPBMI0 and SPBMI1 pins will be the input pins and SPBMO0 and SPBMO1 pins will be the output pins. SPBIO20, SPBIO21, SPBIO30, and SPBIO31 pins are not used.
Figure 17.21 and Figure 17.22 show the transfer format examples.

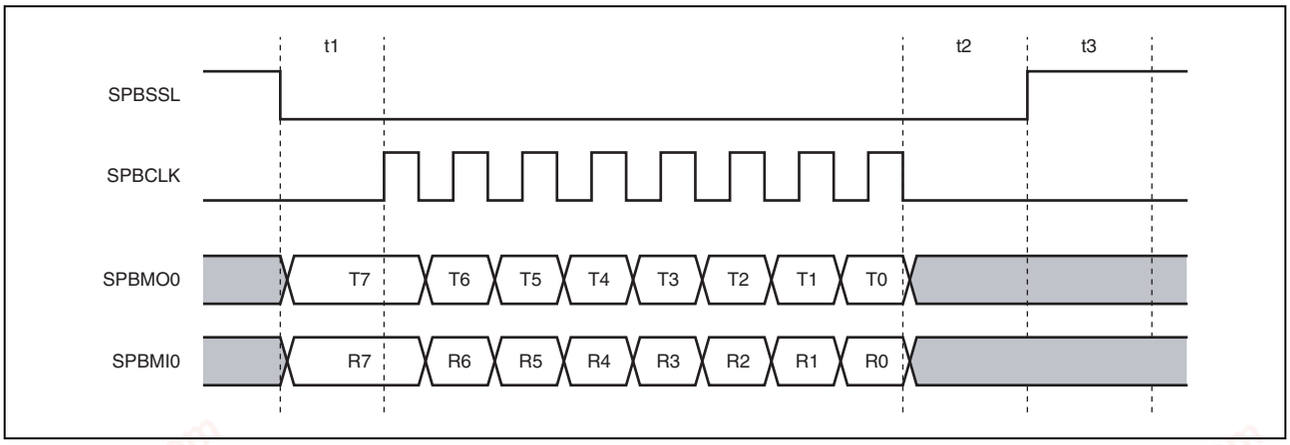


Figure 17.21 Transfer Format Example with 1-Bit Data Size and One Serial Flash Memory Connected

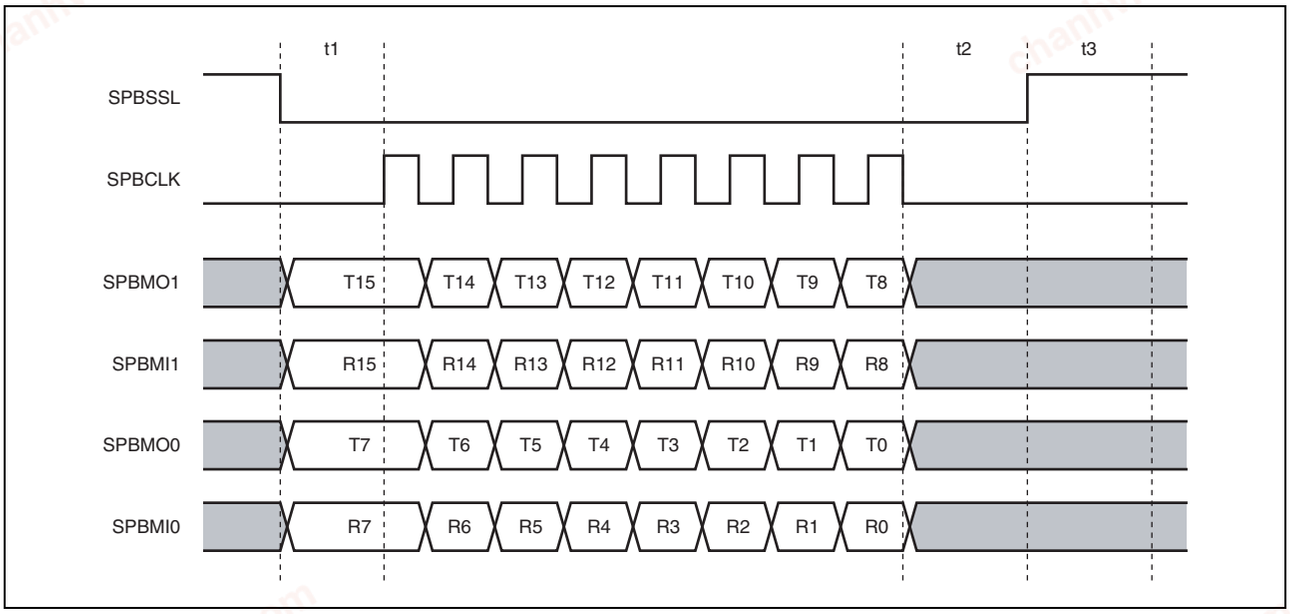


Figure 17.22 Transfer Format Example with 1-Bit Data Size and Two Serial Flash Memories Connected

chanhvinh.com

(b) 2-bit Size

When the size is set to 2 bits, SPBIO0_0, SPBIO01, SPBIO10, and SPBIO11 pins will be either the input pins or the output pins. SPBIO20, SPBIO21, SPBIO30, and SPBIO31 pins are not used.
Figure 17.23 and Figure 17.24 show the transfer format examples.

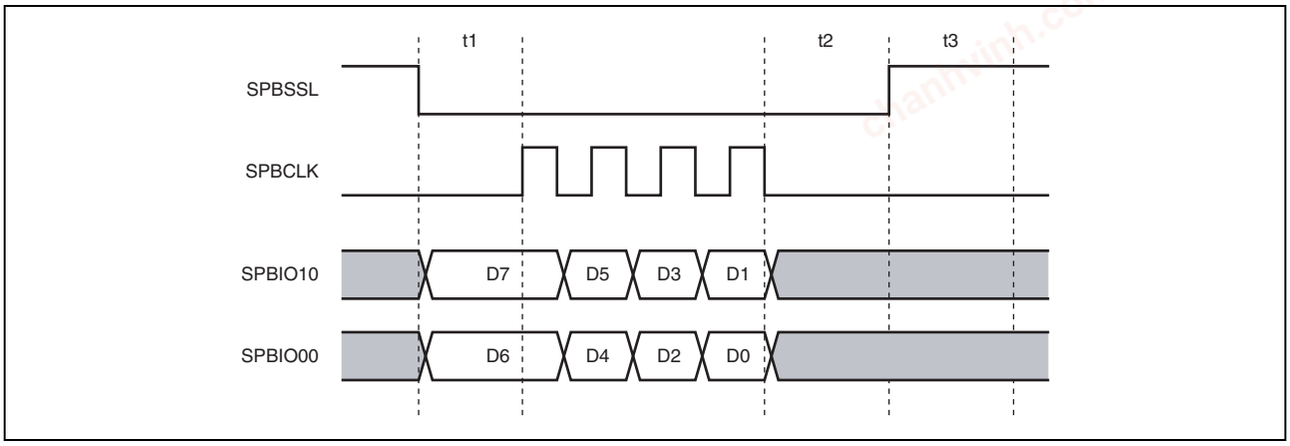


Figure 17.23 Transfer Format Example with 2-Bit Data Size and One Serial Flash Memory Connected

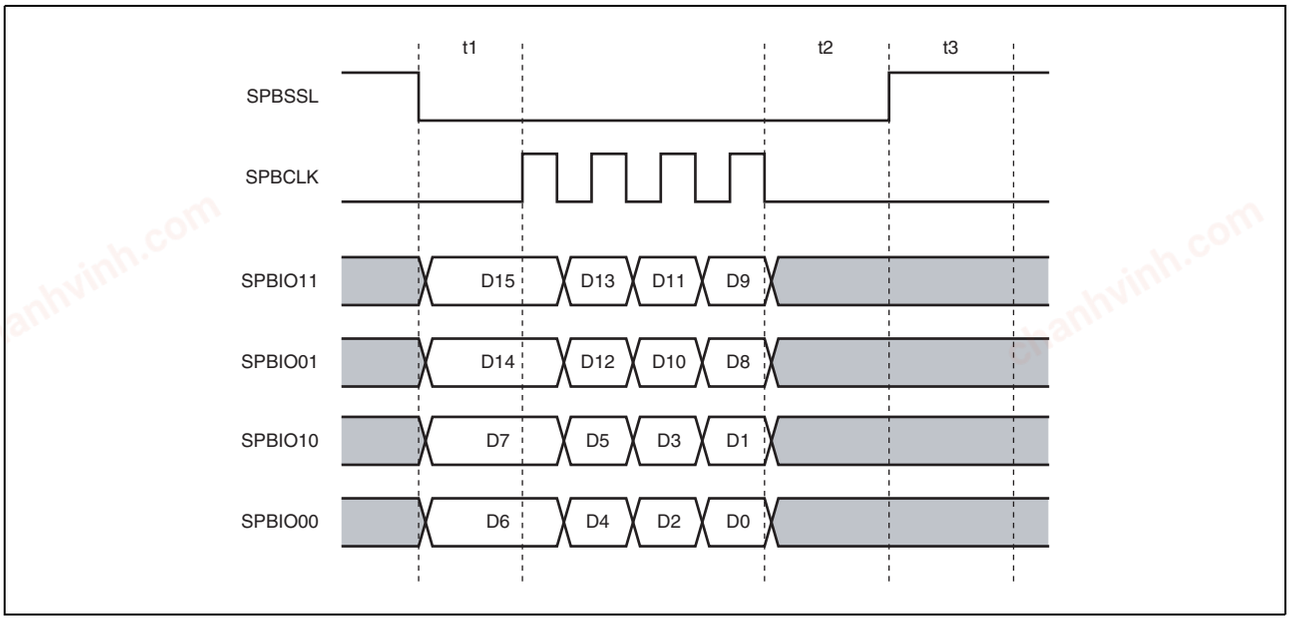


Figure 17.24 Transfer Format Example with 2-Bit Data Size and Two Serial Flash Memories Connected

chanhvinh.com

(c) 4-bit Size

When the size is set to 4 bits, SPBIO00, SPBIO01, SPBIO10, SPBIO11, SPBIO20, SPBIO21, SPBIO30, and SPBIO31 pins will be either the input pins or the output pins.

Figure 17.25 and Figure 17.26 show the transfer format examples.

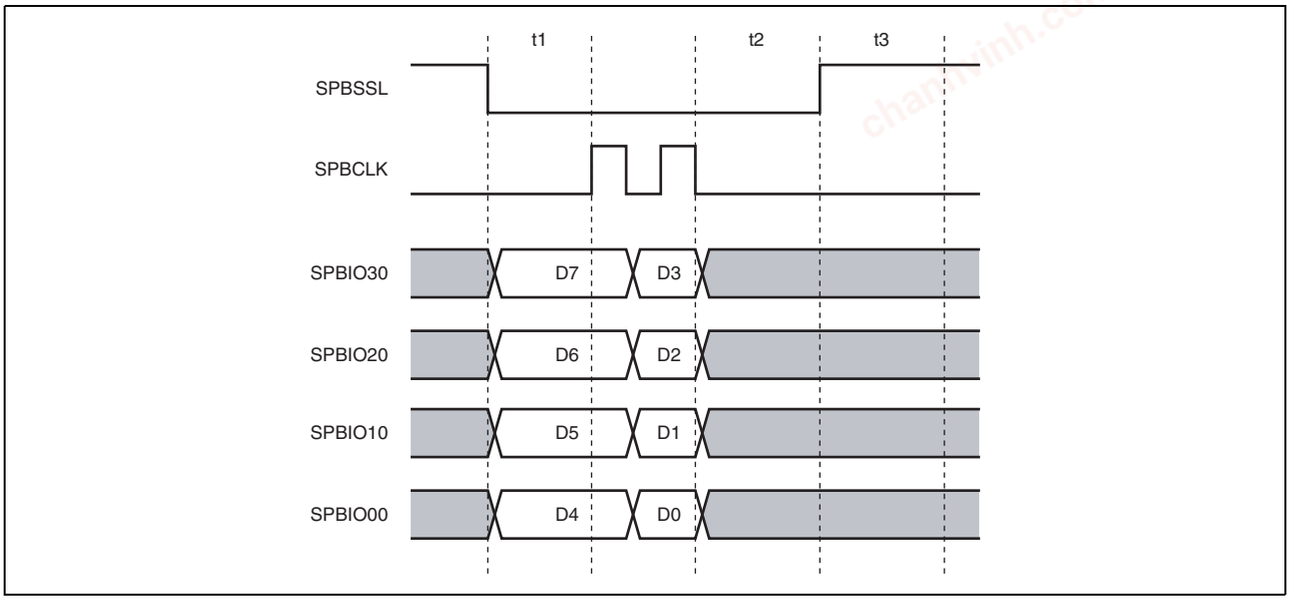


Figure 17.25 Transfer Format Example with 4-Bit Data Size and One Serial Flash Memory Connected

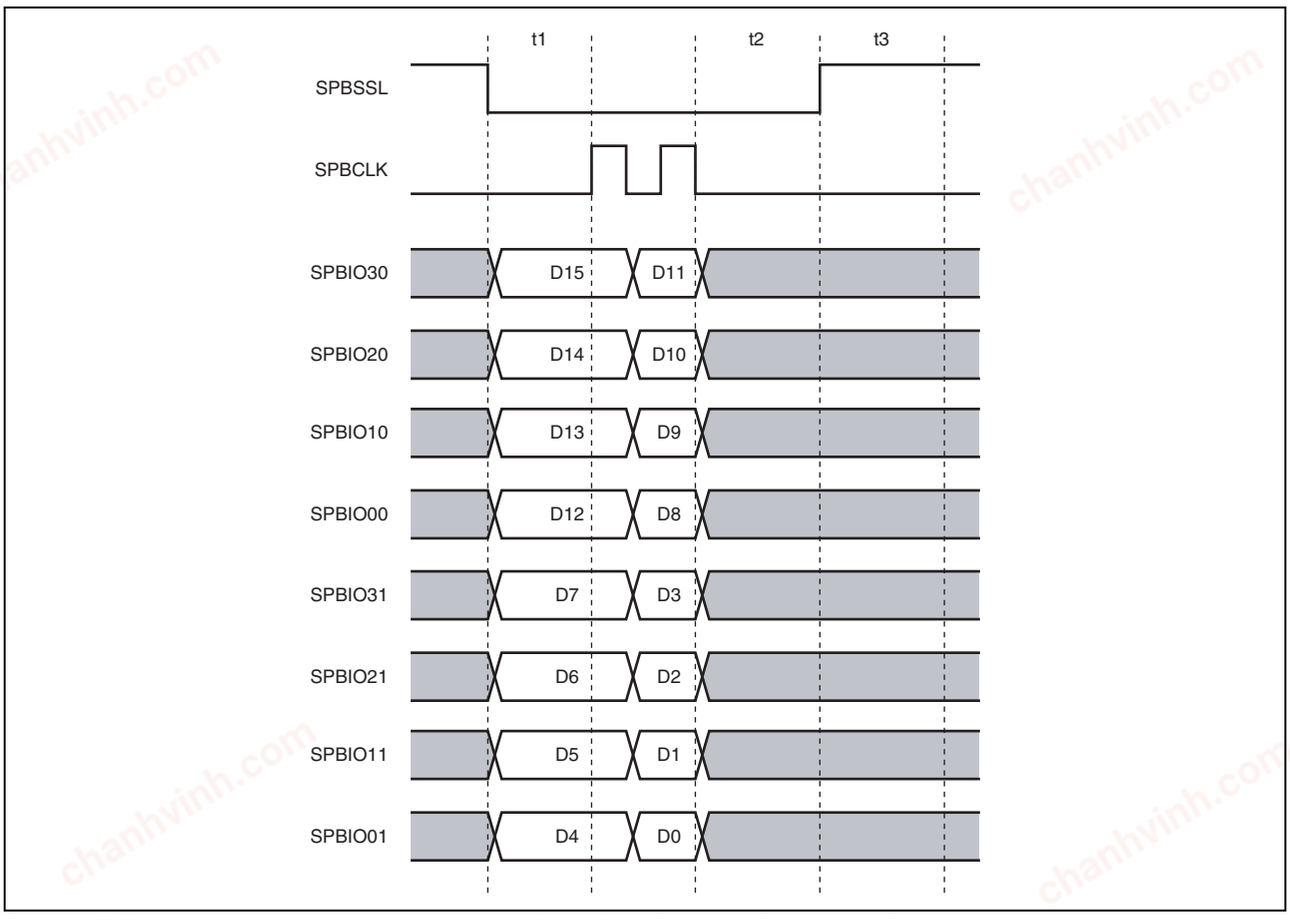


Figure 17.26 Transfer Format Example with 4-Bit Data Size and Two Serial Flash Memories Connected

chanhvinh.com

17.5.11 Data Pin Control

With this module, the status of pins can be automatically changed based on the data size to be used and the read/write settings. The pin status during the SPBSSL negation can be set with the MOIIO3, MOIIO2, MOIIO1, and MOIIO0 bits in CMNCR. The SPBSSL and SPBCLK pins are always output pins. The status of respective pins is specified in Table 17.6 to Table 17.9.

Table 17.6 Pin Status (1)

		SPBSSL Assertion		
		Command, Optional Command, Address, Option Data		
Pin	SPBSSL Negation	1-bit Size	2-bit Size	4-bit Size
SPBMO0/ SPBIO00, SPBMO1/ SPBIO01	MOIIO0 bit value	Output	Output	Output
SPBMI0/ SPBIO10, SPBMI1/ SPBIO11	MOIIO1 bit value	Hi-Z	Output	Output
SPBIO20, SPBIO21	MOIIO2 bit value	IO2FV bit value	IO2FV bit value	Output
SPBIO30, SPBIO31	MOIIO3 bit value	IO3FV bit value	IO3FV bit value	Output

Table 17.7 Pin Status (2)

	Transfer Data					
	External Address Space Read Operation			SPI Operation		
				SPIRE Bit = 1, SPIWE Bit = 0		
Pin	1-bit Size	2-bit Size	4-bit Size	1-bit Size	2-bit Size	4-bit Size
SPBMO0/ SPBIO00, SPBMO1/ SPBIO01	IO0FV bit value	Input	Input	IO0FV bit value	Input	Input
SPBMI0/ SPBIO10, SPBMI1/ SPBIO11	Input	Input	Input	Input	Input	Input
SPBIO20, SPBIO21	IO2FV bit value	IO2FV bit value	Input	IO2FV bit value	IO2FV bit value	Input
SPBIO30, SPBIO31	IO3FV bit value	IO3FV bit value	Input	IO3FV bit value	IO3FV bit value	Input

chanhvinh.com

Table 17.8 Pin Status (3)

	Transfer Data					
	SPI Operation					
	SPIRE Bit = 0, SPIWE Bit = 1			SPIRE Bit = 1, SPIWE Bit = 1		
Pin	**1-bit Size**	**2-bit Size**	**4-bit Size**	**1-bit Size**	**2-bit Size**	**4-bit Size**
SPBMO0/ SPBIO00, SPBMO1/ SPBIO01	Output	Output	Output	Output	Setting prohibited	Setting prohibited
SPBMI0/ SPBIO10, SPBMI1/ SPBIO11	Hi-Z	Output	Output	Input	Setting prohibited	Setting prohibited
SPBIO20, SPBIO21	IO2FV bit value	IO2FV bit value	Output	IO2FV bit value	Setting prohibited	Setting prohibited
SPBIO30, SPBIO31	IO3FV bit value	IO3FV bit value	Output	IO3FV bit value	Setting prohibited	Setting prohibited

Table 17.9 Pin Status (4)

	Dummy Cycle		
Pin	**1-bit Size**	**2-bit Size**	**4-bit Size**
SPBMO0/ SPBIO00, SPBMO1/ SPBIO01	IO0FV bit value	Hi-Z	Hi-Z
SPBMI0/ SPBIO10, SPBMI1/ SPBIO11	Hi-Z	Hi-Z	Hi-Z
SPBIO20, SPBIO21	IO2FV bit value	IO2FV bit value	Hi-Z
SPBIO30, SPBIO31	IO3FV bit value	IO3FV bit value	Hi-Z

chanhvinh.com

17.5.12 SPBSSL Pin Control

Negation conditions of the SPBSSL pin are as follows.

(1) External Address Space Read Mode

(a) Normal read operation (RBE bit in DRCR = 0)

SPBSSL negated after completing the data transfer and t2 cycle.

(b) Burst read without automatic SPBSSL negation (RBE bit in DRCR = 1, SSLE bit in DRCR = 0)

SPBSSL negated after completing the data transfer and t2 cycle.

(c) Burst read with automatic SPBSSL negation (RBE bit in DRCR = 1, SSLE bit in DRCR = 1)

- SPBSSL negated after t2 cycle when the read address is not continuous with the previously read address
- SPBSSL negated after the SSLN bit in DRCR is set to 1

(2) SPI Operating Mode

(a) SPBSSL pin assertion not retained (SSLKP bit in SMCR = 0)

SPBSSL negated after completing the data transfer and t2 cycle.

(b) SPBSSL pin assertion retained (SSLKP bit in SMCR = 1)

SPBSSL not negated.
When to be negated, data should be transferred after setting the SSLKP bit to 0.

17.5.13 Flags

This module has two flag bits SSLF and TEND in CMNSR. These bits are read-only bits.

(1) SSLF Bit

This bit indicates the SPBSSL pin status. The status is 1 when the SPBSSL is asserted, and the status is 0 when the SPBSSL is negated.

(2) TEND Bit

This bit indicates whether transfer of data is in progress or the transfer of data has ended.
During t1 time period, data transfer, t2 time period, t3 time period, and waiting for read access by burst read and SPBSSL automatic negation, the TEND bit is read as 0 to indicate that the transfer of data is in progress.
When other than the above, the TEND bit is read as 1 to indicate that transfer of data has ended.

(3) Register Re-writing Timing

The status of the TEND bit determines the rewritable registers.
The registers which can be written to, except the SSLN bit in DRCR, should be modified when TEND = 1. Read SMRDR0 and SMRDR1 when TEND = 1. CMNSR can always be read.

chanhvinh.com

17.6 Usage Notes

17.6.1 Notes on Transfer to Read Data in SPI Operating Mode

If the setting for the bit mode is for division by two or more in SPI operating mode, take note of the following points for caution when setting the SPI mode enable setting register (SMENR) to enable transfer only for reading data.
“Transfer only for reading data” indicates transfer to read data while the CDE, OCDE, ADE[3:0], and OPDE[3:0] bits in SMENR are all 0.

(1) Transfer to read data while the signal on the SPBSSL pin is de-asserted

Set the SMENR.SPIDE[3:0] bits to 1100 or 1111 when transfer only for reading data is to proceed.
Transfer will not proceed normally if the setting of the SMENR.SPIDE[3:0] bits is 1000.

(2) Transfer to read data while the signal on the SPBSSL pin is asserted

When transfer only for reading data is to proceed, set the SMENR.SPIDE[3:0] bits to 1100 or 1111, or end the immediately preceding transfer with reading data.
When the immediately preceding transfer is of a command, optional command, address, or option data, or is transfer for writing data, the subsequent transfer only for reading data will not proceed normally if the setting of the SMENR.SPIDE[3:0] bits is 1000.

17.6.2 Notes on Starting Transfer from the SPBSSL Retained State in SPI Operating Mode

Be sure to set the SPIWE bit in the SMCR register to 1 when the transfer of a command, optional command, address, or option data is started while the SPBSSL pin is being asserted in SPI operating mode.

chanhvinh.com

18. I²C Bus Interface

This section gives an overall description of the I^2C bus interface (RIIC).

The first section describes the features specific to this LSI, including the number of channels and the register base addresses. The subsequent sections describe the RIIC's functions and registers.

18.1 Features

18.1.1 Channels

This LSI has the following number of channels of the I^2C bus interface (RIIC).

Table 18.1 Channels of RIIC

Product Name	RZ/A1L 176 pins	RZ/A1L 208 pins
Number of channels	4	
Name	RIICn (n = 0 to 3)	

Table 18.2 Index

Index	Description
n	Throughout this section, each channel of the I^2C bus interface is identified by the index "n" (n = 0 to 3). For example, RIICnCR1 is I^2C bus control register 1.

18.1.2 Register Base Addresses

The base address <RIICn_base> of each RIICn is listed in the following table.

All RIICn register addresses are given as values obtained by adding offsets to the register base address <RIICn_base> for each channel.

Table 18.3 Register Base Address

Channel	Base Address Name	Base Address
RIIC0	<RIIC0_base>	$FCFE\ E000_H$
RIIC1	<RIIC1_base>	$FCFE\ E400_H$
RIIC2	<RIIC2_base>	$FCFE\ E800_H$
RIIC3	<RIIC3_base>	$FCFE\ EC00_H$

chanhvinh.com

18.1.3 External I/O Signals

The following table shows the external I/O signals of the RIICn.

Table 18.4 RIICn Pin Configuration

Channel	Multiplexed Pin Name	Function
RIIC0	RIIC0SCL	RIIC0 serial clock I/O pin
	RIIC0SDA	RIIC0 serial data I/O pin
RIIC1	RIIC1SCL	RIIC1 serial clock I/O pin
	RIIC1SDA	RIIC1 serial data I/O pin
RIIC2	RIIC2SCL	RIIC2 serial clock I/O pin
	RIIC2SDA	RIIC2 serial data I/O pin
RIIC3	RIIC3SCL	RIIC3 serial clock I/O pin
	RIIC3SDA	RIIC3 serial data I/O pin

chanhvinh.com

18.2 Overview

18.2.1 Functional Overview

Communications format

- I²C bus format or SMBus format
- Master mode or slave mode selectable
- Automatic securing of the various set-up times, hold times, and bus-free times for the transfer rate

Transfer rate

Up to 400 kbps

SCL clock

For master operation, the duty cycle of the SCL clock is selectable in the following range.

0% < Duty cycle < 100%

Issuing and detecting conditions

- Start, restart, and stop conditions are automatically generated.
- Start conditions (including restart conditions) and stop conditions are detected.

Slave address

- Up to three slave-address settings can be made.
- Seven- and ten-bit address formats are supported (along with the use of both at once).
- General call addresses, device ID addresses, and SMBus host addresses are detectable.

Acknowledgement

- For transmission, the acknowledge bit is automatically loaded
  - Transfer of the next data for transmission can be automatically suspended on detection of a not-acknowledge bit.
- For reception, the acknowledge bit is automatically transmitted
  - If a wait between the eighth and ninth clock cycles has been selected, software control of the value in the acknowledge field in response to the received value is possible.

Wait function

- In reception, the following periods of waiting can be obtained by holding the clock signal (SCL) at the low level:
  - Waiting between the eighth and ninth clock cycles
  - Waiting between the ninth clock cycle and the first clock cycle of the next transfer (WAIT function)

SDA output delay function

Timing of the output of transmitted data, including the acknowledge bit, can be delayed.

chanhvinh.com

Arbitration

- For multi-master operation
  - Operation to synchronize the SCL (clock) signal in cases of conflict with the SCL signal from another master is possible.
  - When issuing the start condition would create conflict on the bus, loss of arbitration is detected by testing for non-matching between the internal signal for the SDA line and the level on the SDA line.
  - In master operation, loss of arbitration is detected by testing for non-matching between the signal on the SDA line and the internal signal for the SDA line.
- Loss of arbitration due to detection of the start condition while the bus is busy is detectable (to prevent the issuing of double start conditions).
- Loss of arbitration in transfer of a not-acknowledge bit due to the internal signal for the SDA line and the level on the SDA line not matching is detectable.
- Loss of arbitration due to non-matching of internal and line levels for data is detectable in slave transmission.

Timeout function

The internal time-out function is capable of detecting long-interval stop of the SCL (clock signal).

Noise removal

The interface incorporates analog noise filters and digital noise filters for input on the RIICnSCL and RIICnSDA pins, and the width for noise cancellation by the digital noise filters is adjustable by software.

Interrupt sources

- Eight sources:
  - Transmission complete
  - Receive data full
  - Transmit data empty
  - Detection of a stop condition
  - Detection of a start condition
  - Reception of a NACK
  - Arbitration lost
  - Timeout

Low power consumption function

Module-stop state can be set.

chanhvinh.com

18.2.2 Block Diagram

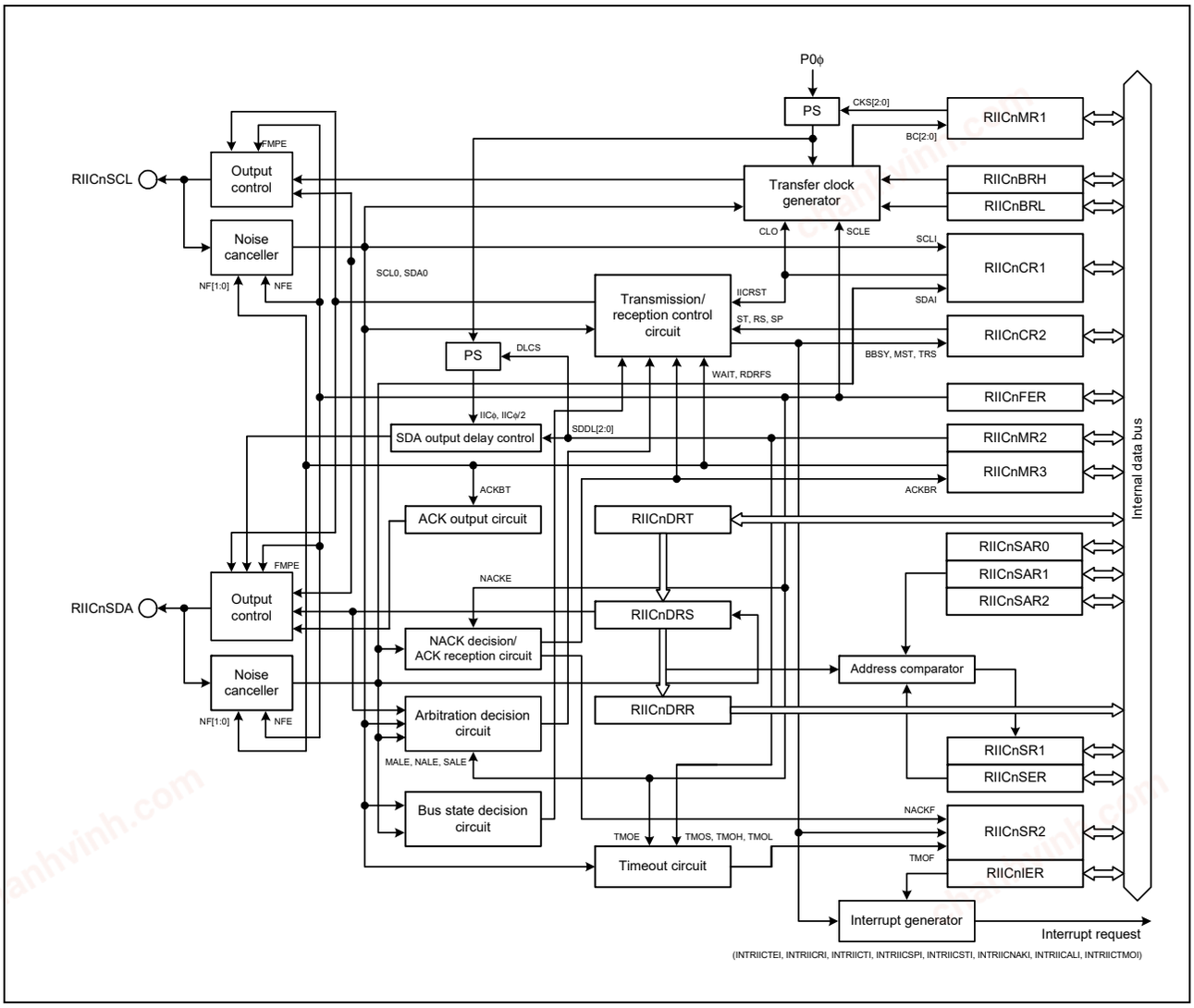


Figure 18.1 Block Diagram of RIIC

chanhvinh.com

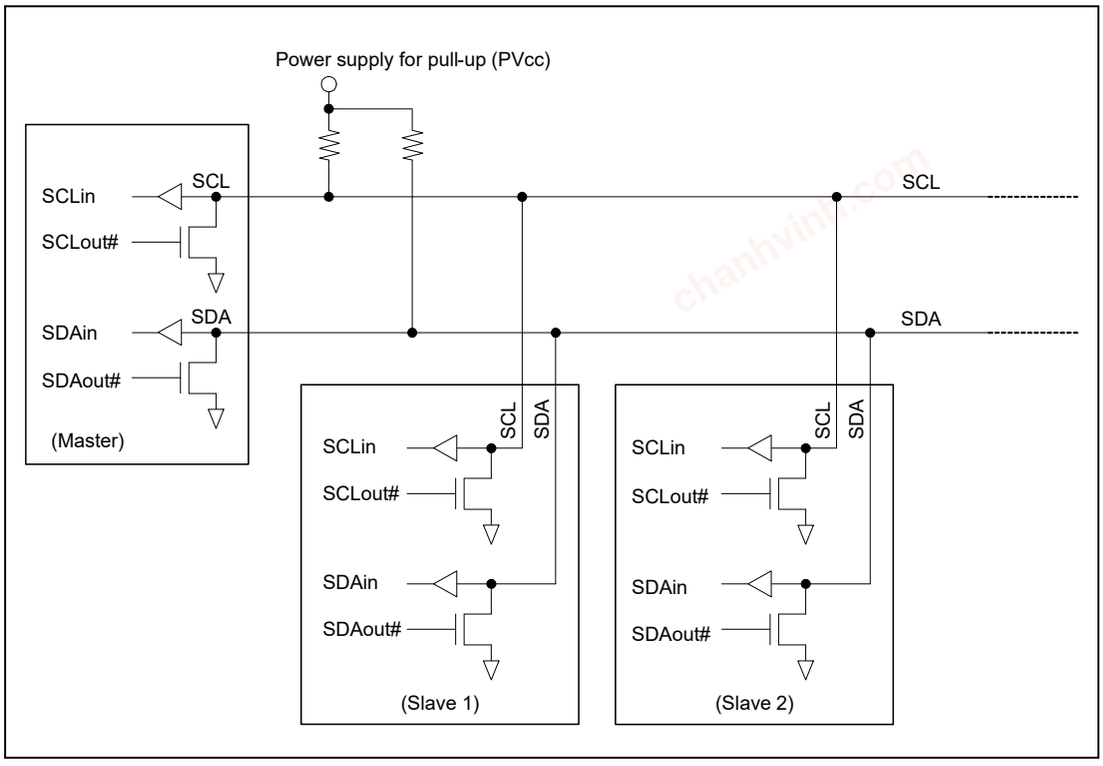


Figure 18.2 Connections to the External Circuit by the I/O Pins (I^2C Bus Configuration Example)

RIICnSCL and RIICnSDA are Schmitt input/open-drain output pins for both master and slave operations. Because the output is open drain, an external pull-up resistor is required.

chanhvinh.com

18.3 Registers

18.3.1 RIICnCR1 — I²C Bus Control Register 1

Access: RIICnCR1 is a 32-bit readable/writable register.
RIICnCR1L and RIICnCR1H are 16-bit readable/writable registers.
RIICnCR1LL, RIICnCR1LH, RIICnCR1HL, and RIICnCR1HH are 8-bit readable/writable registers.

Address: RIICnCR1: <RIICn_base> + 0000_H
RIICnCR1L: <RIICn_base> + 0000_H, RIICnCR1H: <RIICn_base> + 0002_H
RIICnCR1LL: <RIICn_base> + 0000_H, RIICnCR1LH: <RIICn_base> + 0001_H, RIICnCR1HL: <RIICn_base> + 0002_H,
RIICnCR1HH: <RIICn_base> + 0003_H

Initial Value: 0000 001F$_H$. This register is initialized by any reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
Initial value	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 18.5 RIICnCR1 register contents (1/2)

Bit Position	Bit Name	Function
31 to 8	—	Reserved These bits are read as 0. The write value should be 0.
7	ICE	I²C Bus Interface Enable 0: Output from the RIICnSCL and RIICnSDA pins is disabled (and input to the RIICnSCL and RIICnSDA pins is enabled). 1: Enabled (RIICnSCL and RIICnSDA pins driven) (This bit can select the RIIC reset or internal reset in combination with the IICRST bit.)
6	IICRST	I²C Bus Interface Internal Reset 0: Clears the RIIC reset or internal reset. 1: Initiates the RIIC reset or internal reset. (Clears the bit counter and the SCL/SDA output latch)
5	CLO	Extra SCL Clock Cycle Output 0: Does not output an extra SCL clock cycle (default). 1: Outputs an extra SCL clock cycle. (The CLO bit is cleared automatically after one clock cycle is output.)
4	SOWP*2	SCLO/SDAO Write Protect 0: Allows the SCLO and SDAO bits to be rewritten. (This bit is read as 1.)
3	SCLO*1,*2	SCL Output Control • Read: 0: RIICnSCL pin output is at a low level. 1: RIICnSCL pin is in a high-impedance state. • Write: 0: Changes the RIICnSCL pin output to a low level. 1: Changes the RIICnSCL pin in a high-impedance state. (High level output is achieved through an external pull-up resistor.)

chanhvinh.com

Table 18.5 RIICnCR1 register contents (2/2)

Bit Position	Bit Name	Function
2	SDAO*1,*2	SDA Output Control • Read: 0: RIICnSDA pin output is at a low level. 1: RIICnSDA pin is in a high-impedance state. • Write: 0: Changes the RIICnSDA pin output to a low level. 1: Changes the RIICnSDA pin in a high-impedance state. (High level output is achieved through an external pull-up resistor.)
1	SCLI	SCL Bus Input Monitor 0: RIICnSCL pin input is at a low level. 1: RIICnSCL pin input is at a high level.
0	SDAI	SDA Bus Input Monitor 0: RIICnSDA pin input is at a low level. 1: RIICnSDA pin input is at a high level.

Note 1. Do not write to these bits during communication. Changing a value during communication may cause a transmission or reception failure or an AL error.

Note 2. To change the SDAO and SCLO bits, set the SOWP bit to 0 at the same timing to set the SDAO and SCLO bits to 0.

CLO Bit (Extra SCL Clock Cycle Output)

This bit is used to output an extra SCL clock cycle for debugging or error processing.

Normally, set the bit to 0. Setting the bit to 1 in a normal communication state causes a communication error.

For details on this function, see Section 18.13.2, Extra SCL Clock Cycle Output Function.

IICRST Bit (I²C Bus Interface Internal Reset)

This bit is used to reset the internal states of the RIIC.

Setting this bit to 1 initiates an RIIC reset or internal reset.

Whether an RIIC reset or internal reset is initiated is determined according to the combination with the ICE bit. Table 18.6 lists the resets of the RIIC.

The RIIC reset resets all registers including the RIICnCR2.BBSY flag (except ICE and IICRST) and internal states of the RIIC, and the internal reset resets the bit counter (RIICnMR1.BC[2:0] bits), the I²C bus shift register (RIICnDRS), and the I²C bus status registers (RIICnSR1 and RIICnSR2) as well as the internal states of the RIIC. For the reset conditions for each register, see Section 18.15, Reset Function of RIIC.

An internal reset initiated with the IICRST bit set to 1 during operation (with the ICE bit set to 1) resets the internal states of the RIIC without initializing the port settings and the control and setting registers of the RIIC when the bus or RIIC hangs up due to a communication error.

If the RIIC hangs up in a low level output state, resetting the internal states cancels the low level output state and releases the bus with the RIICnSCL pin and RIICnSDA pin at a high impedance.

chanhvinh.com

CAUTION

If an internal reset is initiated using the IICRST bit for a bus hang-up occurred during communication with the master device in slave mode, the states may become different between the slave device and the master device (due to the difference in the bit counter information). For this reason, do not initiate an internal reset in slave mode, but initiate restoration processing from the master device. If an internal reset is necessary because the RIIC hangs up with the SCL line in a low level output state in slave mode, initiate an internal reset and then issue a restart condition from the master device or resume communication from the start condition issuance after issuing a stop condition. If communication is restarted by initiating a reset solely in the slave device without issuing a start condition or restart condition from the master device, synchronization will be lost because the master and slave devices operate asynchronously.

Table 18.6 RIIC Resets

IICRST	ICE	State	Specifications
1	0	RIIC reset	Resets all registers (except ICE and IICRST) and internal states of the RIIC.
	1	Internal reset	Reset the RIICnMR1.BC[2:0] bits, and the RIICnSR1, RIICnSR2, RIICnDRS registers and the internal states of the RIIC.

ICE Bit (I²C Bus Interface Enable)

The ICE bit selects driving or non-driving of the RIICnSCL and RIICnSDA pins. Moreover, this bit can perform two types of reset in combination with the IICRST bit. For the types of reset, see Table 18.6, RIIC Resets.

Set the ICE bit to 1 when using RIIC. Setting the ICE bit to 1 selects driving of the RIICnSCL and RIICnSDA pins.

Set the ICE bit to 0 when RIIC is not to be used. Clearing the ICE bit to 0 disables output from the RIICnSCL and RIICnSDA pins.

chanhvinh.com

18.3.2 RIICnCR2 — I²C Bus Control Register 2

Access: RIICnCR2 is a 32-bit readable/writable register.
RIICnCR2L and RIICnCR2H are 16-bit readable/writable registers.
RIICnCR2LL, RIICnCR2LH, RIICnCR2HL, and RIICnCR2HH are 8-bit readable/writable registers.

Address: RIICnCR2: <RIICn_base> + 0004_H
RIICnCR2L: <RIICn_base> + 0004_H, RIICnCR2H: <RIICn_base> + 0006_H
RIICnCR2LL: <RIICn_base> + 0004_H, RIICnCR2LH: <RIICn_base> + 0005_H, RIICnCR2HL: <RIICn_base> + 0006_H,
RIICnCR2HH: <RIICn_base> + 0007_H

Initial Value: $0000\ 0000_H$ This register is initialized by any reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	BBSY	MST	TRS	—	SP	RS	ST	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R

Table 18.7 RIICnCR2 register contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved These bits are read as 0. The write value should be 0.
7	BBSY	Bus Busy Detection Flag 0: The I²C bus is released (bus free state). 1: The I²C bus is occupied (bus busy state or in the bus free state).
6	MST	Master/Slave Mode 0: Slave mode 1: Transmit mode
5	TRS	Transmit/Receive Mode 0: Receive mode 1: Transmit mode
4	—	Reserved This bit is read as 0. The write value should be 0.
3	SP	Stop Condition Issuance Request 0: Does not request to issue a stop condition. 1: Requests to issue a stop condition.
2	RS	Restart Condition Issuance Request 0: Does not request to issue a restart condition. 1: Requests to issue a restart condition.
1	ST	Start Condition Issuance Request 0: Does not request to issue a start condition. 1: Requests to issue a start condition.
0	—	Reserved This bit is read as 0. The write value should be 0.

ST Bit (Start Condition Issuance Request)

This bit is used to request transition to master mode and issuance of a start condition.

When this bit is set to 1 to request to issue a start condition, a start condition is issued when the BBSY flag is set to 0 (bus free).

chanhvinh.com

For details on the start condition issuance, see Section 18.12, Start Condition/Restart Condition/ Stop Condition Issuing Function.

[Setting condition]

When 1 is written to the ST bit

[Clearing conditions]

- When 0 is written to the ST bit
- When a start condition has been issued
- When the RIICnSR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

CAUTION

Set the ST bit to 1 (start condition issuance request) when the BBSY flag is set to 0 (bus free).
Note that arbitration may be lost if the ST bit is set to 1 (start condition issuance request) when the BBSY flag is set to 1 (bus busy).

RS Bit (Restart Condition Issuance Request)

This bit is used to request that a restart condition be issued in master mode.

When this bit is set to 1 to request to issue a restart condition, a restart condition is issued when the BBSY flag is set to 1 (bus busy) and the MST bit is set to 1 (master mode).

For details on the restart condition issuance, see Section 18.12, Start Condition/Restart Condition/ Stop Condition Issuing Function.

[Setting condition]

When 1 is written to the RS bit with the RIICnCR2.BBSY flag set to 1

[Clearing conditions]

- When 0 is written to the RS bit
- When a restart condition has been issued or a start condition is detected
- When a stop condition is detected
- When the RIICnCR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

CAUTIONS

1. Do not set the RS bit to 1 while issuing a stop condition.
2. It is commended to issue a restart condition in master transmit mode. If the RS bit is set to 1 (restart condition issuance request) in mode other than master mode, the restart condition is not issued in this mode but the restart condition issuance request bit remains set. If the operating mode changes to master mode with the bit not being cleared, the restart condition may be issued.

chanhvinh.com

SP Bit (Stop Condition Issuance Request)

This bit is used to request that a stop condition be issued in master mode.

When this bit is set to 1 to request to issue a stop condition, a stop condition is issued when the BBSY flag is set to 1 (bus busy) and the MST bit is set to 1 (master mode).

For details on the stop condition issuance, see Section 18.12, Start Condition/Restart Condition/ Stop Condition Issuing Function.

[Setting condition]

When 1 is written to the SP bit with both the RIICnCR2.BBSY flag and the RIICnCR2.MST bit set to 1

[Clearing conditions]

- When 0 is written to the SP bit
- When a stop condition has been issued or a stop condition is detected
- When the RIICnSR2.AL (arbitration-lost) flag is set to 1
- When a start condition and a restart condition are detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

CAUTIONS

1. Wring to the SP bit is not possible while the setting of the BBSY flag is 0 (bus free).
2. Do not set the SP bit to 1 while a restart condition is being issued.

TRS Bit (Transmit/Receive Mode)

This bit indicates transmit or receive mode.

The RIIC is in receive mode when the TRS bit is set to 0 and is in transmit mode when the bit is set to 1. Combination of this bit and the MST bit indicates the operating mode of the RIIC.

The value of the TRS bit is automatically changed to the value for transmission mode or reception mode by detection or issuing of a start condition, setting or clearing of the R/W# bit, etc.

[Setting conditions]

- When a start condition is issued normally according to the start condition issuance request (when a start condition is detected with the ST bit set to 1)
- When the R/W# bit added to the slave address is set to 0 in master mode
- When the address received in slave mode matches the address enabled in RIICnSER, with the R/W# bit set to 1

[Clearing conditions]

- When a stop condition is detected
- The RIICnSR2.AL (arbitration-lost) flag being set to 1
- In master mode, reception of a slave address to which an R/W# bit with the value 1 is appended
- In slave mode, a match between the received address and the address enabled in RIICnSER when the value of the received R/W# bit is 0 (including cases where the received address is the general call address)
- In slave transmit mode, a restart condition is detected (a restart condition is detected with

chanhvinh.com

RIICnCR2.BBSY = 1 and RIICnCR2.MST = 0)

- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

MST Bit (Master/Slave Mode)

This bit indicates master or slave mode.

The RIIC is in slave mode when the MST bit is set to 0 and is in master mode when the bit is set to 1. Combination of this bit and the TRS bit indicates the operating mode of the RIIC.

The value of the MST bit is automatically changed to the value for master mode or slave mode by detection or issuing of a start condition, etc.

[Setting conditions]

- When a start condition is issued normally according to the start condition issuance request (when a start condition is detected with the ST bit set to 1)

[Clearing conditions]

- When a stop condition is detected
- When the RIICnSR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

BBSY Flag (Bus Busy Detection)

The BBSY flag indicates whether the I^2C bus is occupied (bus busy) or released (bus free).

This bit is set to 1 when the SDA line changes from high to low under the condition of SCL = high, assuming that a start condition has been issued.

When the SDA line changes from low to high under the condition of SCL = high, this bit is cleared to 0 after the bus free time (specified in RIICnBRL) start condition is not detected, assuming that a stop condition has been issued.

[Setting condition]

When a start condition is detected

[Clearing conditions]

- When the bus free time (specified in RIICnBRL) start condition is not detected after detecting a stop condition
- When 1 is written to the RIICnCR1.IICRST bit with the RIICnCR1.ICE bit set to 0 (RIIC reset)

chanhvinh.com

18.3.3 RIICnMR1 — I²C Bus Mode Register 1

Access: RIICnMR1 is a 32-bit readable/writable register.
RIICnMR1L and RIICnMR1H are 16-bit readable/writable registers.
RIICnMR1LL, RIICnMR1LH, RIICnMR1HL, and RIICnMR1HH are 8-bit readable/writable registers.

Address: RIICnMR1: <RIICn_base> + 0008_H
RIICnMR1L: <RIICn_base> + 0008_H, RIICnMR1H: <RIICn_base> + $000A_H$
RIICnMR1LL: <RIICn_base> + 0008_H, RIICnMR1LH: <RIICn_base> + 0009_H, RIICnMR1HL: <RIICn_base> + $000A_H$,
RIICnMR1HH: <RIICn_base> + $000B_H$

Initial Value: 0000 0008_H This register is initialized by any reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	CKS[2:0]			BCWP	BC[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.8 RIICnMR1 register contents

Bit Position	Bit Name	Function
31 to 7	—	Reserved These bits are read as 0. The write value should be 0.
6 to 4	CKS[2:0]	Internal Reference Clock (IICϕ) Selection b6 b4 0 0 0: IICϕ = P0ϕ/1 0 0 1: IICϕ = P0ϕ/2 0 1 0: IICϕ = P0ϕ/4 0 1 1: IICϕ = P0ϕ/8 1 0 0: IICϕ = P0ϕ/16 1 0 1: IICϕ = P0ϕ/32 1 1 0: IICϕ = P0ϕ/64 1 1 1: IICϕ = P0ϕ/128
3	BCWP*1	BC Write Protect 0: Enables a value to be written in the BC[2:0] bits. (This bit is read as 1.)
2 to 0	BC[2:0]	Bit Counter b2 b0 0 0 0: 9 bits 0 0 1: 2 bits 0 1 0: 3 bits 0 1 1: 4 bits 1 0 0: 5 bits 1 0 1: 6 bits 1 1 0: 7 bits 1 1 1: 8 bits

Note 1. When rewriting the BC[2:0] bits, write 0 to the BCWP bit simultaneously.

chanhvinh.com

BC[2:0] Bits (Bit Counter)

These bits function as a counter that indicates the number of bits remaining to be transferred at the detection of a rising edge on the SCL line. Although these bits are writable and readable, it is not necessary to access these bits under normal conditions.

To write to these bits, specify the number of bits to be transferred plus one (data is transferred with an additional acknowledge bit) between transferred frames when the SCL line is at a low level.

The values of the BC[2:0] bits return to 000_B at the end of a data transfer including the acknowledge bit or when a start condition including a restart condition is detected.

chanhvinh.com

18.3.4 RIICnMR2 — I²C Bus Mode Register 2

Access: RIICnMR2 is a 32-bit readable/writable register.
RIICnMR2L and RIICnMR2H are 16-bit readable/writable registers.
RIICnMR2LL, RIICnMR2LH, RIICnMR2HL, and RIICnMR2HH are 8-bit readable/writable registers.

Address: RIICnMR2: <RIICn_base> + $000C_H$
RIICnMR2L: <RIICn_base> + $000C_H$, RIICnMR2H: <RIICn_base> + $000E_H$
RIICnMR2LL: <RIICn_base> + $000C_H$, RIICnMR2LH: <RIICn_base> + $000D_H$, RIICnMR2HL: <RIICn_base> + $000E_H$,
RIICnMR2HH: <RIICn_base> + $000F_H$

Initial Value: $0000\ 0006_H$ This register is initialized by any reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DLCS	SDDL[2:0]			—	TMOH	TMOL	TMOS
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 18.9 RIICnMR2 register contents (1/2)

Bit Position	Bit Name	Function
31 to 8	—	Reserved These bits are read as 0. The write value should be 0.
7	DLCS	SDA Output Delay Clock Source Selection 0: The internal reference clock (IICϕ) is selected as the clock source of the SDA output delay counter. 1: The internal reference clock divided by 2 (IICϕ/2) is selected as the clock source of the SDA output delay counter.*1
6 to 4	SDDL[2:0]	SDA Output Delay Counter • When RIICnMR2.DLCS = 0 (IICϕ) b6 b4 0 0 0: No output delay 0 0 1: 1 IICϕ cycle 0 1 0: 2 IICϕ cycles 0 1 1: 3 IICϕ cycles 1 0 0: 4 IICϕ cycles 1 0 1: 5 IICϕ cycles 1 1 0: 6 IICϕ cycles 1 1 1: 7 IICϕ cycles • When RIICnMR2.DLCS = 1 (IICϕ/2) b6 b4 0 0 0: No output delay 0 0 1: 1 or 2 IICϕ cycles 0 1 0: 3 or 4 IICϕ cycles 0 1 1: 5 or 6 IICϕ cycles 1 0 0: 7 or 8 IICϕ cycles
3	—	Reserved This bit is read as 0. The write value should be 0.
2	TMOH	Timeout H Count Control 0: Count is disabled while the SCL line is at a high level. 1: Count is enabled while the SCL line is at a high level.
1	TMOL	Timeout L Count Control 0: Count is disabled while the SCL line is at a low level. 1: Count is enabled while the SCL line is at a low level.

chanhvinh.com

Table 18.9 RIICnMR2 register contents (2/2)

Bit Position	Bit Name	Function
0	TMOS	Timeout Detection Time Selection 0: Long mode is selected. 1: Short mode is selected.

Note 1. The setting DLCS = 1 (IICϕ/2) only becomes valid when SCL is at the low level. When SCL is at the high level, the setting DLCS = 1 becomes invalid and the clock source becomes the internal reference clock (IICϕ).

TMOS Bit (Timeout Detection Time Selection)

This bit is used to select long mode or short mode for the timeout detection time when the timeout function is enabled (RIICnFER.TMOE bit = 1). When this bit is set to 0, long mode is selected. When this bit is set to 1, short mode is selected. In long mode, the timeout detection internal counter functions as a 16 bit-counter. In short mode, the counter functions as a 14 bit-counter. While the SCL line is in the state that enables this counter as specified by bits TMOH and TMOL, the counter counts up in synchronization with the internal reference clock (IICϕ) as a count source.

For details on the timeout function, see Section 18.13.1, Timeout Function.

TMOL Bit (Timeout L Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL line is held low when the timeout function is enabled (RIICnFER.TMOE bit = 1).

TMOH Bit (Timeout H Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL line is held high when the timeout function is enabled (RIICnFER.TMOE bit = 1).

SDDL[2:0] Bits (SDA Output Delay Setup Counter)

The SDA output can be delayed by the SDDL[2:0] setting. This counter works with the clock source selected by the DLCS bit. The setting of this function can be used for all types of SDA output, including the transmission of the acknowledge bit.

For details on this function, see Section 18.7, Facility for Delaying SDA Output.

CAUTION

Set the SDA output delay time to meet the I²C bus standard (within the data enable time/acknowledge enable time*1) or the SMBus standard (within the data hold time: 300 [ns] or more, and SCL-clock low-level period - the data setup time: 250 [ns]). Note that, if a value outside the standard is set, communication with communication devices may malfunction or it may seemingly become a start condition or stop condition depending on the bus state.

Note 1. Data enable time/acknowledge enable time
3,450 [ns] (0 to 100 [kbps]: standard mode (Sm))
900 [ns] (0 to 400 [kbps]: fast mode (Fm))

chanhvinh.com

18.3.5 RIICnMR3 — I²C Bus Mode Register 3

Access: RIICnMR3 is a 32-bit readable/writable register.
RIICnMR3L and RIICnMR3H are 16-bit readable/writable registers.
RIICnMR3LL, RIICnMR3LH, RIICnMR3HL, and RIICnMR3HH are 8-bit readable/writable registers.

Address: RIICnMR3: <RIICn_base> + 0010_H
RIICnMR3L: <RIICn_base> + 0010_H, RIICnMR3H: <RIICn_base> + 0012_H
RIICnMR3LL: <RIICn_base> + 0010_H, RIICnMR3LH: <RIICn_base> + 0011_H, RIICnMR3HL: <RIICn_base> + 0012_H,
RIICnMR3HH: <RIICn_base> + 0013_H

Initial Value: $0000\ 0000_H$ This register is initialized by any reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SMBE	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	W	R/W	R	R/W	R/W

Table 18.10 RIICnMR3 register contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved These bits are read as 0. The write value should be 0.
7	SMBE	SMBus/I²C Bus Selection 0: I²C bus selection 1: SMBus selection
6	WAIT*2	WAIT 0: No WAIT (The period between ninth clock cycle and first clock cycle is not held low.) 1: WAIT (The period between ninth clock cycle and first clock cycle is held low.) Low-hold is released by reading RIICnDRR.
5	RDRFS*2	RDRF Flag Set Timing Selection 0: The RDRF flag is set at the rising edge of the ninth SCL clock cycle. (The SCL line is not held low at the falling edge of the eighth clock cycle.) 1: The RDRF flag is set at the rising edge of the eighth SCL clock cycle. (The SCL line is held low at the falling edge of the eighth clock cycle.) Low-hold is released by writing a value to the ACKBT bit.
4	ACKWP*1	ACKBT Write Protect 0: Modification of the ACKBT bit is disabled. 1: Modification of the ACKBT bit is enabled.
3	ACKBT*1	Transmit Acknowledge 0: A 0 is sent as the acknowledge bit (ACK transmission). 1: A 1 is sent as the acknowledge bit (NACK transmission).
2	ACKBR	Receive Acknowledge 0: A 0 is received as the acknowledge bit (ACK reception). 1: A 1 is received as the acknowledge bit (NACK reception).
1, 0	NF[1:0]	Noise Filter Stage Selection b1 b0 0 0: Noise of up to one IICϕ cycle is filtered out (single-stage filter). 0 1: Noise of up to two IICϕ cycles is filtered out (2-stage filter). 1 0: Noise of up to three IICϕ cycles is filtered out (3-stage filter). 1 1: Noise of up to four IICϕ cycles is filtered out (4-stage filter).

Note 1. If it is attempted to write 1 to both ACKWP and ACKBT bits, the ACKBT bit cannot be set to 1.

chanhvinh.com

Note 2. The WAIT and RDRFS bits are valid only in receive mode (invalid in transmit mode).

NF[1:0] Bits (Noise Filter Stage Selection)

These bits are used to select the noise width that can be eliminated for the signal input to RIICnSCL or RIICnSDA pin.

ACKBR Bit (Receive Acknowledge)

This bit is used to store the acknowledge bit information received from the receive device in transmit mode.

[Setting condition]

When 1 is received as the acknowledge bit with the RIICnCR2.TRS bit set to 1

[Clearing conditions]

- When 0 is received as the acknowledge bit with the RIICnCR2.TRS bit set to 1
- When 1 is written to the RIICnCR1.IICRST bit while the RIICnCR1.ICE bit is 0 (RIIC reset)

ACKBT Bit (Transmit Acknowledge)

This bit is used to set the bit to be sent at the acknowledge timing in receive mode.

[Setting condition]

When 1 is written to this bit with the ACKWP bit set to 1

[Clearing conditions]

- When 0 is written to this bit with the ACKWP bit set to 1
- When stop condition issuance is detected
- When 1 is written to the RIICnCR1.IICRST bit while the RIICnCR1.ICE bit is 0 (RIIC reset)

CAUTION

The ACKBT bit must be written to while the ACKWP bit is 1. If the ACKBT bit is written to with the ACKWP bit cleared to 0, writing to the ACKBT bit is disabled.

ACKWP Bit (ACKBT Write Protect)

This bit is used to control the modification of the ACKBT bit.

RDRFS Bit (RDRF Flag Set Timing Selection)

This bit is used to select the RDRF flag set timing in receive mode and also to select whether to hold the SCL line low at the falling edge of the eighth SCL clock cycle.

When the RDRFS bit is 0, the SCL line is not held low at the falling edge of the eighth SCL clock cycle, and the RDRF flag is set to 1 at the rising edge of the ninth SCL clock cycle.

When the RDRFS bit is 1, the RDRF flag is set to 1 at the rising edge of the eighth SCL clock cycle and the SCL line is held low at the falling edge of the eighth SCL clock cycle. The low-hold of the SCL line is released by writing a value to the ACKBT bit.

After data is received with this setting, the SCL line is automatically held low before the acknowledge bit is sent. This enables processing to send ACK (ACKBT = 0) or NACK (ACKBT = 1) according to receive data.

chanhvinh.com

WAIT Bit (WAIT)

This bit is used to control whether to hold the period between the ninth SCL clock cycle and the first SCL clock cycle low until the receive data buffer (RIICnDRR) is completely read each time single-byte data is received in receive mode.

When the WAIT bit is 0, the receive operation is continued without holding the period between the ninth and the first SCL clock cycle low. When both the RDRFS and WAIT bits are 0, continuous receive operation is enabled with the double buffer.

When the WAIT bit is 1, the SCL line is held low from the falling edge of the ninth clock cycle until the RIICnDRR value is read each time single-byte data is received. This enables receive operation in byte units.

CAUTION

When the value of the WAIT bit is to be read, be sure to read the RIICnDRR beforehand.

SMBE Bit (SMBus Select)

Setting this bit to 1 enables the RIICnSER.HOAE bit.

chanhvinh.com

18.3.6 RIICnFER — I²C Bus Function Enable Register

Access: RIICnFER is a 32-bit readable/writable register.
RIICnFERL and RIICnFERH are 16-bit readable/writable registers.
RIICnFERLL, RIICnFERLH, RIICnFERHL, and RIICnFERHH are 8-bit readable/writable registers.

Address: RIICnFER: <RIICn_base> + 0014_H
RIICnFERL: <RIICn_base> + 0014_H, RIICnFERH: <RIICn_base> + 0016_H
RIICnFERLL: <RIICn_base> + 0014_H, RIICnFERLH: <RIICn_base> + 0015_H, RIICnFERHL: <RIICn_base> + 0016_H,
RIICnFERHH: <RIICn_base> + 0017_H

Initial Value: $0000\ 0072_H$ This register is initialized by any reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
Initial value	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.11 RIICnFER register contents

Bit Position	Bit Name	Function
31 to 7	—	Reserved These bits are read as 0. The write value should be 0.
6	SCLE	SCL Synchronous Circuit Enable 0: No SCL synchronous circuit is used. 1: An SCL synchronous circuit is used.
5	NFE	Digital Noise Filter Circuit Enable 0: No digital noise filter circuit is used. 1: A digital noise filter circuit is used.
4	NACKE	NACK Reception Transfer Suspension Enable 0: Transfer operation is not suspended during NACK reception (transfer suspension disabled). 1: Transfer operation is suspended during NACK reception (transfer suspension enabled).
3	SALE	Slave Arbitration-Lost Detection Enable 0: Slave arbitration-lost detection is disabled. 1: Slave arbitration-lost detection is enabled.
2	NALE	NACK Transmission Arbitration-Lost Detection Enable 0: NACK transmission arbitration-lost detection is disabled. 1: NACK transmission arbitration-lost detection is enabled.
1	MALE	Master Arbitration-Lost Detection Enable 0: Master arbitration-lost detection is disabled. (Disables the arbitration-lost detection function and does not clear the RIICnCR2.MST and TRS bits automatically when arbitration is lost.) 1: Master arbitration-lost detection is enabled. (Enables the arbitration-lost detection function and clears the RIICnCR2.MST and TRS bits automatically when arbitration is lost.)
0	TMOE	Timeout Function Enable 0: The timeout function is disabled. 1: The timeout function is enabled.

chanhvinh.com

TMOE Bit (Timeout Function Enable)

This bit is used to enable or disable the timeout function.

For details on the timeout function, see Section 18.13.1, Timeout Function.

MALE Bit (Master Arbitration-Lost Detection Enable)

This bit is used to specify whether to use the arbitration-lost detection function in master mode. Normally, set this bit to 1.

NALE Bit (NACK Transmission Arbitration-Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when ACK is detected during transmission of NACK in receive mode (such as when slaves with the same address exist on the bus or when two or more masters select the same slave device simultaneously with different number of receive bytes).

SALE Bit (Slave Arbitration-Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when a value different from the value being transmitted is detected on the bus in slave transmit mode (such as when slaves with the same address exist on the bus or when a mismatch with the transmit data occurs due to noise).

NACKE Bit (NACK Reception Transfer Suspension Enable)

This bit is used to specify whether to continue or discontinue the transfer operation when NACK is received in transmit mode. Normally, set this bit to 1.

When NACK is received with the NACKE bit set to 1, the next transfer operation is suspended.

When the NACKE bit is 0, the next transfer operation is continued regardless of the received acknowledge content.

SCLE Bit (SCL Synchronous Circuit Enable)

This bit is used to specify whether to synchronize the SCL clock with the SCL input clock. Normally, set this bit to 1.

When the SCLE bit is cleared to 0 (SCL synchronous circuit not used), the RIIC does not synchronize the SCL clock with the SCL input clock. In this setting, the RIIC outputs the SCL clock with the transfer rate set in RIICnBRH and RIICnBRL regardless of the SCL line state. For this reason, if the bus load of the I^2C bus line is much larger than the specification value or if the SCL clock output overlaps in multiple masters, the short-cycle SCL clock that does not meet the specification may be output. When no SCL synchronous circuit is used, it also affects the issuance of a start condition, restart condition, and stop condition, and the continuous output of extra SCL clock cycles.

This bit must not be cleared to 0 except for checking the output of the transfer rate.

chanhvinh.com

18.3.7 RIICnSER — I²C Bus Status Enable Register

Access: RIICnSER is a 32-bit readable/writable register.
RIICnSERL and RIICnSERH are 16-bit readable/writable registers.
RIICnSERLL, RIICnSERLH, RRIICnSERHL, and RIICnSERHH are 8-bit readable/writable registers.

Address: RIICnSER: <RIICn_base> + 0018_H
RIICnSERL: <RIICn_base> + 0018_H, RIICnSERH: <RIICn_base> + $001A_H$
RIICnSERLL: <RIICn_base> + 0018_H, RIICnSERLH: <RIICn_base> + 0019_H, RIICnSERHL: <RIICn_base> + $001A_H$,
RIICnSERHH: <RIICn_base> + $001B_H$

Initial Value: $0000\ 0009_H$ This register is initialized by any reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	HOAE	—	DIDE	—	GCE	SAR2	SAR1	SAR0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
R/W	R	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R/W	R/W	R/W

Table 18.12 RIICnSER register contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved These bits are read as 0. The write value should be 0.
7	HOAE	Host Address Enable 0: Host address detection is disabled. 1: Host address detection is enabled.
6	—	Reserved This bit is read as 0. The write value should be 0.
5	DIDE	Device-ID Address Detection Enable 0: Device-ID address detection is disabled. 1: Device-ID address detection is enabled.
4	—	Reserved This bit is read as 0. The write value should be 0.
3	GCE	General Call Address Enable 0: General call address detection is disabled. 1: General call address detection is enabled.
2	SAR2	Slave Address Register 2 Enable 0: Slave address in RIICnSAR2 is disabled. 1: Slave address in RIICnSAR2 is enabled.
1	SAR1	Slave Address Register 1 Enable 0: Slave address in RIICnSAR1 is disabled. 1: Slave address in RIICnSAR1 is enabled.
0	SAR0	Slave Address Register 0 Enable 0: Slave address in RIICnSAR0 is disabled. 1: Slave address in RIICnSAR0 is enabled.

chanhvinh.com

SARy Bit (Slave Address Register y Enable) (y = 0 to 2)

This bit is used to enable or disable the received slave address and the slave address set in RIICnSARy.

When this bit is set to 1, the slave address set in RIICnSARy is enabled and is compared with the received slave address.

When this bit is cleared to 0, the slave address set in RIICnSARy is disabled and is ignored even if it matches the received slave address.

GCE Bit (General Call Address Enable)

This bit is used to specify whether to ignore the general call address ($0000\ 000_B$ + 0 [W]: All 0) when it is received.

When this bit is set to 1, if the received slave address matches the general call address, the RIIC recognizes the received slave address as the general call address independently of the slave addresses set in RIICnSARy (y = 0 to 2) and performs data receive operation.

When this bit is cleared to 0, the received slave address is ignored even if it matches the general call address.

DIDE Bit (Device-ID Address Detection Enable)

This bit is used to specify whether to recognize and execute the Device-ID address when a device ID ($1111\ 100_B$) is received in the first frame after a start condition or restart condition is detected.

When this bit is set to 1, if the received first frame matches the device ID, the RIIC recognizes that the Device-ID address has been received. When the following R/W# bit is 0 [W], the RIIC recognizes the second and the following frames as slave addresses and continues the receive operation.

When this bit is cleared to 0, the RIIC ignores the received first frame even if it matches the device ID address and recognizes the first frame as a normal slave address.

For details on the device-ID address detection, see **Section 18.9.3, Device-ID Address Detection**.

HOAE Bit (Host Address Enable)

This bit is used to specify whether to ignore received host address ($0001\ 000_B$) when the RIICnMR3.SMBS bit is 1.

When this bit is set to 1 while the RIICnMR3.SMBS bit is 1, if the received slave address matches the host address, the RIIC recognizes the received slave address as the host address independently of the slave addresses set in RIICnSARy (y = 0 to 2) and performs the receive operation.

When the RIICnMR3.SMBS bit or the HOAE bit is cleared to 0, the received slave address is ignored even if it matches the host address.

chanhvinh.com

18.3.8 RIICnIER — I²C Bus Interrupt Enable Register

Access: RIICnIER is a 32-bit readable/writable register.
RIICnIERL and RIICnIERH are 16-bit readable/writable registers.
RIICnIERLL, RIICnIERLH, RIICnIERHL, and RIICnIERHH are 8-bit readable/writable registers.

Address: RIICnIER: <RIICn_base> + $001C_H$
RIICnIERL: <RIICn_base> + $001C_H$, RIICnIERH: <RIICn_base> + $001E_H$
RIICnIERLL: <RIICn_base> + $001C_H$, RIICnIERLH: <RIICn_base> + $001D_H$, RIICnIERHL: <RIICn_base> + $001E_H$,
RIICnIERHH: <RIICn_base> + $001F_H$

Initial Value: $0000\ 0000_H$ This register is initialized by any reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.13 RIICnIER register contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved These bits are read as 0. The write value should be 0.
7	TIE	Transmit Data Empty Interrupt Enable 0: Transmit data empty interrupt (INTRIICTI) request is disabled. 1: Transmit data empty interrupt (INTRIICTI) request is enabled.
6	TEIE	Transmit End Interrupt Enable 0: Transmit end interrupt (INTRIICTEI) request is disabled. 1: Transmit end interrupt (INTRIICTEI) request is enabled.
5	RIE	Receive Data Full Interrupt Enable 0: Receive data full interrupt (INTRIICRI) request is disabled. 1: Receive data full interrupt (INTRIICRI) request is enabled.
4	NAKIE	NACK Reception Interrupt Enable 0: NACK reception interrupt (INTRIICNAKI) request is disabled. 1: NACK reception interrupt (INTRIICNAKI) request is enabled.
3	SPIE	Stop Condition Detection Interrupt Enable 0: Stop condition detection interrupt (INTRIICSPI) request is disabled. 1: Stop condition detection interrupt (INTRIICSPI) request is enabled.
2	STIE	Start Condition Detection Interrupt Enable 0: Start condition detection interrupt (INTRIICSTI) request is disabled. 1: Start condition detection interrupt (INTRIICSTI) request is enabled.
1	ALIE	Arbitration-Lost Interrupt Enable 0: Arbitration-lost interrupt (INTRIICALI) request is disabled. 1: Arbitration-lost interrupt (INTRIICALI) request is enabled.
0	TMOIE	Timeout Interrupt Enable 0: Timeout interrupt (INTRIICTMOI) request is disabled. 1: Timeout interrupt (INTRIICTMOI) request is enabled.

chanhvinh.com

TMOIE Bit (Timeout Interrupt Enable)

This bit is used to enable or disable timeout interrupt (INTRIICTMOI) requests when the RIICnSR2.TMOF flag is set to 1. An INTRIICTMOI interrupt request is canceled by clearing the TMOF flag or the TMOIE bit to 0.

ALIE Bit (Arbitration-Lost Interrupt Enable)

This bit is used to enable or disable arbitration-lost interrupt (INTRIICALI) requests when the RIICnSR2.AL flag is set to 1. An INTRIICALI interrupt request is canceled by clearing the AL flag or the ALIE bit to 0.

STIE Bit (Start Condition Detection Interrupt Enable)

This bit is used to enable or disable start condition detection interrupt (INTRIICSTI) requests when the RIICnSR2.START flag is set to 1. An INTRIICSTI interrupt request is canceled by clearing the START flag or the STIE bit to 0.

SPIE Bit (Stop Condition Detection Interrupt Enable)

This bit is used to enable or disable stop condition detection interrupt (INTRIICSPI) requests when the RIICnSR2.STOP flag is set to 1. An INTRIICSPI interrupt request is canceled by clearing the STOP flag or the SPIE bit to 0.

NAKIE Bit (NACK Reception Interrupt Enable)

This bit is used to enable or disable NACK reception interrupt (INTRIICNAKI) requests when the RIICnSR2.NACKF flag is set to 1. An INTRIICNAKI interrupt request is canceled by clearing the NACKF flag or the NAKIE bit to 0.

RIE Bit (Receive Data Full Interrupt Enable)

This bit is used to enable or disable receive data full interrupt (INTRIICRI) requests when the RIICnSR2.RDRF flag is set to 1. An INTRIICRI interrupt request is canceled by clearing the RDRF flag or the RIE bit to 0.

TEIE Bit (Transmit End Interrupt Enable)

This bit is used to enable or disable transmit end interrupts (INTRIICTEI) when the RIICnSR2.TEND flag is set to 1. An INTRIICTEI interrupt request is canceled by clearing the TEND flag or the TEIE bit to 0.

TIE Bit (Transmit Data Empty Interrupt Enable)

This bit is used to enable or disable transmit data empty interrupts (INTRIICTI) when the RIICnSR2.TDRE flag is set to 1.

chanhvinh.com

18.3.9 RIICnSR1 — I²C Bus Status Register 1

Access: RIICnSR1 is a 32-bit readable/writable register.
RIICnSR1L and RIICnSR1H are 16-bit readable/writable registers.
RIICnSR1LL, RIICnSR1LH, RIICnSR1HL, and RIICnSR1HH are 8/1-bit readable/writable registers.

Address: RIICnSR1: <RIICn_base> + 0020_H
RIICnSR1L: <RIICn_base> + 0020_H, RIICnSR1H: <RIICn_base> + 0022_H
RIICnSR1LL: <RIICn_base> + 0020_H, RIICnSR1LH: <RIICn_base> + 0021_H, RIICnSR1HL: <RIICn_base> + 0022_H,
RIICnSR1HH: <RIICn_base> + 0023_H

Initial Value: $0000\ 0000_H$ This register is initialized by any reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	HOA	—	DID	—	GCA	AAS2	AAS1	AAS0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R(/W) *1	R	R(/W) *1	R	R/W *1	R(/W) *1	R(/W) *1	R(/W) *1

Note 1. Only 0 can be written to this bit.

Table 18.14 RIICnSR1 register contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved These bits are read as 0. The write value should be 0.
7	HOA	Host Address Detection Flag 0: Host address is not detected. 1: Host address is detected.
6	—	Reserved This bit is read as 0. The write value should be 0.
5	DID	Device-ID Address Detection Flag 0: Device-ID command is not detected. 1: Device-ID command is detected.
4	—	Reserved This bit is read as 0. The write value should be 0.
3	GCA	General Call Address Detection Flag 0: General call address is not detected. 1: General call address is detected.
2	AAS2	Slave Address 2 Detection Flag 0: Slave address 2 is not detected. 1: Slave address 2 is detected.
1	AAS1	Slave Address 1 Detection Flag 0: Slave address 1 is not detected. 1: Slave address 1 is detected.
0	AAS0	Slave Address 0 Detection Flag 0: Slave address 0 is not detected. 1: Slave address 0 is detected.

chanhvinh.com

AASy Flag (Slave Address y Detection) (y = 0 to 2)

[Setting conditions]

<For 7-bit address format: RIICnSARy.FSy = 0>

When the received slave address matches the RIICnSARy.SVA[7:1] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

<For 10-bit address format: RIICnSARy.FSy = 1>

When the received slave address matches a value of (1111 0_B + RIICnSARy.SVA[9:8]) and the following address matches the RIICnSARy.SVA[7:0] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the AASy bit after reading AASy = 1
- When a stop condition is detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

<For 7-bit address format: RIICnSARy.FSy = 0>

- When the received slave address does not match the RIICnSARy.SVA[7:1] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

<For 10-bit address format: RIICnSARy.FSy = 1>

- When the received slave address does not match a value of (1111 0_B + RIICnSARy.SVA[9:8]) with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When the received slave address matches a value of (1111 0_B + RIICnSARy.SVA[9:8]) and the following address does not match the RIICnSARy.SVA[7:0] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

GCA Flag (General Call Address Detection)

[Setting condition]

When the received slave address matches the general call address (0000 000_B + 0 [W]) with the RIICnSER.GCE bit set to 1 (general call address detection enabled)
This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the GCA bit after reading GCA = 1
- When a stop condition is detected
- When the received slave address does not match the general call address (0000 000_B + 0 [W]) with the RIICnSER.GCE bit set to 1 (general call address detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

chanhvinh.com

DID Flag (Device-ID Address Detection)

[Setting condition]

- When the first frame received immediately after a start condition or restart condition is detected matches a value of (device ID ($1111\ 100_B$) + 0 [W]) with the RIICnSER.DIDE bit set to 1 (Device-ID address detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.
- When a restart condition is detected after a match with the device ID address and the device ID address ($1111\ 100_B$) + 1 [R] has matched with the RIICnSER.DIDE bit set to 1 (Device-ID address detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the DID bit after reading DID = 1
- When a stop condition is detected
- When the first frame received immediately after a start condition or restart condition is detected does not match a value of (device ID ($1111\ 100_B$)) with the RIICnSER.DIDE bit set to 1 (Device-ID address detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When the first frame received immediately after a start condition or restart condition is detected matches a value of (device ID ($1111\ 100_B$) + 0 [W]) and the second frame does not match any of slave addresses 0 to 2 with the RIICnSER.DIDE bit set to 1 (Device-ID address detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

HOA Flag (Host Address Detection)

[Setting condition]

When the received slave address matches the host address ($0001\ 000_B$) while the RIICnMR3.SMBE bit and RIICnSER.HOAE bit are set to 1 (host address detection enabled)
This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the HOA bit after reading HOA = 1
- When a stop condition is detected
- When 0 is written to the RIICnMR3.SMBE bit or the RIICnSER.HOAE bit
- When the received slave address does not match the host address ($0001\ 000_B$) with the RIICnSER.HOAE bit set to 1 (host address detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

chanhvinh.com

18.3.10 RIICnSR2 — I²C Bus Status Register 2

Access: RIICnSR2 is a 32-bit readable/writable register.
RIICnSR2L and RIICnSR2H are 16-bit readable/writable registers.
RIICnSR2LL, RIICnSR2LH, RIICnSR2HL, and RIICnSR2HH are 8/1-bit readable/writable registers.

Address: RIICnSR2: <RIICn_base> + 0024_H
RIICnSR2L: <RIICn_base> + 0024_H, RIICnSR2H: <RIICn_base> + 0026_H
RIICnSR2LL: <RIICn_base> + 0024_H, RIICnSR2LH: <RIICn_base> + 0025_H, RIICnSR2HL: <RIICn_base> + 0026_H,
RIICnSR2HH: <RIICn_base> + 0027_H

Initial Value: $0000\ 0000_H$ This register is initialized by any reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R(/W)*1	R(/W)*1	R(/W)*1	R(/W)*1	R(/W)*1	R(/W)*1	R(/W)*1

Note 1. Only 0 can be written to this bit.

Table 18.15 RIICnSR2 register contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved These bits are read as 0. The write value should be 0.
7	TDRE	Transmit Data Empty Flag 0: RIICnDRT contains transmit data. 1: RIICnDRT contains no transmit data.
6	TEND	Transmit End Flag 0: Data is being transmitted. 1: Data has been transmitted.
5	RDRF	Receive Data Full Flag 0: RIICnDRR contains no receive data. 1: RIICnDRR contains receive data.
4	NACKF	NACK Receive Flag 0: NACK is not received. 1: NACK is received.
3	STOP	Stop Condition Detection Flag 0: Stop condition is not detected. 1: Stop condition is detected.
2	START	Start Condition Detection Flag 0: Start condition is not detected. 1: Start condition is detected.
1	AL	Arbitration-Lost Flag 0: Arbitration is not lost. 1: Arbitration is lost.
0	TMOF	Timeout Flag 0: Timeout has not occurred. 1: Timeout occurred.

chanhvinh.com

TMOF Flag (Timeout)

This flag is set to 1 when the RIIC recognizes timeout after the SCL line state remains unchanged for a certain period.

[Setting condition]

The timeout function is enabled when the RIICnFER.TMOE bit is 1. It detects an abnormal bus state that the SCL line is held low or high during the following conditions:

- The bus is busy (RIICnCR2.BBSY = 1) in master mode (RIICnCR2.MST = 1).
- The slave address matches that of this module (RIICnSR1 register is not 00_H) and the bus is busy (RIICnCR2.BBSY = 1) in slave mode (RIICnCR2.MST = 0).
- Issuing of a start condition is being requested (RIICnCR2.ST = 1) and the bus is free (RIICnCR2.BBSY = 0).

[Clearing conditions]

- When 0 is written to the TMOF bit after reading TMOF = 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

AL Flag (Arbitration-Lost)

This flag shows that bus mastership has been lost (loss in arbitration) due to a bus conflict or some other reason when a start condition is issued or an address and data are transmitted. The RIIC monitors the level on the SDA line during transmission and, if the level on the line does not match the value of the bit being output, sets the value of the AL bit to 1 to indicate that the bus is occupied by another device.

The RIIC can also set the flag to indicate the detection of loss of arbitration during NACK transmission in receive mode or during data transmission in slave mode.

[Setting conditions]

<When master arbitration-lost detection is enabled: RIICnFER.MALE = 1>

- When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock except for the ACK period during data (including slave address) transmission in master transmit mode (when the SDA line is driven low while the internal SDA output is at a high level (the SDA pin is in the high-impedance state))
- When a start condition is detected while the RIICnCR2.ST bit is 1 (start condition issuance request) or the internal SDA output state does not match the SDA line level
- When the RIICnCR2.ST bit is set to 1 (start condition issuance request) with the RIICnCR2.BBSY flag set to 1.

<When NACK arbitration-lost detection is enabled: RIICnFER.NALE = 1>

When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock in the ACK period during NACK transmission in receive mode

<When slave arbitration-lost detection is enabled: RIICnFER.SALE = 1>

When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock except for the ACK period during data transmission in slave transmit mode

[Clearing conditions]

- When 0 is written to the AL bit after reading AL = 1

chanhvinh.com

- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

Table 18.16 Relationship between Arbitration-Lost Generation Sources and Arbitration-Lost Enable Functions

RIICnFER			RIICnSR2		
MALE	NALE	SALE	AL	Error	Arbitration-Lost Generation Source
1	x	x	1	Start condition issuance error	When internal SDA output state does not match SDA line level when a start condition is detected while the RIICnCR2.ST bit is 1
					When RIICnCR2.ST is set to 1 with RIICnCR2.BBSY set to 1
			1	Transmit data mismatch	When transmit data (including slave address) does not match the bus state in master transmit mode
x	1	x	1	NACK transmission mismatch	When ACK is detected during transmission of NACK in master receive mode or slave receive mode
x	x	1	1	Transmit data mismatch	When transmit data does not match the bus state in slave transmit mode

x: Don't care

START Flag (Start Condition Detection)

[Setting condition]

When a start condition (or a restart condition) is detected

[Clearing conditions]

- When 0 is written to the START bit after reading START = 1
- When a stop condition is detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

STOP Flag (Stop Condition Detection)

[Setting condition]

When a stop condition is detected

[Clearing conditions]

- When 0 is written to the STOP bit after reading STOP = 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

NACKF Flag (NACK Receive)

[Setting condition]

When acknowledge is not received (NACK is received) from the receive device in transmit mode with the RIICnFER.NACKE bit set to 1 (transfer suspension enabled)

[Clearing conditions]

- When 0 is written to the NACKF bit after reading NACKF = 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

chanhvinh.com

CAUTION

When the NACKF flag is set to 1, the RIIC suspends data transmission/reception. Writing to RIICnDRT in transmit mode or reading from RIICnDRR in receive mode with the NACKF flag set to 1 does not enable data transmit/receive operation. To restart data transmission/reception, clear the NACKF flag to 0.

RDRF Flag (Receive Data Full)

[Setting conditions]

- Slave receive mode
  - When the received slave address matches and the RIICnCR2.TRS bit is cleared to 0 after a start condition (or a restart condition) is detected
  - At the rising edge of the eighth or ninth SCL clock cycle (selected by the RIICnMR3.RDRFS bit) after receive data is transferred from RIICnDRS to RIICnDRR
- Master receive mode
  - When the slave address and the data direction are transmitted and the receive mode is entered (the RIICnCR2.TRS bit is set to 1) after a start condition (or a restart condition) is issued
  - At the rising edge of the eighth or ninth SCL clock cycle (selected by the RIICnMR3.RDRFS bit) after receive data is transferred from RIICnDRS to RIICnDRR

[Clearing conditions]

- When 0 is written to the RDRF bit after reading RDRF = 1
- When data is read from RIICnDRR
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

TEND Flag (Transmit End)

[Setting condition]

At the rising edge of the ninth SCL clock cycle while the TDRE flag is 1

[Clearing conditions]

- When 0 is written to the TEND bit after reading TEND = 1
- When data is written to RIICnDRT
- When a stop condition is detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

TDRE Flag (Transmit Data Empty)

[Setting conditions]

- When data has been transferred from RIICnDRT to RIICnDRS and RIICnDRT becomes empty
- When the RIICnCR2.TRS bit is set to 1
  - When the RIICnCR2.MST bit is set to 1 after a start condition (or a restart condition) is detected
  - When the RIIC enters transmit mode from receive mode
- When the received slave address matches while the TRS bit is 1

chanhvinh.com

[Clearing conditions]

- When data is written to RIICnDRT
- When the RIICnCR2.TRS bit is cleared to 0
  - When a stop condition is detected
  - When the RIIC enters receive mode from transmit mode
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

CAUTION

When the NACKF flag is set to 1 while the RIICnFER.NACKE bit is 1, the RIIC suspends data transmission/reception. Here, if the TDRE flag is 0 (next transmit data has been written), data is transferred to the RIICnDRS register and the RIICnDRT register becomes empty at the rising edge of the ninth clock cycle, but the TDRE flag is not set to 1.

chanhvinh.com

18.3.11 RIICnSARy — I²C Slave Address Register y (y = 0 to 2)

Access: RIICnSARy is a 32-bit readable/writable register.
RIICnSARyL and RIICnSARyH are 16-bit readable/writable registers.
RIICnSARyLL, RIICnSARyLH, RIICnSARyHL, and RIICnSARyHH are 8-bit readable/writable registers.

Address: RIICnSAR0: <RIICn_base> + 0028_H
RIICnSAR0L: <RIICn_base> + 0028_H, RIICnSAR0H: <RIICn_base> + $002A_H$
RIICnSAR0LL: <RIICn_base> + 0028_H, RIICnSAR0LH: <RIICn_base> + 0029_H, RIICnSAR0HL: <RIICn_base> + $002A_H$,
RIICnSAR0HH: <RIICn_base> + $002B_H$
RIICnSAR1: <RIICn_base> + $002C_H$
RIICnSAR1L: <RIICn_base> + $002C_H$, RIICnSAR1H: <RIICn_base> + $002E_H$
RIICnSAR1LL: <RIICn_base> + $002C_H$, RIICnSAR1LH: <RIICn_base> + $002D_H$, RIICnSAR1HL: <RIICn_base> + $002E_H$,
RIICnSAR1HH: <RIICn_base> + $002F_H$
RIICnSAR2: <RIICn_base> + 0030_H
RIICnSAR2L: <RIICn_base> + 0030_H, RIICnSAR2H: <RIICn_base> + 0032_H
RIICnSAR2LL: <RIICn_base> + 0030_H, RIICnSAR2LH: <RIICn_base> + 0031_H, RIICnSAR2HL: <RIICn_base> + 0032_H,
RIICnSAR2HH: <RIICn_base> + 0033_H

Initial Value: $0000\ 0000_H$ This register is initialized by any reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FSy	—	—	—	—	—	SVA[9:1]									SVA0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.17 RIICnSARy register contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved These bits are read as 0. The write value should be 0.
15	FSy	7-Bit/10-Bit Address Format Selection 0: The 7-bit address format is selected. 1: The 10-bit address format is selected.
14 to 10	—	Reserved These bits are read as 0. The write value should be 0.
9 to 1	SVA[9:1]	7-Bit Address/10-Bit Address Upper Bits A slave address is set. • When the FSy bit is 0 (7-bit address format), the SVA[7:1] bits are Valid and form a 7-bit slave address. • When the FSy bit is 1 (10-bit address format), SVA[9:1] bits form a 10-bit slave address (combined with the SVA0 bit).
0	SVA0	10-Bit Address LSB The least significant bit (LSB) of a 10-bit slave address is set. • When the FSy bit is 0 (7-bit address format), this bit is invalid. • When the FSy bit is 1 (10-bit address format), this bit is a 10-bit slave address (combined with the SVA[9:1] bits).

SVA0 Bit (10-Bit Address LSB)

When the 10-bit address format is selected (RIICnSARy.FSy = 1), this bit functions as the LSB of a 10-bit address and forms a 10-bit address in combination with the SVA[9:1] bits.
When the RIICnSER.SARy bit is set to 1 (RIICnSARy enabled) and the RIICnSARy.FSy bit is 1, this bit is valid. While the RIICnSARy.FSy bit or SARy bit is 0, the setting of this bit is ignored.

chanhvinh.com

SVA[9:1] Bits (7-Bit Address/10-Bit Address Upper Bits)

When the 7-bit address format is selected (RIICnSARy.FSy = 0), these bits function as a 7-bit address. When the 10-bit address format is selected (RIICnSARy.FSy = 1), these bits function as a 10-bit address in combination with the SVA0 bit.
While the RIICnSER.SARy bit is 0, the setting of these bits is ignored.

FSy Bit (7-Bit/10-Bit Address Format Selection)

This bit is used to select 7-bit address or 10-bit address for slave address y (in RIICnSARy).

When the RIICnSER.SARy bit is set to 1 (RIICnSARy enabled) and the RIICnSARy.FSy bit is 0, the 7-bit address format is selected for slave address y, the RIICnSARy.SVA[7:1] setting is valid, and the settings of the SVA[9:8] bits and the RIICnSARy.SVA0 bit are ignored.

When the RIICnSER.SARy bit is set to 1 (RIICnSARy enabled) and the RIICnSARy.FSy bit is 1, the 10-bit address format is selected for slave address y and the settings of the SVA[9:1] bits and the SVA0 bit are valid.

While the RIICnSER.SARy bit is 0 (RIICnSARy disabled), the setting of the RIICnSARy.FSy bit is invalid.

chanhvinh.com

18.3.12 RIICnBRL — I²C Bus Bit Rate Low-Level Register

Access: RIICnBRL is a 32-bit readable/writable register.
RIICnBRLL and RIICnBRLH are 16-bit readable/writable registers.
RIICnBRLLL, RIICnBRLLH, RIICnBRLHL, and RIICnBRLHH are 8-bit readable/writable registers.

Address: RIICnBRL: <RIICn_base> + 0034_H
RIICnBRLL: <RIICn_base> + 0034_H, RIICnBRLH: <RIICn_base> + 0036_H
RIICnBRLLL: <RIICn_base> + 0034_H, RIICnBRLLH: <RIICn_base> + 0035_H, RIICnBRLHL: <RIICn_base> + 0036_H,
RIICnBRLHH: <RIICn_base> + 0037_H

Initial Value: 0000 $00FF_H$ This register is initialized by any reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	BRL[4:0]				
Initial value	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 18.18 RIICnBRL register contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved These bits are read as 0. The write value should be 0.
7 to 5	—	Reserved These bits are read as 1. The write value should be 1.
4 to 0	BRL[4:0]	Bit Rate Low-Level Period Low-level period of SCL clock

The RIICnBRL register is a 5-bit register that is used to set the width at low level for the SCL clock.

It also works to generate the data setup time for automatic SCL low-hold operation (see Section 18.10, Automatically Low-Hold Function for SCL); when the RIIC is used only in slave mode, this register needs to be set to a value equal to or longer than the data setup time*1.

RIICnBRL counts the low-level period with the internal reference clock source (IICϕ) specified by the RIICnMR1.CKS[2:0] bits.

Note 1. Data setup time ($t_{SU: DAT}$)
250 [ns] (0 to 100 [kbps]: standard mode (Sm))
100 [ns] (0 to 400 [kbps]: fast mode (Fm))

chanhvinh.com

18.3.13 RIICnBRH — I²C Bus Bit Rate High-Level Register

Access: RIICnBRH is a 32-bit readable/writable register.
RIICnBRHL and RIICnBRHH are 16-bit readable/writable registers.
RIICnBRHLL, RIICnBRHLH, RIICnBRHHL, and RIICnBRHHH are 8-bit readable/writable registers.

Address: RIICnBRH: <RIICn_base> + 0038_H
RIICnBRHL: <RIICn_base> + 0038_H, RIICnBRHH: <RIICn_base> + $003A_H$
RIICnBRHLL: <RIICn_base> + 0038_H, RIICnBRHLH: <RIICn_base> + 0039_H, RIICnBRHHL: <RIICn_base> + $003A_H$,
RIICnBRHHH:<RIICn_base> + $003B_H$

Initial Value: 0000 $00FF_H$ This register is initialized by any reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	BRH[4:0]				
Initial value	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 18.19 RIICnBRH register contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved These bits are read as 0. The write value should be 0.
7 to 5	—	Reserved These bits are read as 1. The write value should be 1.
4 to 0	BRH[4:0]	Bit Rate High-Level Period High-level period of SCL clock

RIICnBRH is a 5-bit register to set the high-level period of SCL clock. RIICnBRH is valid in master mode. If the RIIC is used only in slave mode, this register need not to set the high-level period.

RIICnBRH counts the high-level period with the internal reference clock source (IICϕ) specified by the RIICnMR1.CKS[2:0] bits.

The frequency and duty cycle are calculated using one of the following expressions (1) to (5) according to the register settings.

CAUTION

The minimum value that can be specified in RIICnBRL and RIICnBRH is determined according to the values of the SCLE and NFE bits in RIICnFER and the NF bit in RIICnMR3. For details of the minimum specifiable value, see Table 18.20.

(1) When SCLE = 0

Frequency = 1 / {[(BRH + 1) + (BRL + 1)] / IICϕ + tr + tf}

Duty cycle = {tr + (BRH + 1) / IICϕ} / {tr + tf + [(BRH + 1) + (BRL + 1)] / IICϕ}

chanhvinh.com

(2) When SCLE = 1, NFE = 0, CKS = 000 (IICϕ = P0ϕ)
Frequency = 1 / {[(BRH + 3) + (BRL + 3)] / IICϕ + tr + tf}
Duty cycle = {tr + (BRH + 3) / IICϕ} / {tr + tf + [(BRH + 3) + (BRL + 3)] / IICϕ}

(3) When SCLE = 1, NFE = 1, CKS = 000 (IICϕ = P0ϕ)
Frequency = 1 / {[(BRH + 3 + nf) + (BRL + 3 + nf)] / IICϕ + tr + tf}
Duty cycle = {tr + (BRH + 3 + nf) / IICϕ} / {tr + tf + [(BRH + 3 + nf) + (BRL + 3 + nf)] / IICϕ}

(4) When SCLE = 1, NFE = 0, CKS ≠ 000 (IICϕ < P0ϕ)
Frequency = 1 / {[(BRH + 2) + (BRL + 2)] / IICϕ + tr + tf}
Duty cycle = {tr + (BRH + 2) / IICϕ} / {tr + tf + [(BRH + 2) + (BRL + 2)] / IICϕ}

(5) When SCLE = 1, NFE = 1, CKS ≠ 000 (IICϕ < P0ϕ)
Frequency = 1 / {[(BRH + 2 + nf) + (BRL + 2 + nf)] / IICϕ + tr + tf}
Duty cycle = {tr + (BRH + 2 + nf) / IICϕ} / {tr + tf + [(BRH + 2 + nf) + (BRL + 2 + nf)] / IICϕ}

Symbols in the expressions

SCLE: RIICnFER.SCLE bit

BRH: RIICnBRH.BRH[4:0] bits

BRL: RIICnBRL.BRL[4:0] bits

CKS: RIICnMR1.CKS bits

NFE: RIICnFER.NFE bit

IICϕ: Internal reference clock selected by the CKS bits

tf: SCL signal falling time [s] *1

tr: SCL signal rising time [s] *1

nf: Number of digital noise filter stages specified in the RIICnMR3.NF[0.1] bits

Note 1. The rising time (tr) and falling time (tf) of the SCL signal depend on the total capacitance of the bus line (Cb) and pull-up resistor (Rp). For details, see I²C Bus Standard from NXP Semiconductors.

Table 18.20 Minimum Specifiable Value for RIICnBRL and RIICnBRH

SCLE	NFE	nf	Minimum Pulse Width that Passes through Digital Filter	Minimum Specifiable Value for BRH and BRL	Pulse Width when Minimum Value is Specified
0	0	—	1 × IICϕ	1	2 × IICϕ
0	1	1	2 × IICϕ	2	3 × IICϕ
0	1	2	3 × IICϕ	3	4 × IICϕ
0	1	3	4 × IICϕ	4	5 × IICϕ
0	1	4	5 × IICϕ	5	6 × IICϕ
IICϕ cycle > P0ϕ cycle (CKS ≠ 000)					
1	0	—	1 × IICϕ	0	2 × IICϕ
1	1	1	2 × IICϕ	1	4 × IICϕ
1	1	2	3 × IICϕ	2	6 × IICϕ

chanhvinh.com

Table 18.20 Minimum Specifiable Value for RIICnBRL and RIICnBRH

SCLE	NFE	nf	Minimum Pulse Width that Passes through Digital Filter	Minimum Specifiable Value for BRH and BRL	Pulse Width when Minimum Value is Specified
1	1	3	4 × IICϕ	3	8 × IICϕ
1	1	4	5 × IICϕ	4	10 × IICϕ
IICϕ cycle = P0ϕ cycle (CKS = 000)					
1	0	—	2 × P0ϕ	0	3 × IICϕ
1	1	1	3 × P0ϕ	1	5 × IICϕ
1	1	2	4 × P0ϕ	2	7 × IICϕ
1	1	3	5 × P0ϕ	3	9 × IICϕ
1	1	4	6 × P0ϕ	4	11 × IICϕ

Table 18.21 and Table 18.22 list examples of RIICnBRH/RIICnBRL settings.

Table 18.21 Examples of RIICnBRH/RIICnBRL Settings for Transfer Rate (when RIICnFER.SCLE = 1 and RIICnFER.NFE = 0)

Transfer Rate [kbps]	Peripheral Clock Operating Frequency P0ϕ [MHz]								
	25			30			33		
	RIICnMR1. CKS[2:0]	RIICnBRH. BRH	RIICnBRL. BRL	RIICnMR1. CKS[2:0]	RIICnBRH. BRH	RIICnBRL. BRL	RIICnMR1. CKS[2:0]	RIICnBRH. BRH	RIICnBRL. BRL
10	110_B	21 ($F5_H$)	14 (EE_H)	110_B	25 ($F9_H$)	18 ($F2_H$)	111_B	12 (EC_H)	10 (EA_H)
50	100_B	20 ($F4_H$)	7 ($E7_H$)	100_B	20 ($F4_H$)	13 (ED_H)	100_B	22 ($F6_H$)	15 (EF_H)
100	010_B	30 (FE_H)	29 (FD_H)	011_B	18 ($F2_H$)	16 ($F0_H$)	011_B	20 ($F4_H$)	18 ($F2_H$)
400	000_B	27 (FB_H)	30 (FE_H)	001_B	16 ($F0_H$)	18 ($F2_H$)	001_B	18 ($F2_H$)	20 ($F4_H$)

Table 18.22 Examples of RIICnBRH/RIICnBRL Settings for Transfer Rate (when RIICnFER.SCLE = 1, RIICnFER.NFE= 1, and Number of NF Stages = 4)

Transfer Rate [kbps]	Peripheral Clock Operating Frequency P0ϕ [MHz]								
	25			30			33		
	RIICnMR1. CKS[2:0]	RIICnBRH. BRH	RIICnBRL. BRL	RIICnMR1. CKS[2:0]	RIICnBRH. BRH	RIICnBRL. BRL	RIICnMR1. CKS[2:0]	RIICnBRH. BRH	RIICnBRL. BRL
10	110_B	19 ($F3_H$)	8 ($E8_H$)	110_B	24 ($F8_H$)	11 (EB_H)	110_B	28 (FC_H)	12 (EC_H)
50	011_B	28 (FC_H)	22 ($F6_H$)	100_B	19 ($F3_H$)	7 ($E7_H$)	100_B	20 ($F4_H$)	10 (EA_H)
100	010_B	26 (FA_H)	25 ($F9_H$)	011_B	14 (EE_H)	12 (EC_H)	011_B	16 ($F0_H$)	14 (EE_H)
400	000_B	23 ($F7_H$)	26 (FA_H)	001_B	11 (EB_H)	15 (EF_H)	001_B	14 (EE_H)	16 ($F0_H$)

chanhvinh.com

18.3.14 RIICnDRT — I²C Bus Transmit Data Register

Access: RIICnDRT is a 32-bit readable/writable register.
RIICnDRTL and RIICnDRTH are 16-bit readable/writable registers.
RIICnDRTLL, RIICnDRTLH, RIICnDRTHL, and RIICnDRTHH are 8-bit readable/writable registers.

Address: RIICnDRT: <RIICn_base> + $003C_H$
RIICnDRTL: <RIICn_base> + $003C_H$, RIICnDRTH: <RIICn_base> + $003E_H$
RIICnDRTLL: <RIICn_base> + $003C_H$, RIICnDRTLH: <RIICn_base> + $003D_H$, RIICnDRTHL: <RIICn_base> + $003E_H$,
RIICnDRTHH: <RIICn_base> + $003F_H$

Initial Value: 0000 $00FF_H$ This register is initialized by any reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DRT[7:0]							
Initial value	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When RIICnDRT detects a space in the I^2C bus shift register (RIICnDRS), it transfers the transmit data that has been written to RIICnDRT to RIICnDRS and starts transmitting data in transmit mode.

The double-buffer structure of RIICnDRT and RIICnDRS allows continuous transmit operation if the next transmit data has been written to RIICnDRT while the RIICnDRS data is being transmitted.

RIICnDRT can always be read and written. Write transmit data to RIICnDRT once when a transmit data empty interrupt (INTRIICTI) request is generated. When writing to bits 8 to 15, be sure to write 0 to these bits.

chanhvinh.com

18.3.15 RIICnDRR — I²C Bus Receive Data Register

Access: RIICnDRR is a 32-bit readable/writable register.
RIICnDRRL and RIICnDRRH are 16-bit readable/writable registers.
RIICnDRRLL, RIICnDRRLH, RIICnDRRHL, and RIICnDRRHH are 8-bit readable/writable registers.

Address: RIICnDRR: <RIICn_base> + 0040_H
RIICnDRRL: <RIICn_base> + 0040_H, RIICnDRRH: <RIICn_base> + 0042_H
RIICnDRRLL: <RIICn_base> + 0040_H, RIICnDRRLH: <RIICn_base> + 0041_H, RIICnDRRHL: <RIICn_base> + 0042_H,
RIICnDRRHH: <RIICn_base> + 0043_H

Initial Value: $0000\ 0000_H$ This register is initialized by any reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DRR[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

When 1 byte of data has been received, the received data is transferred from the I²C bus shift register (RIICnDRS) to RIICnDRR to enable the next data to be received.

The double-buffer structure of RIICnDRS and RIICnDRR allows continuous receive operation if the received data has been read from RIICnDRR while RIICnDRS is receiving data.

RIICnDRR cannot be written. Read data from RIICnDRR once when a receive data full interrupt (INTRIICRI) request is generated.

If DRR receives the next receive data before the current data is read from RIICnDRR (while the RIICnSR2.RDRF flag is 1), the RIIC automatically holds the SCL clock low one cycle before the RDRF flag is set to 1 next.

chanhvinh.com

18.3.16 RIICnDRS — I²C Bus Shift Register

Access: This register is not accessible.

Address: —

Initial Value: 0000 00FF$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DRS[7:0]							
Initial value	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

RIICnDRS is an 8-bit shift register to transmit and receive data.

During transmission, transmit data is transferred from RIICnDRT to RIICnDRS and is sent from the SDA pin. During reception, data is transferred from RIICnDRS to RIICnDRR after 1 byte of data has been received.

RIICnDRS cannot be accessed directly.

chanhvinh.com

18.4 Interrupt Sources

The RIIC issues eight types of interrupt requests: transmission complete, receive data full, transmit data empty, detection of a stop condition, detection of a start condition, reception of a NACK, arbitration lost, and timeout.

Table 18.23 lists details of the interrupt requests. The receive data full and transmit data empty interrupt sources are both capable of launching data transfer by starting the DMAC.

Table 18.23 Interrupt Sources

Symbol	Interrupt Source	Interrupt Flag	DMAC Launching	Priority*1	Interrupt Condition
INTRIICTEI	Transmission complete	TEND	Not possible	High	TEND = 1 • TEIE = 1
INTRIICRI	Receive data full	RDRF	Possible		RDRF = 1 • RIE = 1
INTRIICTI	Transmit data empty	TDRE	Possible		TDRE = 1 • TIE = 1
INTRIICSPI	Detection of a stop condition	STOP	Not possible		STOP = 1 • SPIE = 1
INTRIICSTI	Detection of a start condition	START	Not possible		START = 1 • STIE = 1
INTRIICNAKI	Reception of a NACK	NACKF	Not possible		NACKF = 1 • NAKIE = 1
INTRIICALI	Arbitration lost	AL	Not possible		AL = 1 • ALIE = 1
INTRIICTMOI	Timeout	TMOF	Not possible	Low	TMOF = 1 • TMOIE = 1

Note 1. When the setting value of the interrupt priority register (ICDIPRn) is the same

Clear or mask the each flag during interrupt handling.

CAUTIONS

1. There is a latency (delay) between the execution of a write instruction for a peripheral module by the CPU and actual writing to the module. Thus, when an interrupt flag has been cleared or masked, read the relevant flag again to check whether clearing or masking has been completed, and then return from interrupt processing. Returning from interrupt processing without checking that writing to the module has been completed creates a possibility of repeated processing of the same interrupt.
2. The INTRIICRI and INTRIICTI interrupts do not need to be cleared because they are edge interrupts.
3. When using the INTRIICTEI interrupt, clear the RIICnSR2.TEND flag in the INTRIICTEI interrupt processing.
4. When using the INTRIICSPI interrupt, clear the RIICnSR2.STOP flag in the INTRIICSPI interrupt processing.
5. When using the INTRIICSTI interrupt, clear the RIICnSR2.START flag in the INTRIICSTI interrupt processing.
6. When using the INTRIICNAKI interrupt, clear the RIICnSR2.NACKF flag in the INTRIICNAKI interrupt processing.
7. When using the INTRIICALI interrupt, clear the RIICnSR2.AL flag in the INTRIICALI interrupt processing.
8. When using the INTRIICTMOI interrupt, clear the RIICnSR2.TMOF flag in the INTRIICTMOI interrupt processing.

chanhvinh.com

18.5 Operation

18.5.1 Communication Data Format

The I^2C bus format consists of 8-bit data and 1-bit acknowledge (one frame). After a start condition or restart condition is issued, the master device sends the slave address and data direction in the first frame. The specified slave is valid until a stop condition is issued or a new slave is specified by a restart condition.

Figure 18.3 shows the I^2C bus format, and Figure 18.4 shows the I^2C bus timing.

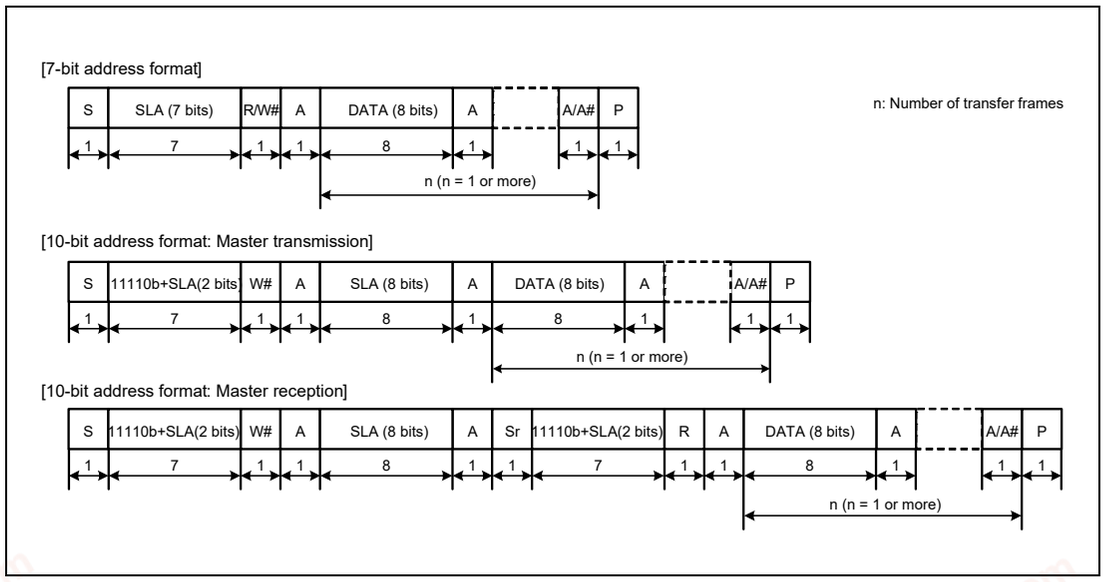


Figure 18.3 I^2C Bus Format

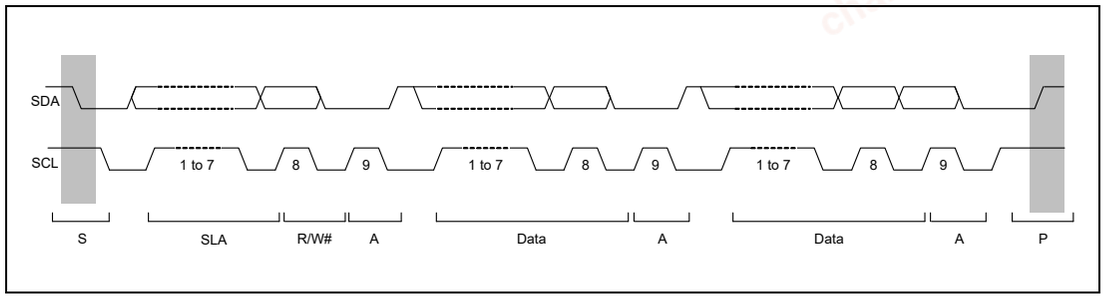


Figure 18.4 I^2C Bus Timing (SLA = 7 Bits)

S: Start condition. The master device drives the SDA line low from high level while the SCL line is at a high level.

SLA: Slave address, by which the master device selects a slave device.

R/W#: Indicates the direction of data transfer: from the slave device to the master device when R/W is 1, or from the master device to the slave device when R/W is 0.

A: Acknowledge. The receive device drives the SDA line low. (In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.)

A#: Not-acknowledge. The receiving device has not returned a response or is not present so the SDA line has remained at the high level.

chanhvinh.com

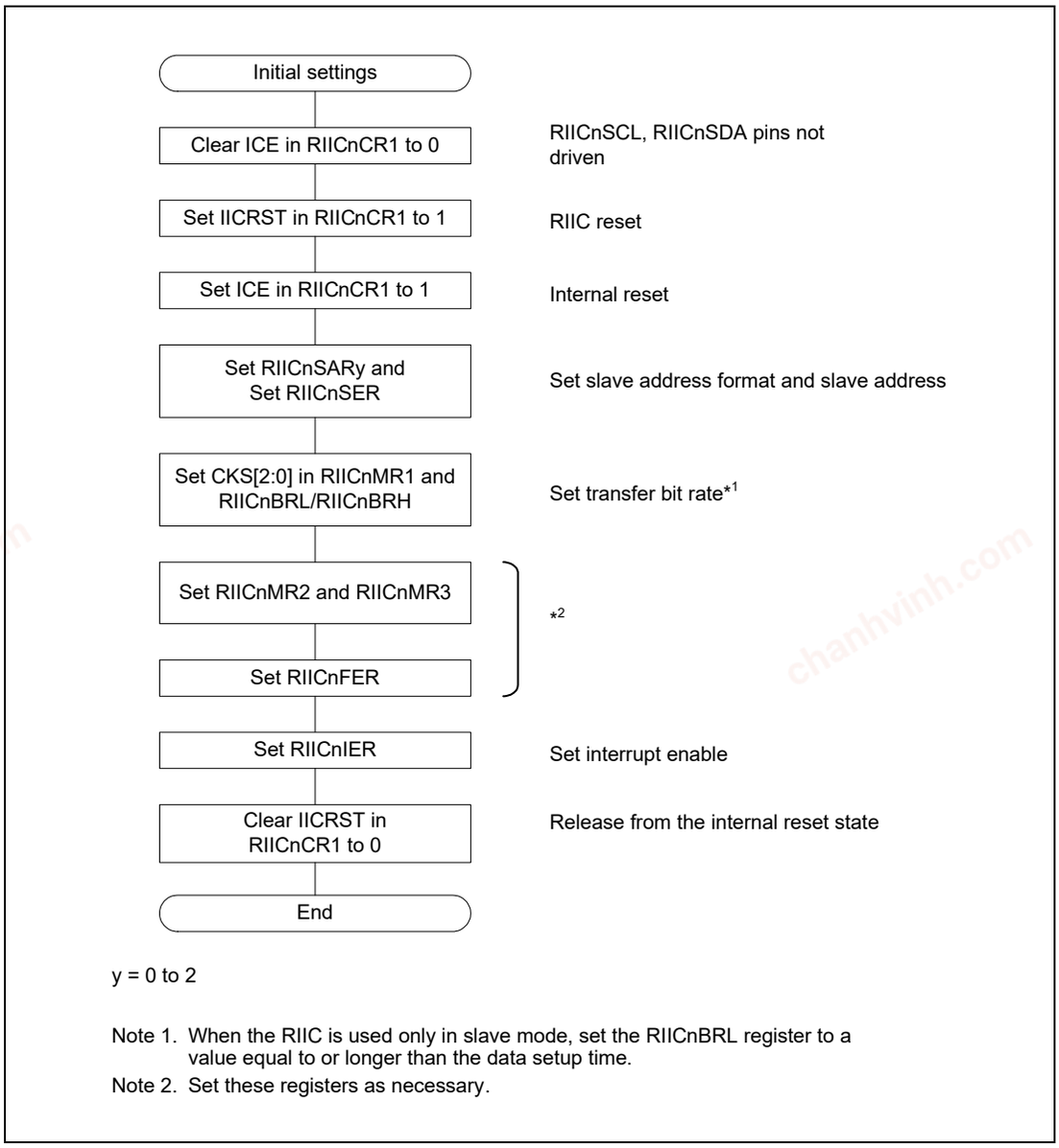
Sr: Restart condition. The master device drives the SDA line low from the high level after the setup time has elapsed with the SCL line at the high level.

DATA: Transmitted or received data

P: Stop condition. The master device drives the SDA line high from low level while the SCL line is at a high level.

18.5.2 Initial Settings

Before starting data transmission and reception, initialize the RIIC according to the procedure in Figure 18.5.



Note 1. When the RIIC is used only in slave mode, set the RIICnBRL register to a value equal to or longer than the data setup time.

Note 2. Set these registers as necessary.

Figure 18.5 Example of RIIC Initialization Flowchart

chanhvinh.com

18.5.3 Master Transmit Operation

In master transmit operation, the RIIC outputs the SCL (clock) and transmitted data signals as the master device, and the slave device returns acknowledgements. Figure 18.6 shows an example of usage of master transmission and Figure 18.7 to Figure 18.9 show the timing of operations in master transmission.

The following describes the procedure and operations for master transmission.

(1) Set the RIICnCR1.IICRST bit 1 to 1 (RIIC reset) and then set the RIICnCR1.ICE bit to 1 (internal reset) with the RIICnCR1.ICE bit cleared to 0 (RIICnSCL and RIICnSDA pins not driven). This initializes the internal state and the various flags of RIICnSR1. After that, set registers RIICnSARy, RIICnSER, RIICnMR1, RIICnBRH, and RIICnBRL (y = 0 to 2), and set the other registers as necessary (for initial settings of the RIIC, see Figure 18.5). When the necessary register settings have been completed, set the RIICnCR1.IICRST bit to 0 (for release from the reset state). This step is not necessary if initialization of the RIIC has already been completed.

(2) Read the RIICnCR2.BBSY flag to check that the bus is open, and then set the RIICnCR2.ST bit to 1 (start condition issuance request). Upon receiving the request, the RIIC issues a start condition. At the same time, the BBSY flag and the RIICnSR2.START flag are automatically set to 1 and the ST bit is automatically cleared to 0. At this time, if the start condition is detected and the internal levels for the SDA output state and the levels on the SDA line have matched while the ST bit is 1, the RIIC recognizes that issuing of the start condition as requested by the ST bit has been successfully completed, and the RIICnCR2.MST and TRS bits are automatically set to 1, placing the RIIC in master transmit mode. The RIICnSR2.TDRE flag is also automatically set to 1 in response to setting of the TRS bit to 1.

(3) Check that the RIICnSR2.TDRE flag is 1, and then write the value for transmission (the slave address and the R/W# bit) to RIICnDRT. Once the data for transmission are written to RIICnDRT, the TDRE flag is automatically cleared to 0, the data are transferred from RIICnDRT to RIICnDRS, and the TDRE flag is again set to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRS bit is automatically updated to select master transmit or master receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the RIIC continues in master transmit mode.
Since the RIICnSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the RIICnCR2.SP bit to issue a stop condition.
For data transmission with an address in the 10-bit format, start by writing 1111 0_B, the two higher-order bits of the slave address, and W# to RIICnDRT as the first address transmission. Then, as the second address transmission, write the eight lower-order bits of the slave address to RIICnDRT.

(4) After confirming that the RIICnSR2.TDRE flag is 1, write the data for transmission to the RIICnDRT register. The RIIC automatically holds the SCL line low until the data for transmission are ready or a stop condition is issued.

(5) After the last byte of the data to be transmitted is written to RIICnDRT register, wait until the value of the RIICnSR2.TEND flag returns to 1, and then set the RIICnCR2.SP bit to 1 (stop condition issuance request). Upon receiving a stop condition issuance request, the RIIC issues the stop condition.

chanhvinh.com

(6) Upon detecting the stop condition, the RIIC automatically clears the RIICnCR2.MST and TRS bits to 00_B and enters slave receive mode. Furthermore, it automatically clears the RIICnSR2.TDRE and TEND flags to 0, and sets the RIICnSR2.STOP flag in to 1.

(7) Clear the RIICnSR2.NACKF and STOP flags to 0.

CAUTION

Operations for transfer start if the RIICnSR2.NACKF flag is cleared to 0 before RIICnSR2.STOP is set to 1. Be sure to confirm that RIICnSR2.STOP is set to 1 before clearing RIICnSR2.NACKF to 0. In particular, when the NACK receive interrupt (INTRIICNAKI) is in use, take care not to clear the NACKF flag to 0 before the STOP flag is set to 1 during interrupt processing.

chanhvinh.com

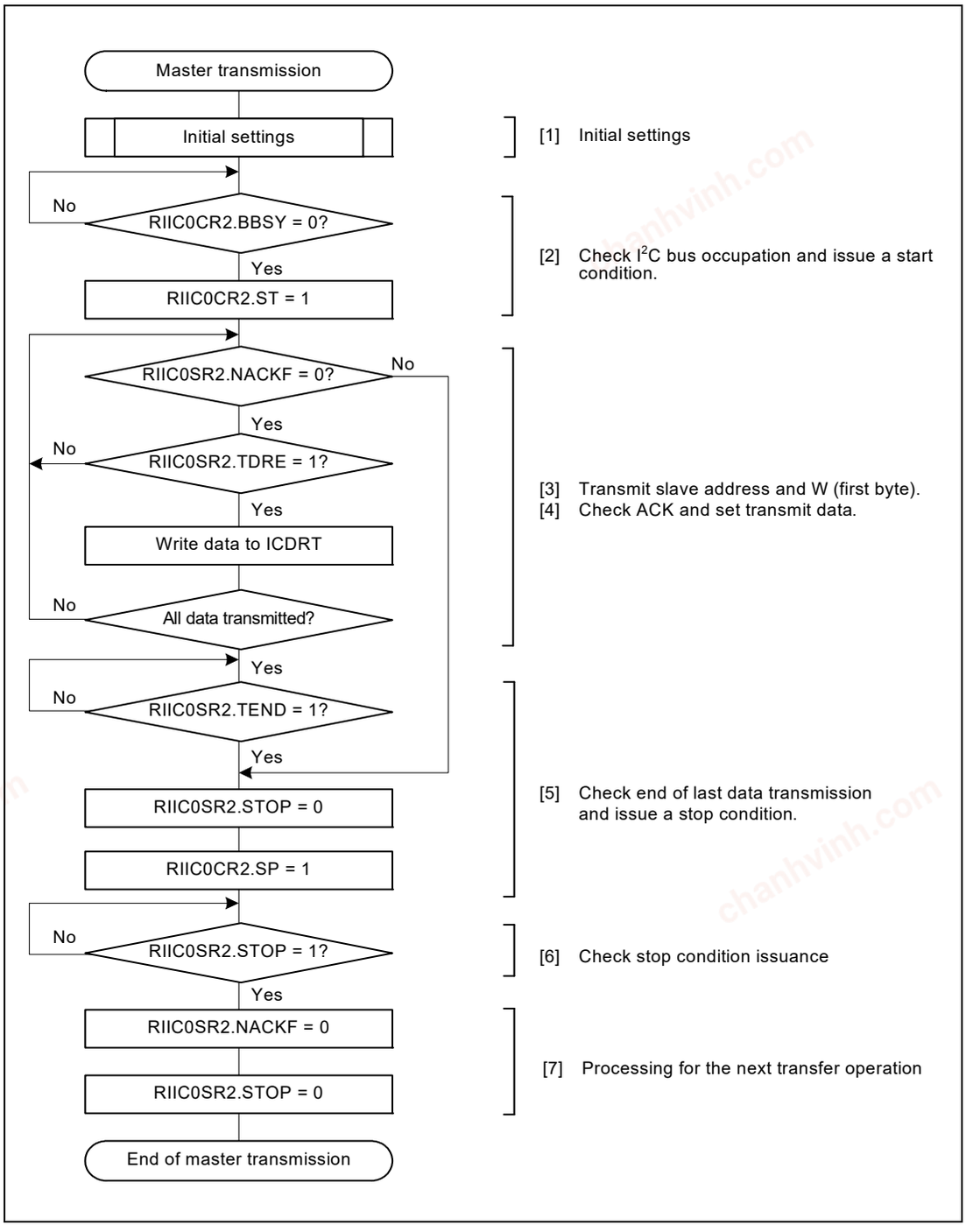


Figure 18.6 Example of Master Transmission Flowchart

chanhvinh.com

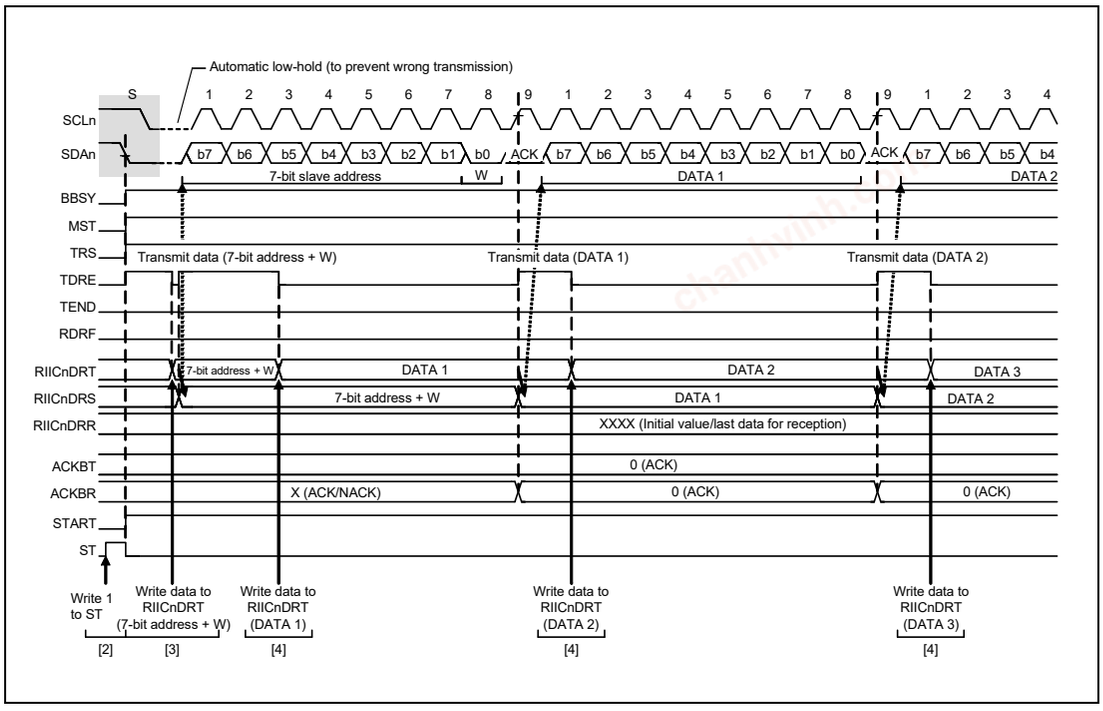


Figure 18.7 Master Transmit Operation Timing (1) (7-Bit Address Format)

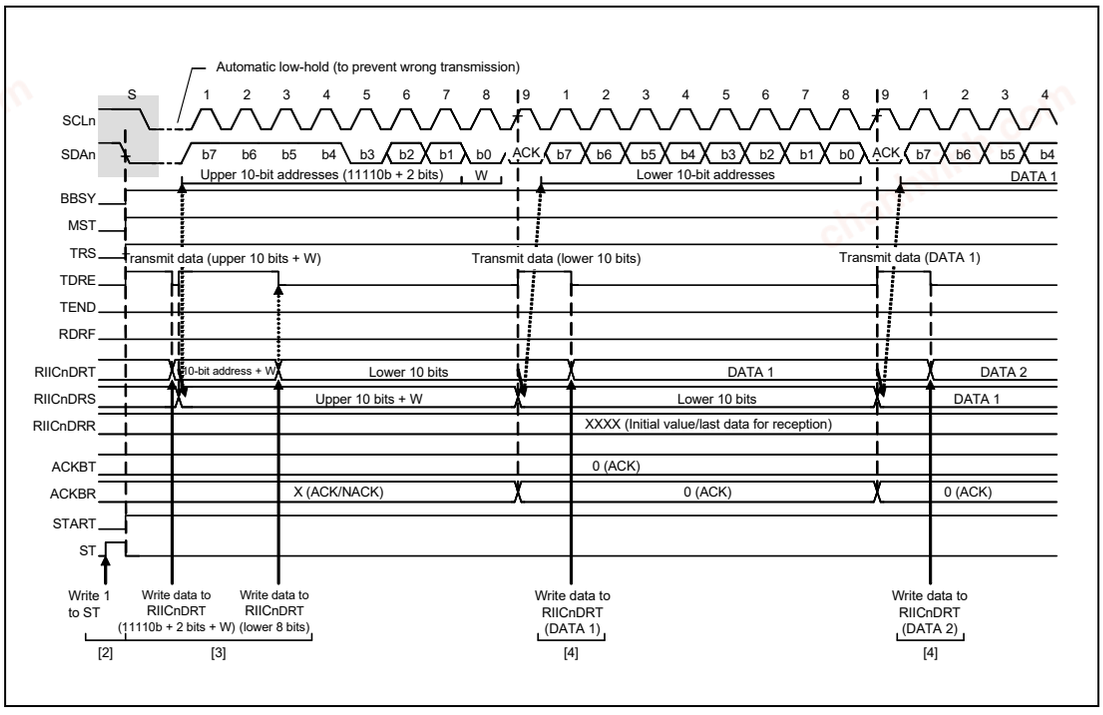


Figure 18.8 Master Transmit Operation Timing (2) (10-Bit Address Format)

chanhvinh.com

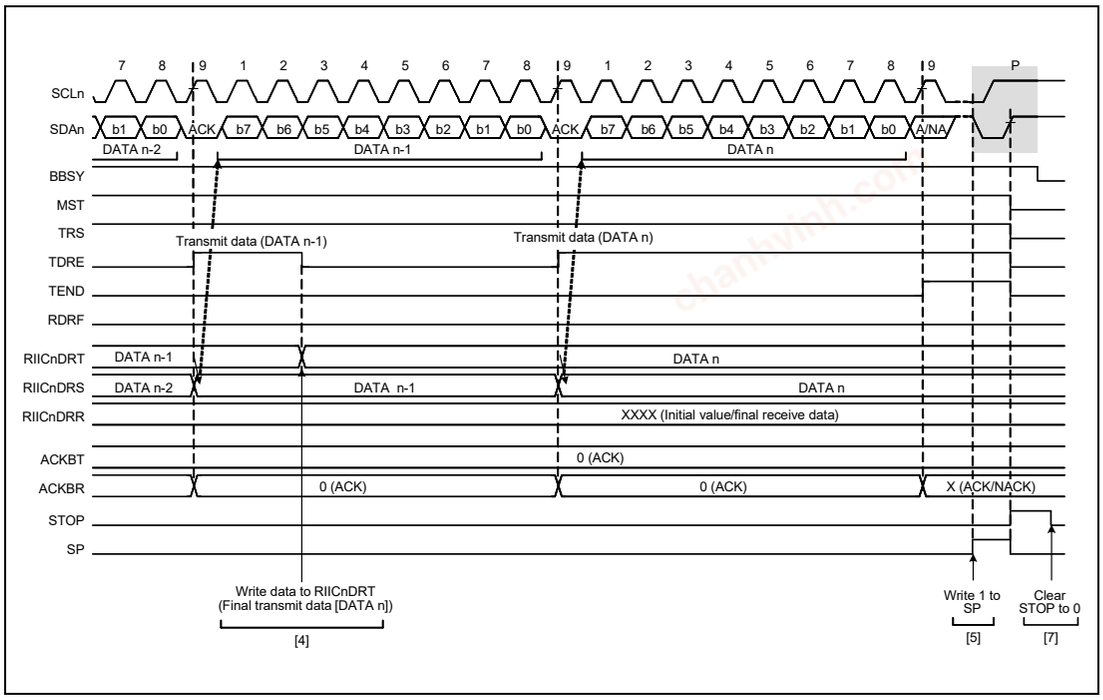


Figure 18.9 Master Transmit Operation Timing (3)

chanhvinh.com

18.5.4 Master Receive Operation

In master receive operation, the RIIC as a master device outputs the SCL (clock) signal, receives data from the slave device, and returns acknowledgements. Since the RIIC must start by sending a slave address to the corresponding slave device, this part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.

Figure 18.10 shows an example of usage for the master reception of 3 or more bytes (7-bit address format), Figure 18.14 shows an example of usage for the master reception of 1 or 2 bytes (7-bit address format), and Figure 18.11 to Figure 18.13 show the timing of operations in master reception.

The following describes the procedure and operations for master reception.

(1) Set the RIICnCR1.IICRST bit to 1 (RIIC reset) and then set the RIICnCR1.ICE bit to 1 (internal reset) with the RIICnCR1.ICE bit cleared to 0 (RIICnSCL and RIICnSDA pins not driven). This initializes the internal state and the various flags of RIICnSR1. After that, set registers RIICnSARy, RIICnSER, RIICnMR1, RIICnBRH, and RIICnBRL (y = 0 to 2), and set the other registers as necessary (for initial settings of the RIIC, see Figure 18.5). When the necessary register settings have been completed, set the RIICnCR1.IICRST bit to 0 (for release from the reset state). This step is not necessary if initialization of the RIIC has already been completed.

(2) Read the RIICnCR2.BBSY flag to check that the bus is open, and then set the RIICnCR2.ST bit to 1 (start condition issuance request). Upon receiving the request, the RIIC issues a start condition. When the RIIC detects the start condition, the BBSY flag and the RIICnSR2.START flag are automatically set to 1 and the ST bit is automatically cleared to 0. At this time, if the start condition is detected and the levels for the SDA output and the levels on the SDA line have matched while the ST bit is 1, the RIIC recognizes that issuing of the start condition as requested by the ST bit has been successfully completed, and the RIICnCR2.MST and TRS bits are automatically set to 1, placing the RIIC in master transmit mode. The RIICnSR2.TDRE flag is also automatically set to 1 in response to setting of the TRS bit to 1.

(3) Check that the RIICnSR2.TDRE flag is 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to RIICnDRT. Once the data for transmission are written to RIICnDRT, the TDRE flag is automatically cleared to 0, the data are transferred from RIICnDRT to RIICnDRS, and the TDRE flag is again set to 1. Once the byte containing the slave address and R/W# bit has been transmitted, the value of the RIICnCR2.TRS bit is automatically updated to select transmit or receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 1, the RIICnCR2.TRS bit is cleared to 0 on the rising edge of the ninth cycle of SCL (the clock signal), placing the RIIC in master receive mode. At this time, the TDRE flag is automatically cleared to 0 and the RIICnSR2.RDRF flag is automatically set to 1.
Since the RIICnSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the RIICnCR2.SP bit to issue a stop condition.
For master reception from a device with a 10-bit address, start by using master transmission to issue the 10-bit address, and then issue a restart condition. After that, transmitting 1111 0_B, the two higher-order bits of the slave address, and the R bit places the RIIC in master receive mode.

(4) Dummy read RIICnDRR after confirming that the RIICnSR2.RDRF flag is 1; this makes the RIIC start output of the SCL (clock) signal and start data reception.

(5) After 1 byte of data has been received, the RIICnSR2.RDRF flag is set to 1 on the rising edge of the eighth or ninth cycle of SCL clock (the clock signal) as selected by the RIICnMR3.RDRFS bit. Reading out RIICnDRR at this time will produce the received data, and the RDRF flag is

chanhvinh.com

automatically cleared to 0 at the same time. Furthermore, the value of the acknowledgement field received during the ninth cycle of SCL clock is returned as the value set in the RIICnMR3.ACKBT bit. Furthermore, if the next byte to be received is the next to last byte, set the RIICnMR3.WAIT bit to 1 (for wait insertion) before reading the RIICnDRR (containing the second byte from last). As well as enabling NACK output even in the case of delays in processing to set the RIICnMR3.ACKBT bit to 1 (NACK) in step (6), due to other interrupts, etc., this fixes the SCL line to the low level on the rising edge of the ninth clock cycle in reception of the last byte, so the state is such that issuing a stop condition is possible.

(6) When the RIICnMR3.RDRFS bit is 0 and the slave device must be notified that it is to end transfer for data reception after transfer of the next (final) byte, set the RIICnMR3.ACKBT bit to 1 (NACK).

(7) After reading out the byte before last from the RIICnDRR register, if the value of the RIICnSR2.RDRF flag is confirmed to be 1, write 1 to the RIICnCR2.SP bit (stop condition issuance request) and then read the last byte from RIICnDRR. When RIICnDRR is read, the RIIC is released from the wait state and issues the stop condition after low-level output in the ninth clock cycle is completed or the SCL line is released from the low-hold state.

(8) Upon detecting the stop condition, the RIIC automatically clears the RIICnCR2.MST and TRS bits to 00_B and enters slave receive mode. Furthermore, detection of the stop condition leads to setting of the RIICnSR2.STOP flag to 1.

(9) After checking that the RIICnSR2.STOP flag is 1, clear the RIICnSR2.NACKF and STOP flags to 0 for the next transfer operation.

chanhvinh.com

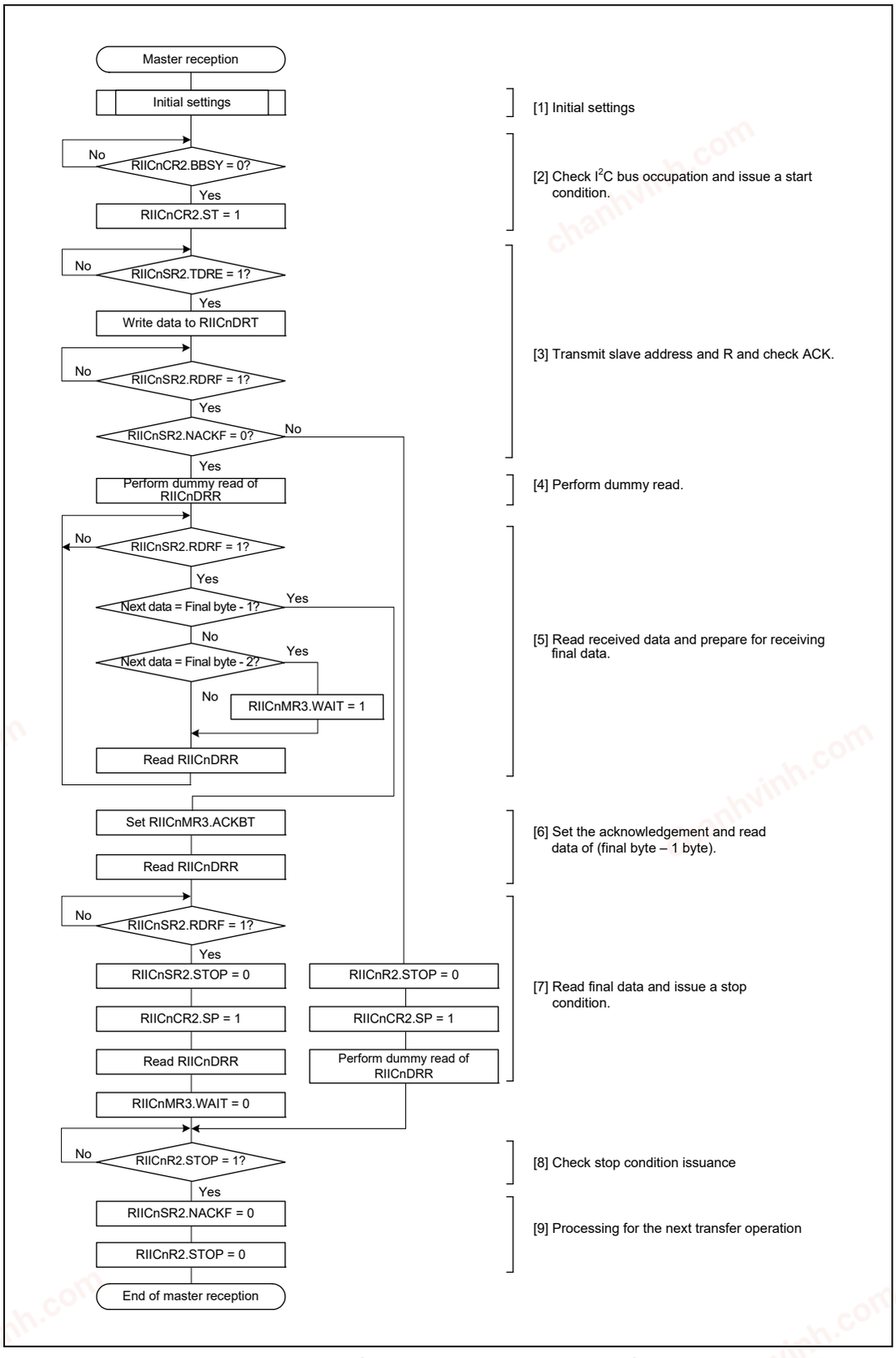


Figure 18.10 Example Flowchart for the Master Reception of 3 or More Bytes (7-Bit Address Format)

chanhvinh.com

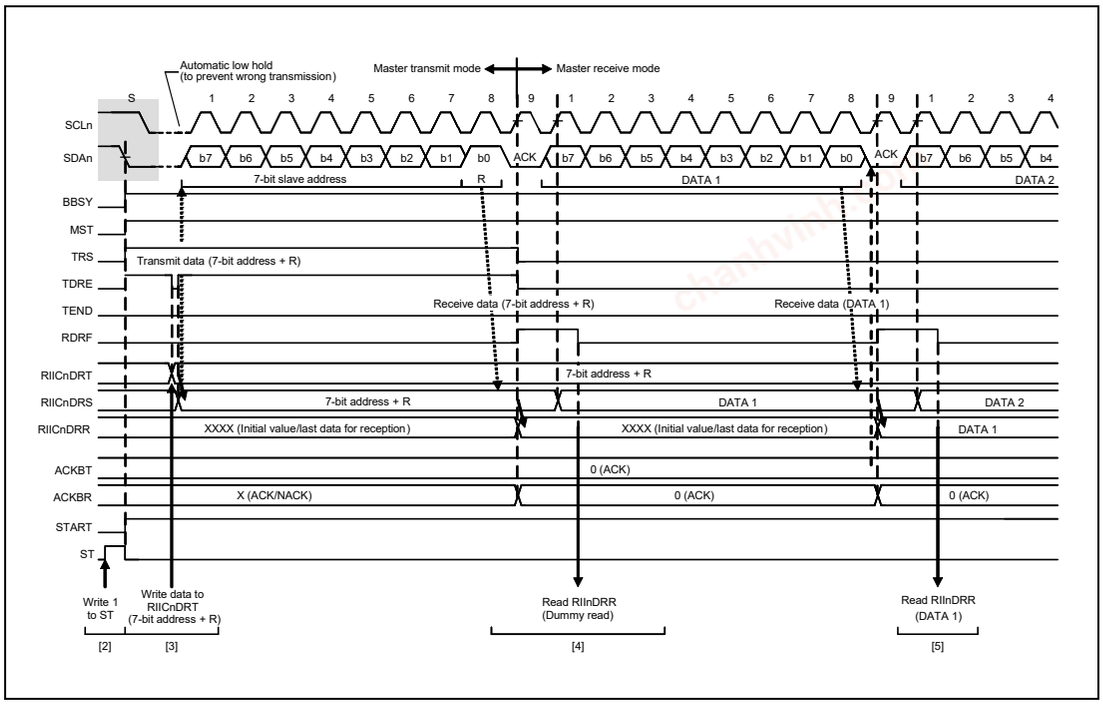


Figure 18.11 Master Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)

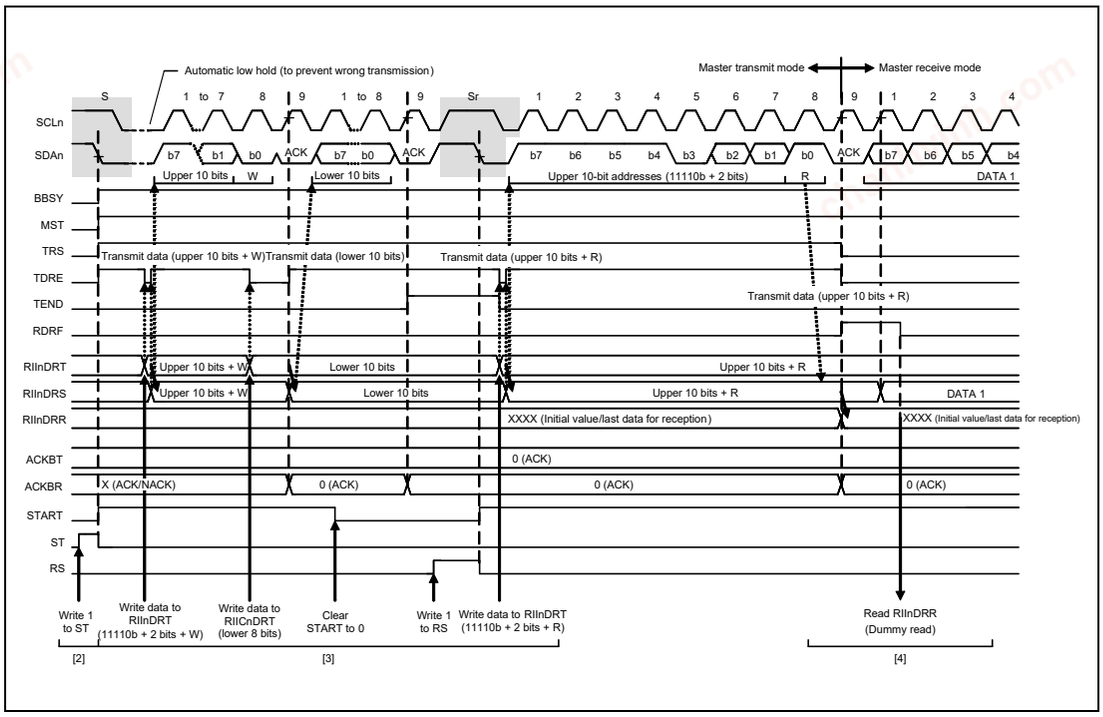


Figure 18.12 Master Receive Operation Timing (2) (10-Bit Address Format, when RDRFS = 0)

chanhvinh.com

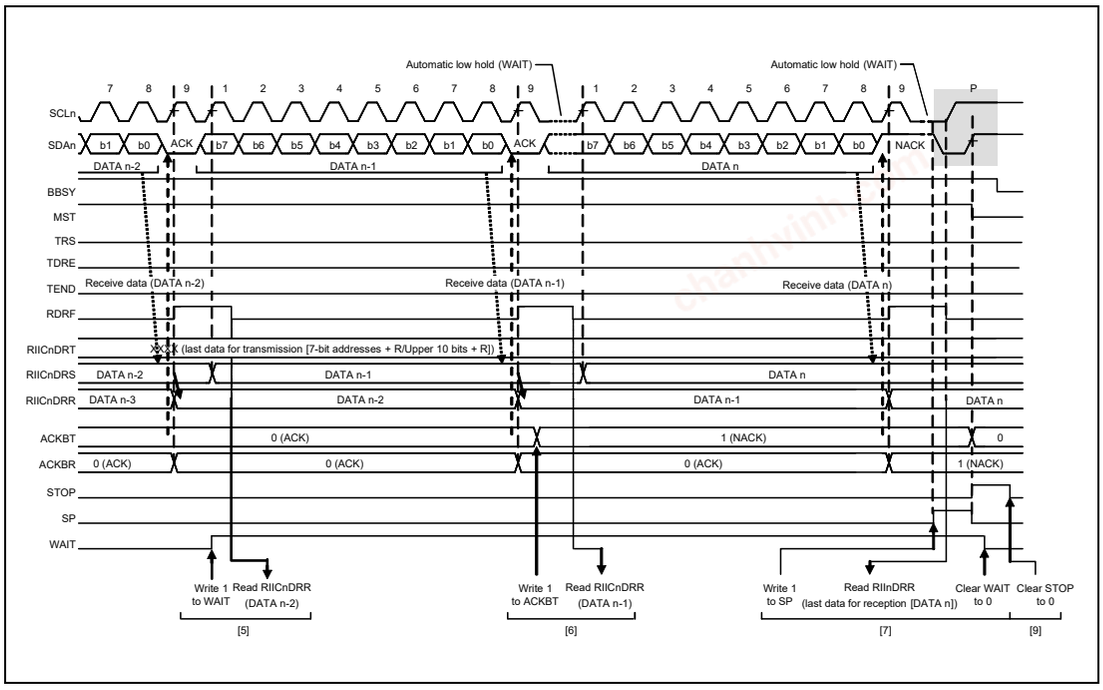


Figure 18.13 Master Receive Operation Timing (3) (when RDRFS = 0)

chanhvinh.com

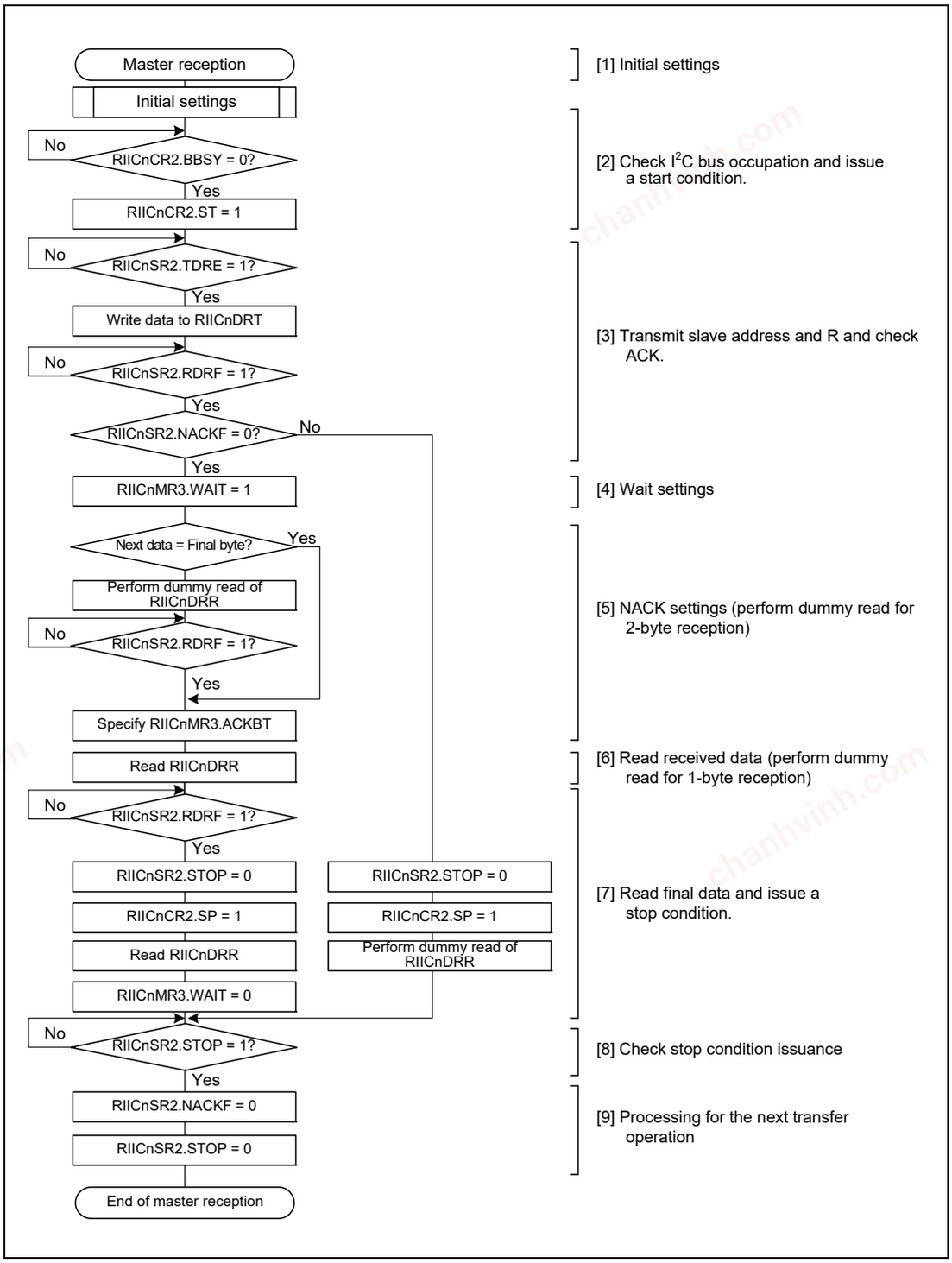


Figure 18.14 Example Flowchart for the Master Reception of 1 or 2 Bytes (7-Bit Address Format)

chanhvinh.com

18.5.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL (clock) signal, the RIIC transmits data as a slave device, and the master device returns acknowledgements.

Figure 18.15 shows an example of usage of slave transmission and Figure 18.16 and Figure 18.17 show the timing of operations in slave transmission.

The following describes the procedure and operations for slave transmission.

(1) Follow the procedure in Figure 18.5 to make initial settings for the RIIC. This step is not necessary if initialization of the RIIC has already been completed. After initial settings, the RIIC will stay in the standby state until it receives a slave address that it matches.

(2) After receiving a matching slave address, the RIIC sets one of the corresponding bits RIICnSR1.HOA, GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the ninth cycle of SCL clock (the clock signal) and outputs the value set in the RIICnMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 1, the RIIC automatically places itself in slave transmit mode by setting both the RIICnCR2.TRS bit and the RIICnSR2.TDRE flag to 1.

(3) After the RIICnSR2.TEND flag is confirmed to be 1, write the data for transmission to the RIICnDRT register. At this time, if the RIIC receives no acknowledge from the master device (receives an NACK signal) while the RIICnFER.NACKE bit is 1, the RIIC suspends transfer of the next data.

(4) Wait unit the RIICnSR2.TEND flag is set to 1 while the RIICnSR2.TDRE flag is 1, after the RIICnSR2.NACKF flag is set to 1 or the last byte for transmission is written to the RIICnDRT register. When the RIICnSR2.NACKF flag or the TEND flag is 1, the RIIC drives the SCL line low on the ninth falling edge of SCL clock.

(5) When the RIICnSR2.NACKF flag or the RIICnSR2.TEND flag is 1, dummy read RIICnDRR to complete the processing. This releases the SCL line.

(6) Upon detecting the stop condition, the RIIC automatically clears bits RIICnSR1.HOA, GCA, and AASy (y = 0 to 2), flags RIICnSR2.TDRE and TEND, and the RIICnCR2.TRS bit to 0, and enters slave receive mode.

(7) Clear the RIICnSR2.NACKF and STOP flags to 0.

CAUTION

Operations for transfer start if the RIICnSR2.NACKF flag is cleared to 0 before RIICnSR2.STOP is set to 1. Be sure to confirm that RIICnSR2.STOP is set to 1 before clearing RIICnSR2.NACKF to 0. In particular, when the NACK receive interrupt (INTRIICNAKI) is in use, take care not to clear the NACKF flag to 0 before the STOP flag is set to 1 during interrupt processing.

chanhvinh.com

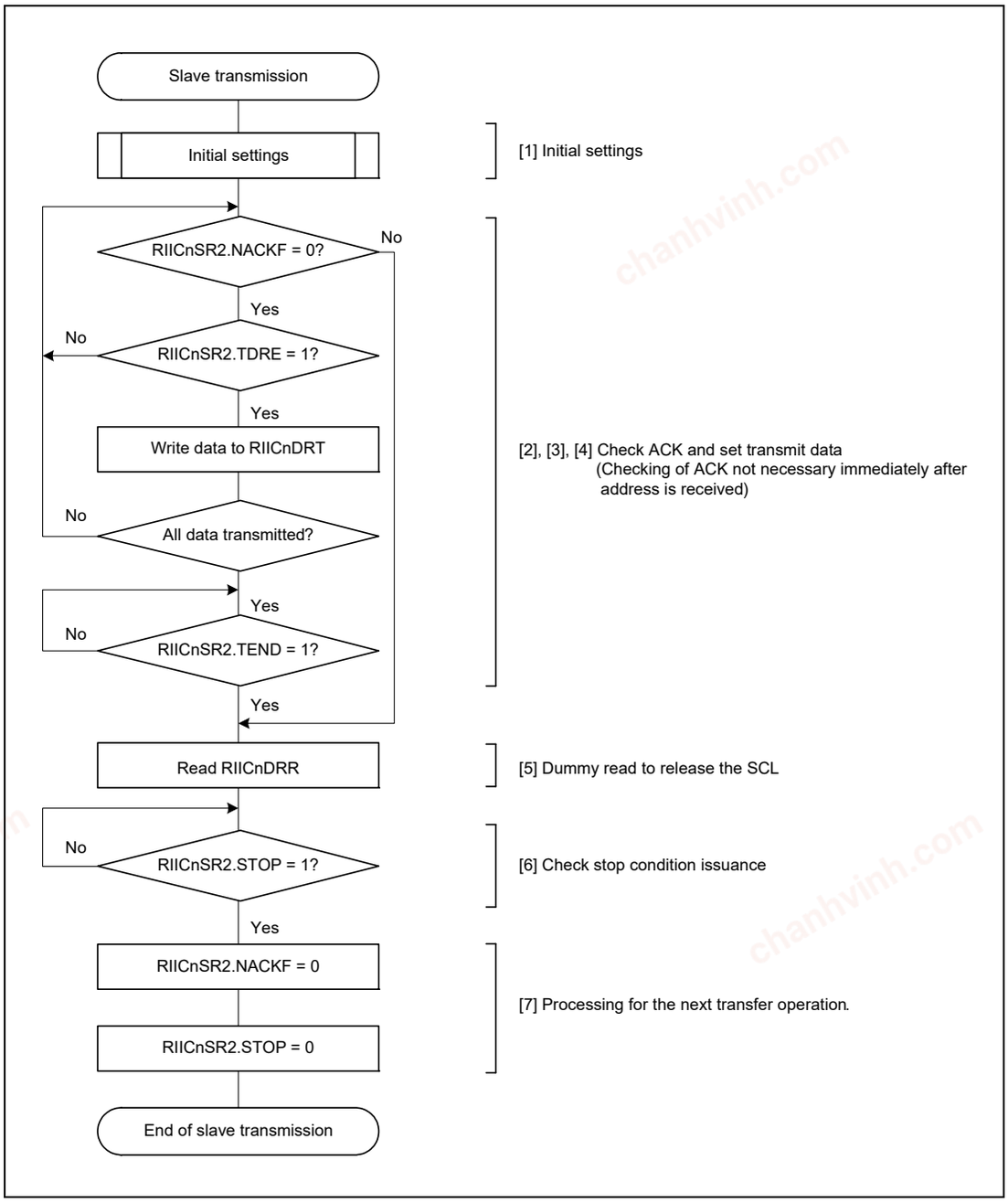


Figure 18.15 Example of Slave Transmission Flowchart

chanhvinh.com

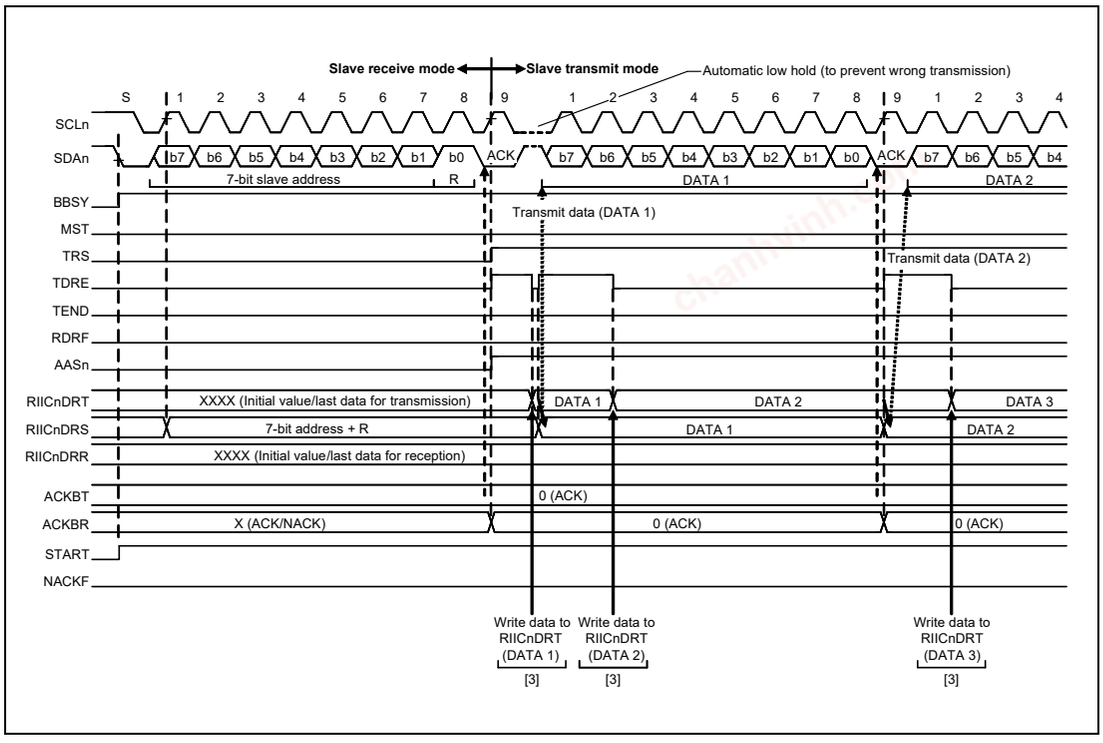


Figure 18.16 Slave Transmit Operation Timing (1) (7-Bit Address Format)

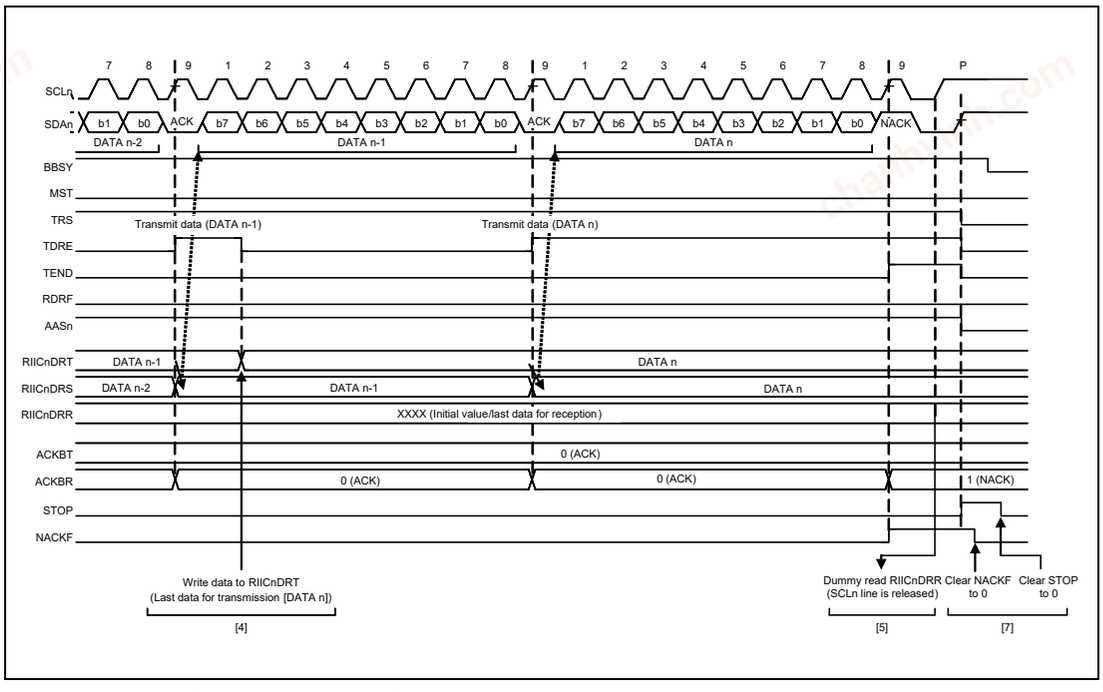


Figure 18.17 Slave Transmit Operation Timing (2)

chanhvinh.com

18.5.6 Slave Receive Operation

In slave receive operation, the master device outputs the SCL clock and transmit data, and the RIIC returns acknowledgements as a slave device.

Figure 18.18 shows an example of usage of slave reception and Figure 18.19 and Figure 18.20 show the timing of operations in slave reception.

The following describes the procedure and operations for slave reception.

(1) Follow the procedure in Figure 18.5 to make initial settings for the RIIC. This step is not necessary if initialization of the RIIC has already been completed. After initial settings, the RIIC will stay in the standby state until it receives a slave address that it matches.

(2) After receiving a matching slave address, the RIIC sets one of the corresponding bits RIICnSR1.HOA, GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the ninth cycle of SCL clock (the clock signal) and outputs the value set in the RIICnMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 0, the RIIC continues to place itself in slave receive mode and sets the RIICnSR2.RDRF flag to 1.

(3) After the RIICnSR2.STOP flag is confirmed to be 0 and the RIICnSR2.RDRF flag to be 1, dummy read RIICnDRR (the dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower eight bits when the 10-bit address format is selected).

(4) When RIICnDRR is read, the RIIC automatically clears the RIICnSR2.RDRF flag to 0. If reading of RIICnDRR is delayed and a next byte is received while the RDRF flag is still set to 1, the RIIC holds the SCL line low from one SCL cycle before the timing with which RDRF should be set. In this case, reading RIICnDRR releases the SCL line from being held at the low level.
When the RIICnSR2.STOP flag is 1 and the RIICnSR2.RDRF flag is also 1, read RIICnDRR until all the data is completely received.

(5) Upon detecting the stop condition, the RIIC automatically clears bits RIICnSR1.HOA, GCA, and AASy (y = 0 to 2) to 0.

(6) After checking that the RIICnSR2.STOP flag is 1, clear the RIICnSR2.STOP flag to 0 for the next transfer operation.

chanhvinh.com

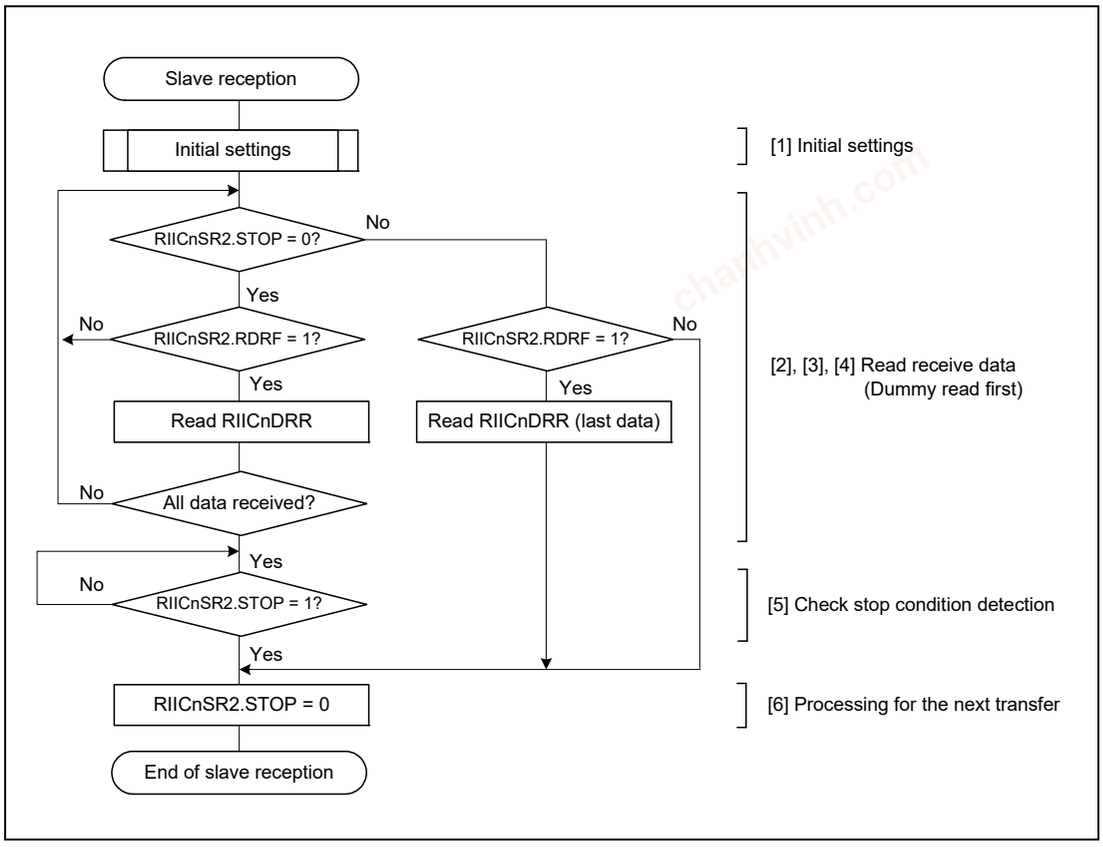


Figure 18.18 Example of Slave Reception Flowchart

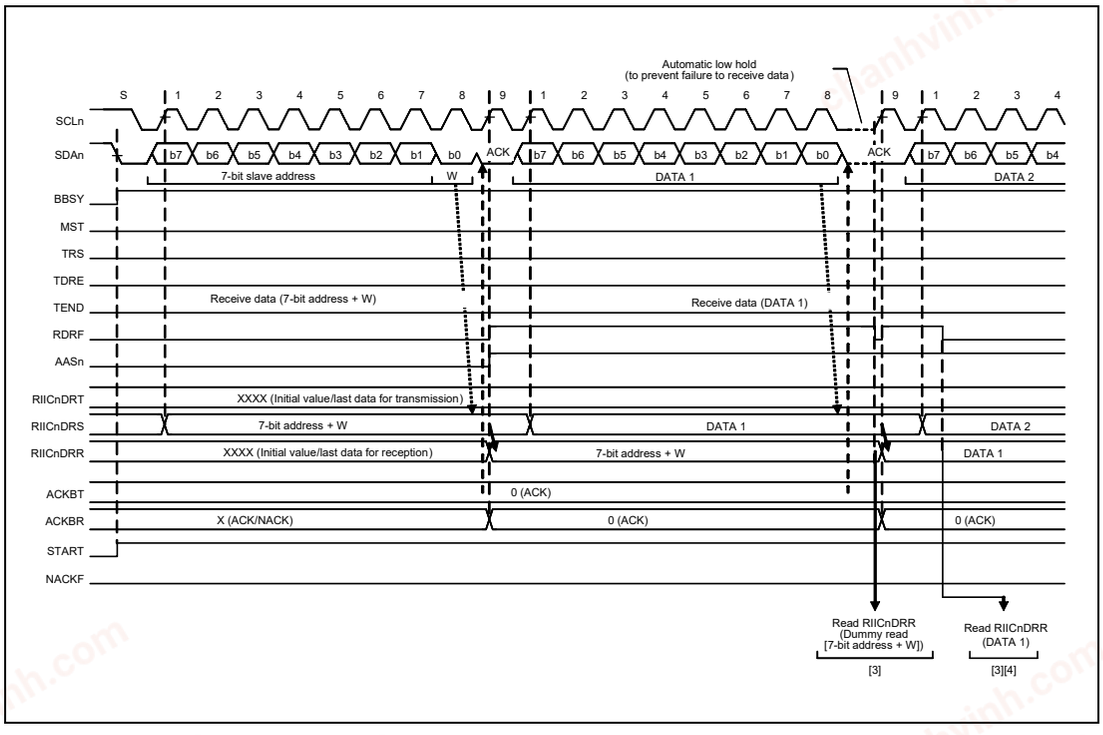


Figure 18.19 Slave Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)

chanhvinh.com

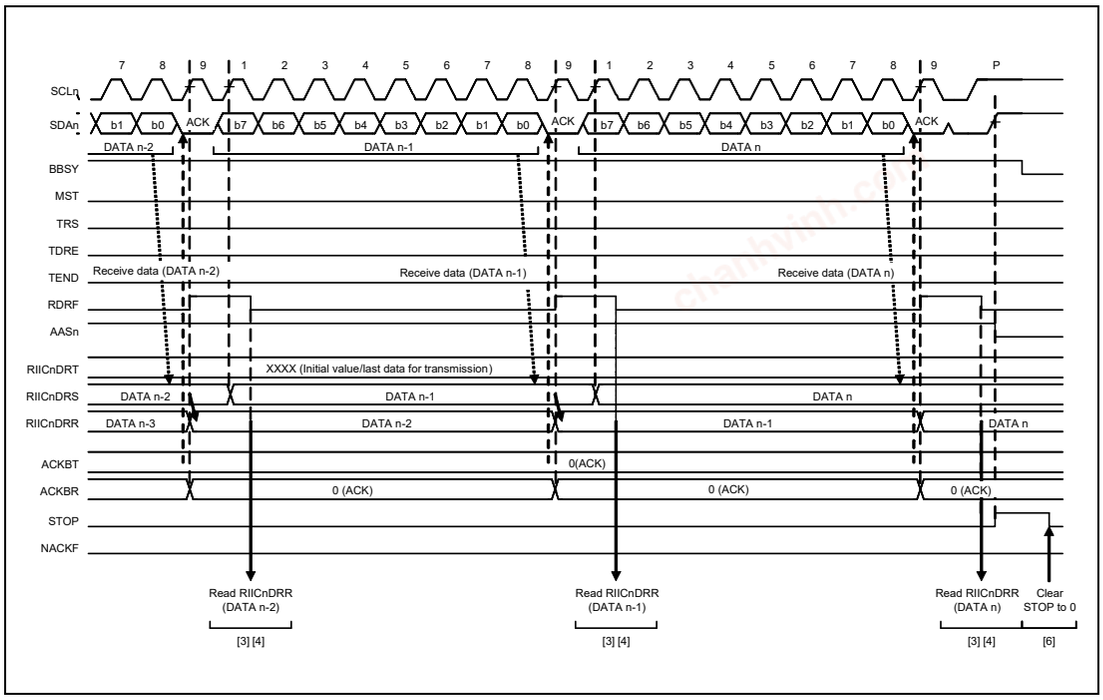


Figure 18.20 Slave Receive Operation Timing (2) (when RDRFS = 0)

chanhvinh.com

18.6 SCL Synchronization Circuit

In generation of the SCL (clock) signal, the RIIC starts counting out the value for width at high level specified in RIICnBRH when it detects a rising edge on the SCL line and drives the SCL line low once counting of the width at high level is complete. When the RIIC detects the falling edge of the SCL line, it starts counting out the width at low level period specified in RIICnBRL, and then stops driving the SCL line (releases the line) once counting of the width at low level is complete. The SCL (clock) signal is thus generated.

If multiple master devices are connected to the I^2C bus, a collision of SCL signals may arise due to contention with another master device. In such cases, the master devices have to synchronize their SCL signals. Since this synchronization of SCL signals must be bit by bit, the RIIC is equipped with a facility (the SCL synchronization circuit) to obtain bit-by-bit synchronization of the SCL clock signals by monitoring the SCL line during communication.

When the RIIC has detected a rising edge on the SCL line and thus started counting out the width at high level specified in RIICnBRH, and the level on the SCL line falls because an SCL signal is being generated by another master device, the RIIC stops counting when it detects the falling edge, drives the level on the SCL line low, and starts counting out the width at low level specified in RIICnBRL. When the RIIC finishes counting out the width at low level, it stops driving the SCL line to the low level (i.e. releases the line). At this time, if the width at low level of the SCL clock signal from the other master device is longer than the width at low level set in the RIIC, the width at low level of the SCL signal will be extended. Once the width at low level for the other master device has ended, the SCL signal rises because the SCL line has been released. When the RIIC finishes outputting the low-level period of the SCL clock, the SCL line is released and the SCL clock rises. That is, in cases of contention of SCL signals from more than one master, the width at high level of the SCL signal is synchronized with that of the clock having the narrower width, and the width at low level of the SCL signal is synchronized with that of the clock having the broader width. However, such synchronization of the SCL signal is only enabled when the RIICnFER.SCLE bit is set to 1.

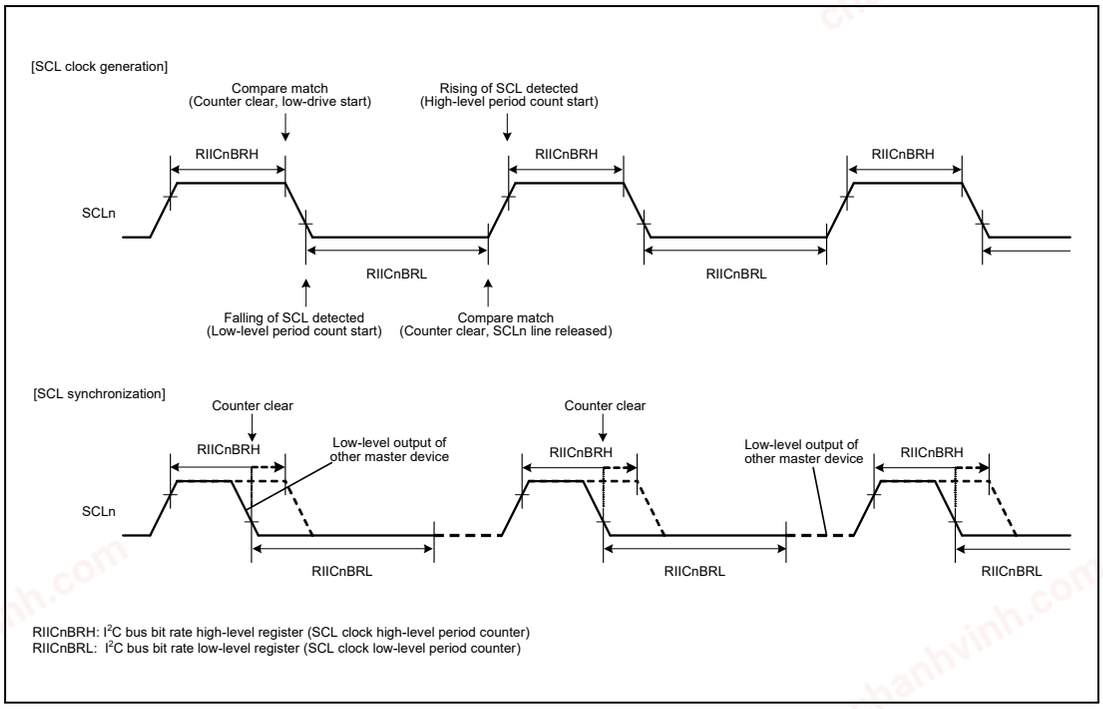


Figure 18.21 Generation and Synchronization of the SCL Signal from the RIIC

chanhvinh.com

18.7 Facility for Delaying SDA Output

The RIIC module incorporates a facility for delaying output on the SDA line. The delay can be applied to all output (issuing of the start, restart, and stop conditions, data, and the ACK and NACK signals) on the SDA line.

With the SDA output delay facility, SDA output is delayed from detection of a falling edge of the SCL signal to ensure that the SDA signal is output within the interval over which the SCL (clock) signal is at the low level. Doing this leads to usage with the aim of preventing erroneous operation of communications devices, with the aim of satisfying the 300-ns (min.) data-hold time requirement of the SMBus specification.

The output delay facility is enabled by setting the RIICnMR2.SDDL[2:0] bits to any value other than 000_B, and disabled by setting the same bits to 000_B.

While the SDA output delay facility is enabled (i.e. while the SDDL[2:0] bits in IMCR2 are set to any value other than 000_B), the RIICnMR2.DLCS bit selects the clock source for counting by the SDA output delay counter as the internal base clock (IICϕ) for the RIIC module or as a clock signal derived by dividing the frequency of the internal base clock by two (IICϕ/2). The counter counts the number of cycles set in the SDDL[2:0] bits in IMCR2. After counting of the set number of cycles of delay is completed, the RIIC module places the required output (start, restart, or stop condition, data, or an ACK or NACK signal) on the SDA line.

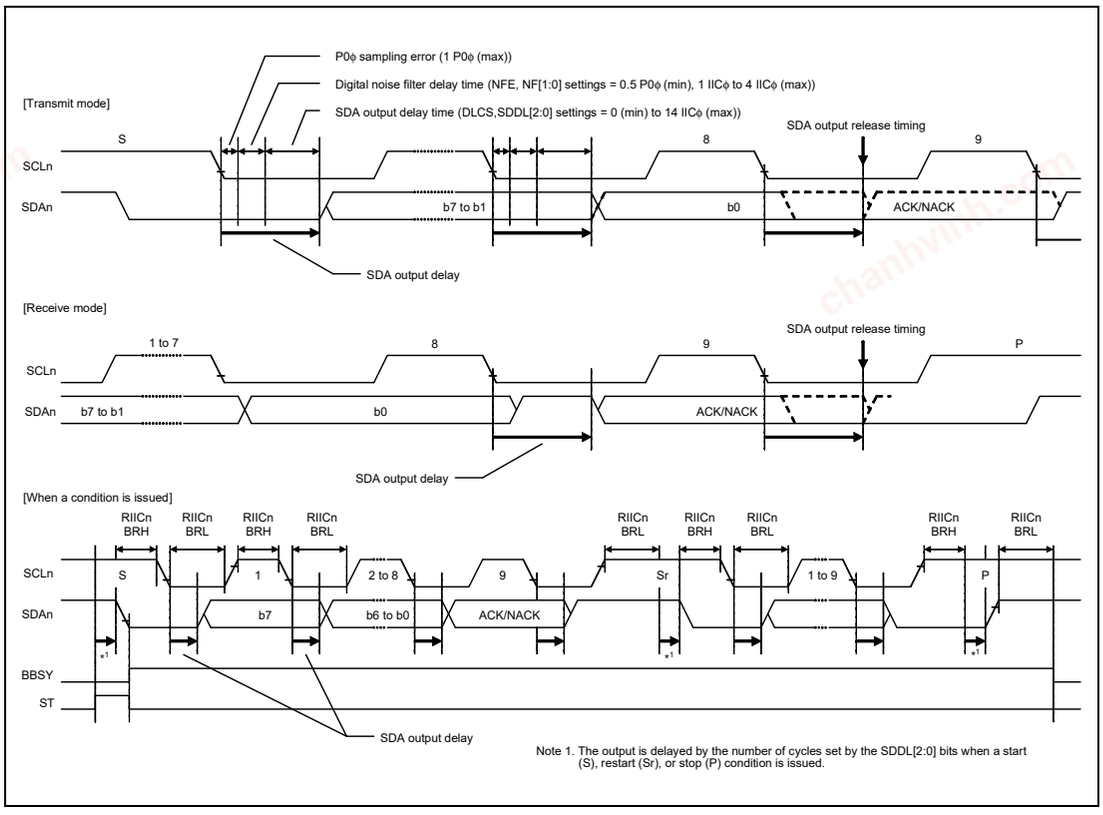


Figure 18.22 SDA Output Delay Facility

chanhvinh.com

18.8 Digital Noise-Filter Circuits

Figure 18.23 is a block diagram of the digital noise-filter circuit. When the NFE bit in the RIICnFER register is set to 1, input to the RIICnSCL and RIICnSDA pins are conveyed to the internal circuitry through digital noise-filter circuits.

The on-chip digital noise-filter circuit of the RIIC consists of four flip-flop circuit stages connected in series and a match-detection circuit.

The number of effective stages in the digital noise filter is selected by the RIICnMR3.NF[1:0] bits. The selected number of effective stages determines the noise-filtering capability as a period from one to four IICϕ cycles.

The input signal to the RIICnSCL pin (or RIICnSDA pin) is sampled on falling edges of the IICϕ signal. When the input signal level matches the output level of the number of effective flip-flop circuit stages as selected by the RIICnMR3.NF[1:0] bits, the signal level is conveyed to the subsequent stage. If the signal levels do not match, the previous value is retained.

Note that if the ratio of the frequencies for P0ϕ and IICϕ is small (when the RIICnMR1.CKS[2:0] bits are set to "000_B"), even necessary signals might be eliminated due to the characteristics of the digital noise filter.

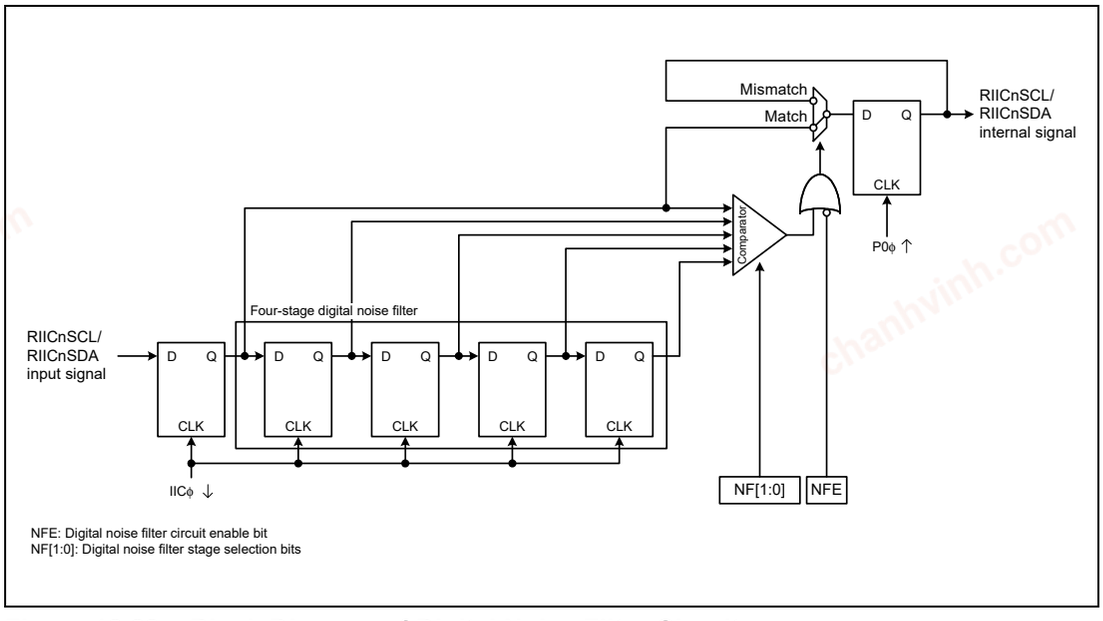


Figure 18.23 Block Diagram of Digital Noise Filter Circuit

chanhvinh.com

18.9 Address Match Detection

The RIIC can set three unique slave addresses in addition to the general call address and host address, and also can set 7-bit or 10-bit slave addresses.

18.9.1 Slave-Address Match Detection

The RIIC can set three unique slave addresses, and has a slave address detection function for each unique slave address. When the RIICnSER.SARy bit (y = 0 to 2) is set to 1, the slave addresses set in RIICnSARy (y = 0 to 2) can be detected.

When the RIIC detects a match of the set slave address, the corresponding RIICnSR1.AASy flag (y = 0 to 2) is set to 1 at the rising edge of the ninth SCL clock cycle, and the RIICnSR2.RDRF flag or the RIICnSR2.TDRE flag is set to 1 by the following R/W# bit. This causes a receive data full interrupt (INTRIICRI) or transmit data empty interrupt (INTRIICTI) to be generated. The AASy flag is used to identify which slave address has been specified.

Figure 18.24 to Figure 18.26 show the AASy flag set timing in three cases.

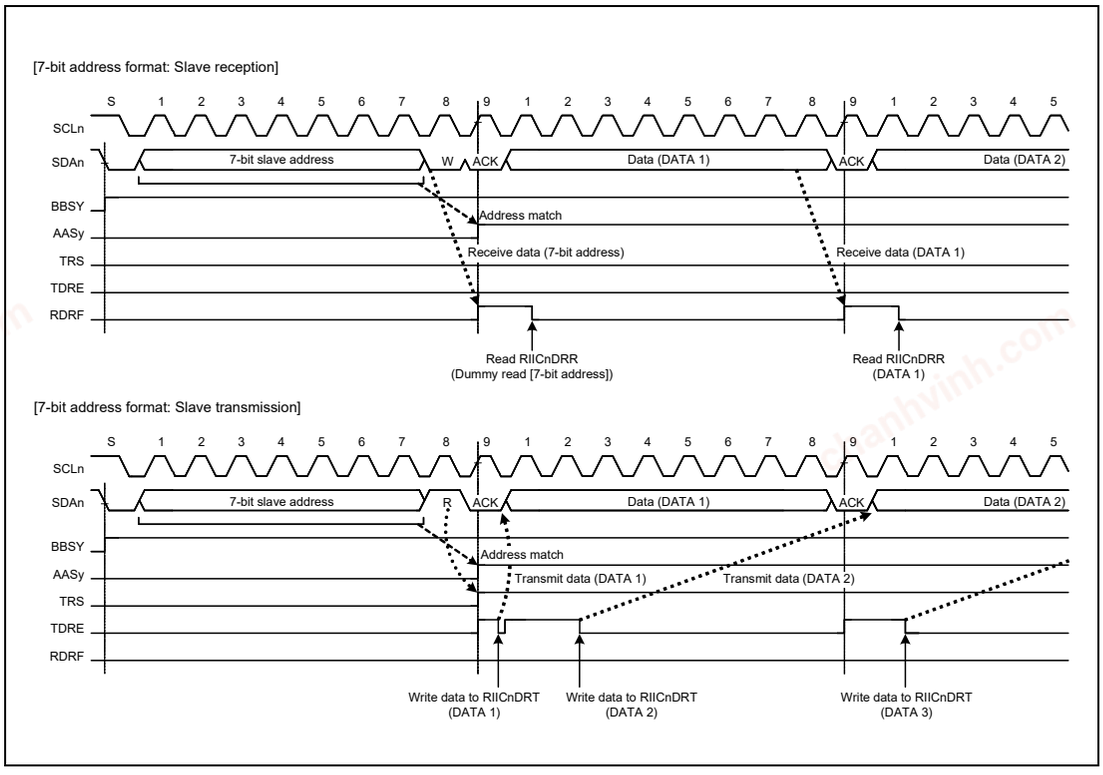


Figure 18.24 AASy Flag Set Timing with 7-Bit Address Format Selected

chanhvinh.com

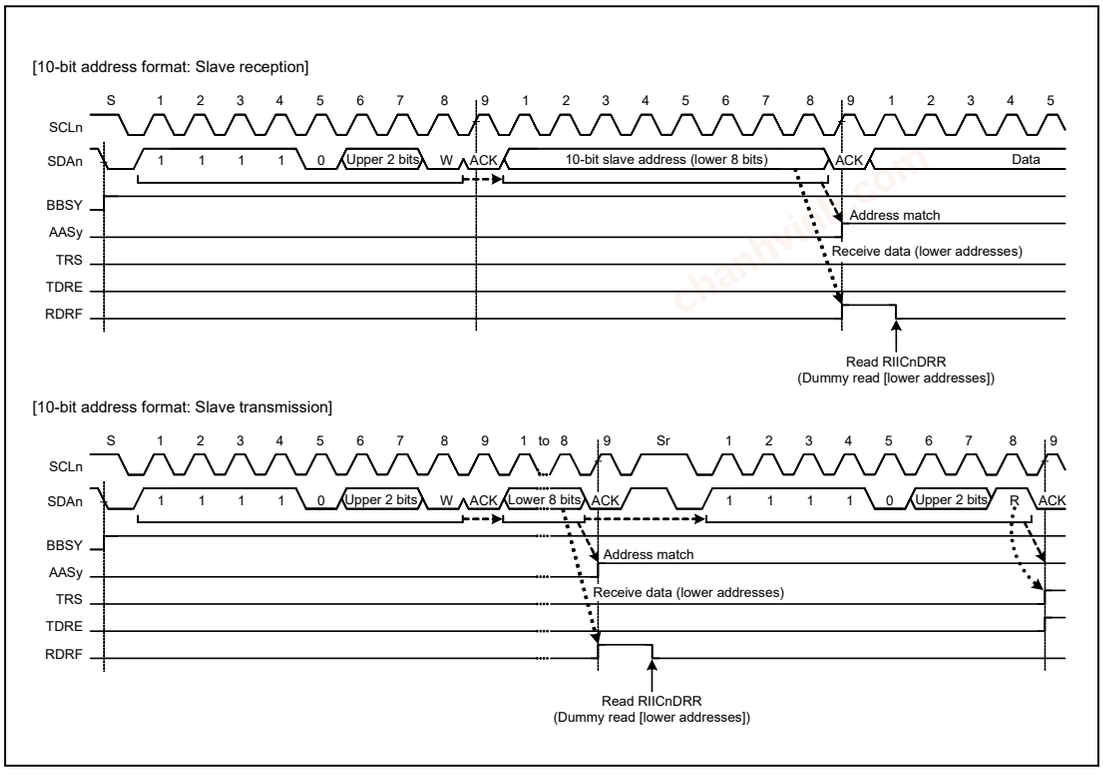


Figure 18.25 AASy Flag Set Timing with 10-Bit Address Format Selected

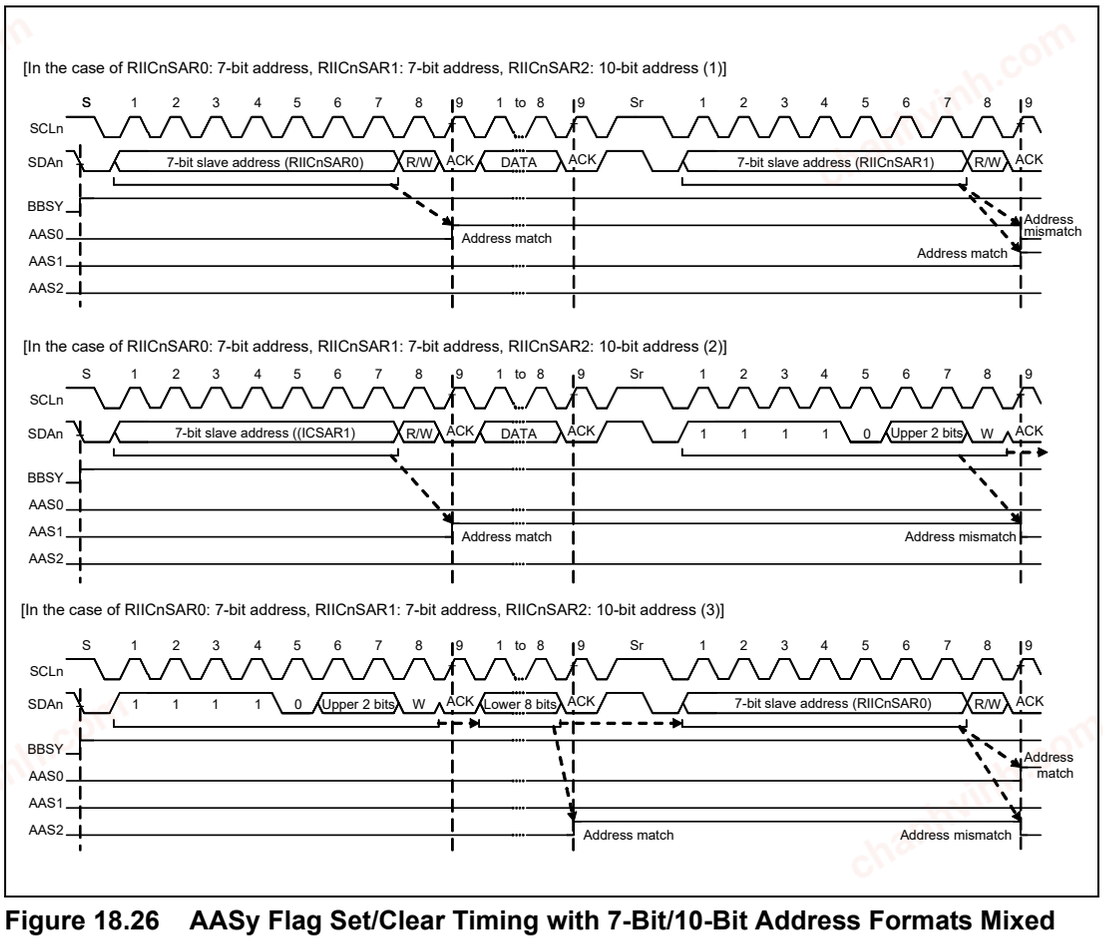


Figure 18.26 AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed

chanhvinh.com

18.9.2 Detection of the General Call Address

The RIIC has a facility for detecting the general call address (0000 000_B + 0 [W]). This is enabled by setting the RIICnSER.GCE bit to 1.

If the address received after a start or restart condition is issued is 0000 000_B + 1[R] (start byte), the RIIC recognizes this as the address of a slave device with an “all-zero” address but not as the general call address.

When the RIIC detects the general call address, both the RIICnSR1.GCA flag and the RIICnSR2.RDRF flag are set to 1 on the rising edge of the ninth cycle of SCL clock. This leads to the generation of a receive data full interrupt (INTRIICRI). The value of the GCA flag can be confirmed to recognize that the general call address has been transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

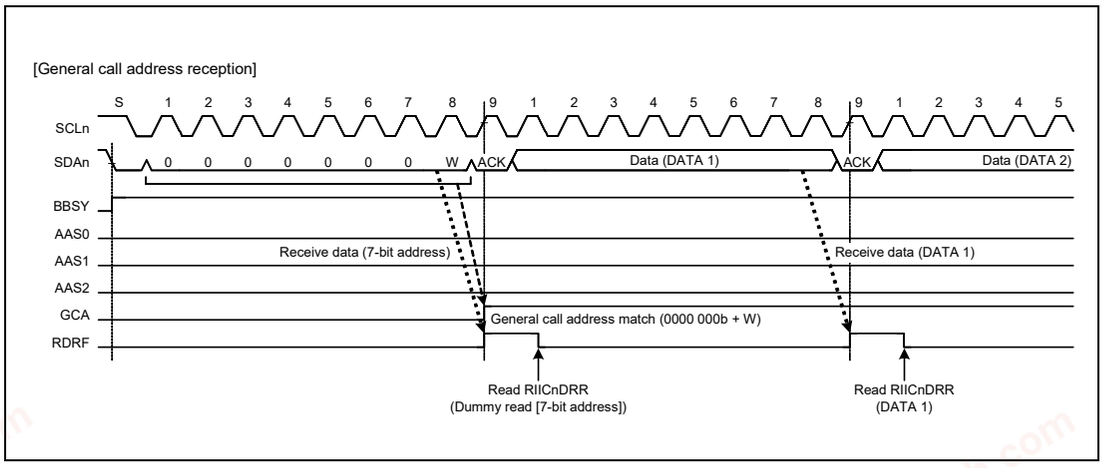


Figure 18.27 Timing of GCA Flag Setting during Reception of General Call Address

chanhvinh.com

18.9.3 Device-ID Address Detection

The RIIC module has a facility for detecting device-ID addresses conformant with the I^2C bus specification (Rev. 03). When the RIIC receives 1111 100_B as the first byte after a start condition or restart condition was issued with the RIICnSER.DIDE bit set to 1, the RIIC recognizes the address as a device ID, sets the RIICnSR1.DID flag to 1 on the rising edge of the ninth SCL clock cycle when the following R/W# bit is 0, and then compares the second and subsequent bytes with its own slave address. If the address matches the value in the slave address register, the RIIC sets the corresponding RIICnSR1.AASy flag (y = 0 to 2) to 1.

After that, when the first byte received after a start or restart condition is issued matches the device ID address (1111 100_B) again and the following R/W# bit is 1, the RIIC does not compare the second and subsequent bytes and sets the RIICnSR2.TDRE flag to 1.

In the device-ID address detection function, the RIIC clears the DID flag to 0 if a match with the RIIC's own slave address is not obtained or a match with the device ID address is not obtained after a match with the RIIC's own slave address and the detection of a restart condition. If the first byte after detection of a start or restart condition matches the device ID address (1111 100_B) and the R/W# bit is 0, the RIIC sets the DID flag to 1 and compares the second and subsequent bytes with the RIIC's slave address. If the R/W# bit is 1, the DID flag holds the previous value and the RIIC does not compare the second and subsequent bytes. Therefore, the reception of a device-ID address can be checked by reading the DID flag after confirming that TDRE = 1.

Furthermore, prepare the device-ID fields (three bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device-ID field as normal data for transmission. For details, see I^2C Bus Standard from NXP Semiconductors.

chanhvinh.com

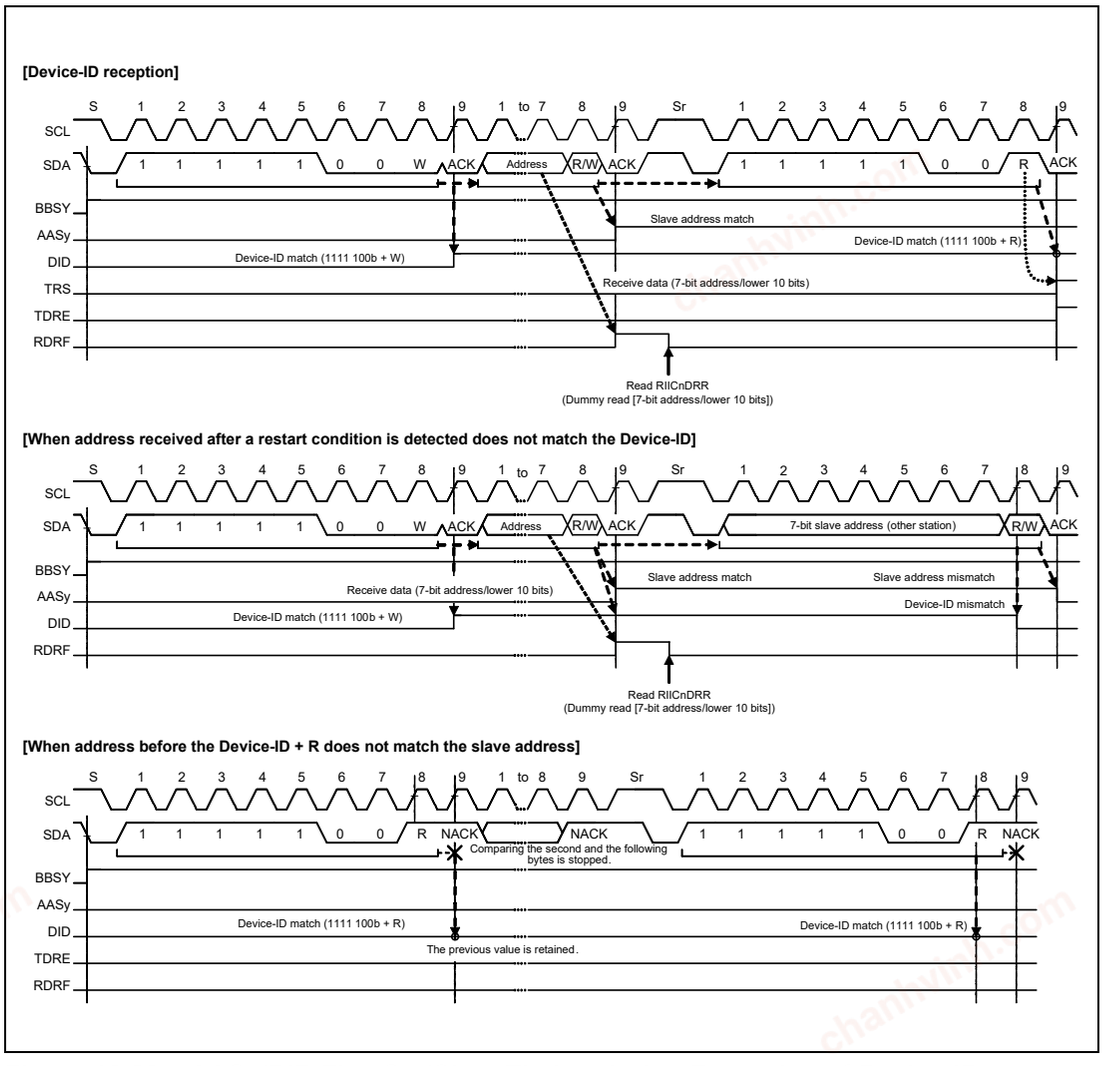


Figure 18.28 AASy/DID Flag Set/Clear Timing during Reception of Device-ID

chanhvinh.com

18.9.4 Host Address Detection

The RIIC has a function to detect the host address while the SMBus is operating. When the RIICnSER.HOAE bit is set to 1 while the RIICnMR3.SMBS bit is 1, the RIIC can detect the host address ($0001\ 000_B$) in slave receive mode (RIICnCR2.MST and TRS bits = 00_B).

When the RIIC detects the host address, the RIICnSR1.HOA flag is set to 1 at the rising edge of the ninth SCL clock cycle, and at the same time, the RIICnSR2.RDRF flag is set to 1 when the R/W# bit is 0 (Wr bit). This causes a receive data full interrupt (INTRIICRI) to be generated. The HOA flag is used to recognize that the host address was sent from the smart battery or other devices.

If the bit following the host address ($0001\ 000_B$) is an Rd bit (R/W# bit = 1), the RIIC can also detect the host address. After the host address is detected, the RIIC operates in the same manner as normal slave operation.

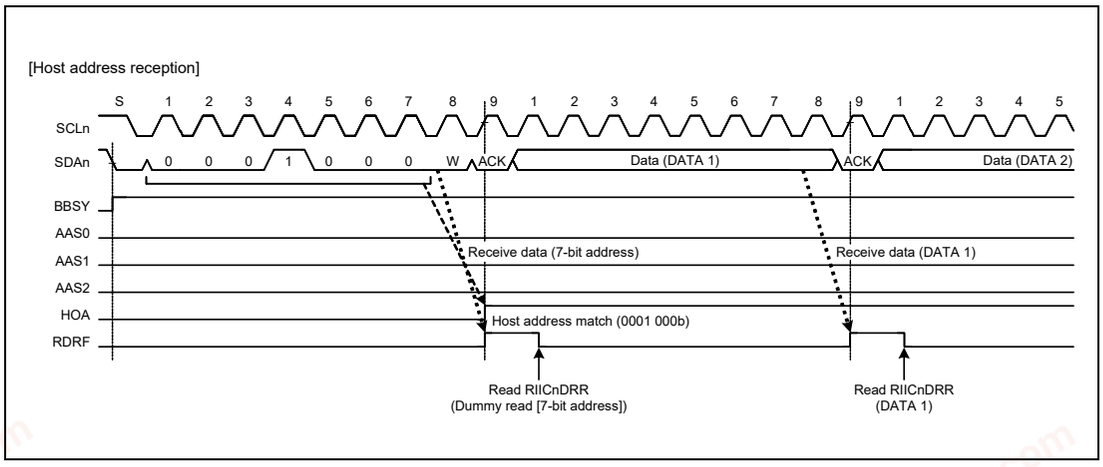


Figure 18.29 HOA Flag Set Timing during Reception of Host Address

chanhvinh.com

18.10 Automatically Low-Hold Function for SCL

18.10.1 Function to Prevent Wrong Transmission of Transmit Data

If the shift register (RIICnDRS) is empty when data have not been written to the transmit data register (RIICnDRT) with the RIIC in transmission mode (RIICnCR2.TRS bit = 1), the SCL signal is automatically held at the low level over the intervals shown below. This low-hold period is extended until data for transmission have been written, which prevents the unintended transmission of erroneous data.

<Master transmit mode>

- Low-level interval after a start condition or restart condition is issued
- Low-level interval of one clock cycle between the ninth clock cycle of one transfer and the first clock cycle of the next

<Slave transmit mode>

- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next

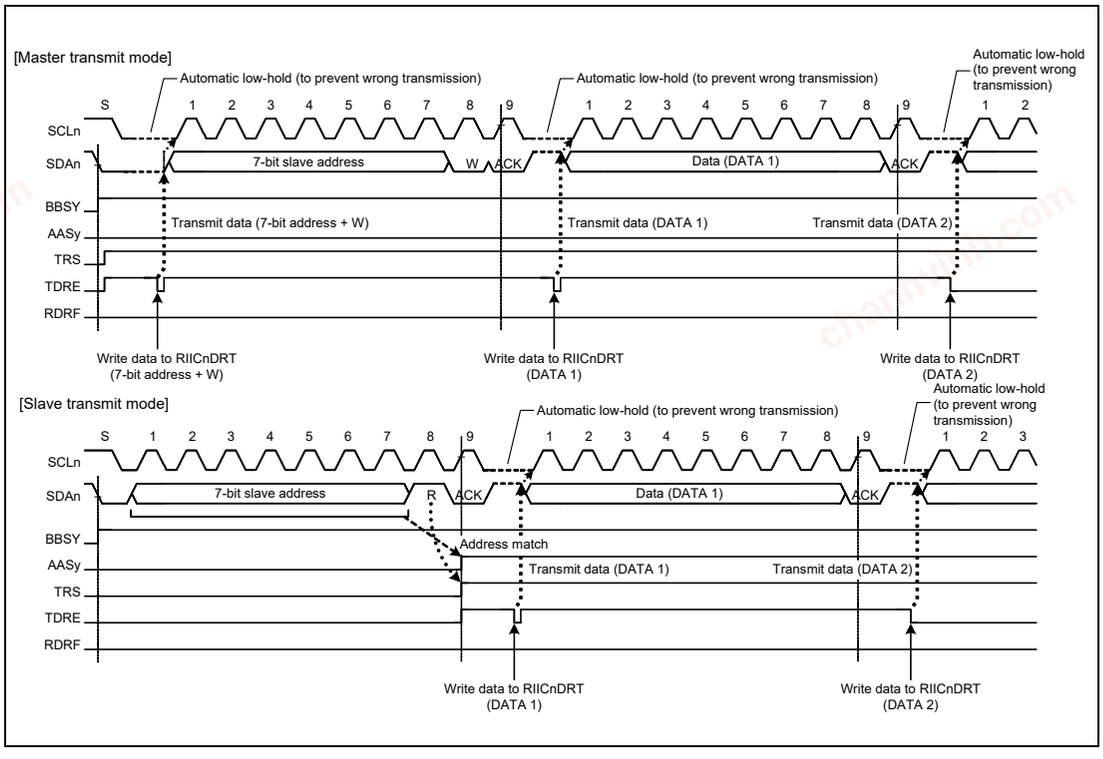


Figure 18.30 Automatic Low-Hold Operation in Transmit Mode

chanhvinh.com

18.10.2 NACK Reception Transfer Suspension Function

The RIIC has a function to suspend transfer operation when NACK is received in transmit mode (RIICnCR2.TRS bit = 1). This function is enabled when the RIICnFER.NACKE bit is set to 1 (transfer suspension enabled). If the next transmit data has already been written (RIICnSR2.TDRE flag = 0) when NACK is received, next data transmission at the falling edge of the ninth SCL clock cycle is automatically suspended. This prevents the SDA line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is suspended by this function (RIICnSR2.NACKF flag = 1), transmit operation and receive operation are discontinued. To restore transmit/receive operation, be sure to clear the NACKF flag to 0. In master transmit mode, clear the NACKF flag to 0 after issuing a restart condition or clear the NACKF and STOP flags to 0 after confirming that a stop condition has been issued, and then issue a start condition.

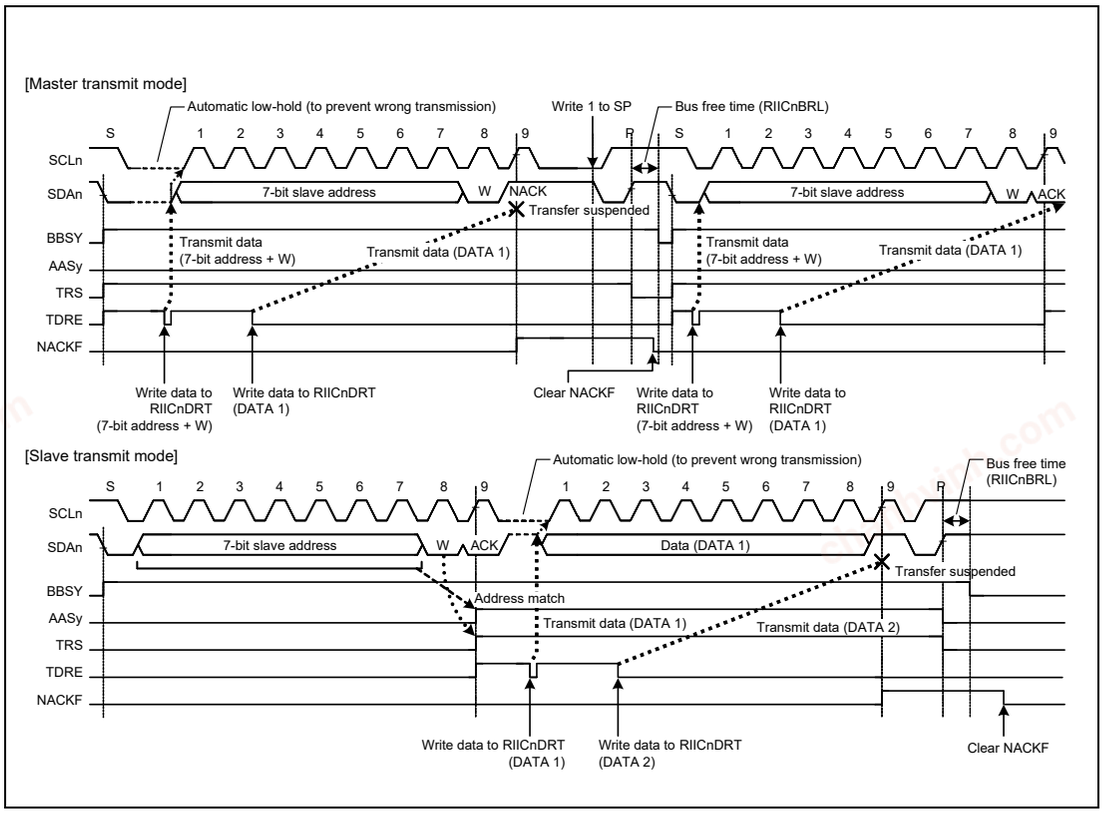


Figure 18.31 Suspension of Data Transfer when NACK is Received (NACKE = 1)

chanhvinh.com

18.10.3 Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (RIICnDRR) read is delayed for a period of one transfer frame or more with receive data full (RIICnSR2.RDRF flag = 1) in receive mode (RIICnCR2.TRS = 0), the RIIC holds the SCL line low automatically immediately before the next data is received to prevent failure to receive data.

This function to prevent failure to receive data using the automatic low-hold function is also enabled even if the read processing of the final receive data is delayed and, in the meantime, the RIIC's own slave address is designated after a stop condition is issued. This function does not disturb other communication because the RIIC does not hold the SCL line low when a mismatch with its own slave address occurs after a stop condition is issued.

Sections in which the SCL line is held low can be selected with a combination of the RIICnMR3.WAIT and RDRFS bits.

(1) One-Byte Receive Operation and Automatic Low-Hold Function Using the WAIT Bit

When the RIICnMR3.WAIT bit is set to 1, the RIIC performs one-byte receive operation using the WAIT bit function.
Furthermore, when the RIICnMR3.RDRFS bit is 0, the RIIC automatically sends the RIICnMR3.ACKBT bit value for the acknowledge bit in the period from the falling edge of the eighth SCL clock cycle to the falling edge of the ninth SCL clock cycle, and automatically holds the SCL line low at the falling edge of the ninth SCL clock cycle using the WAIT bit function. This low-hold is released by reading data from RIICnDRR, which enables bytewise receive operation.

The WAIT bit function is enabled for receive frames after a match with the RIIC's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

(2) One-Byte Receive Operation (ACK/NACK Transmission Control) and Automatic Low-Hold Function Using the RDRFS Bit

When the RIICnMR3.RDRFS bit is set to 1, the RIIC performs one-byte receive operation using the RDRFS bit function.
When the RIICnSR2.RDRFS bit is set to 1, the RDRF flag (receive data full) in RIICnSR2 is set to 1 at the rising edge of the eighth SCL clock cycle, and the SCL line is automatically held low at the falling edge of the eighth SCL clock cycle. This lowhold is released by writing a value to the RIICnMR3.ACKBT bit, but cannot be released by reading data from RIICnDRR, which enables receive operation by the ACK/NACK transmission control according to the data received in byte units.

The RDRFS bit function is enabled for receive frames after a match with the RIIC's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

chanhvinh.com

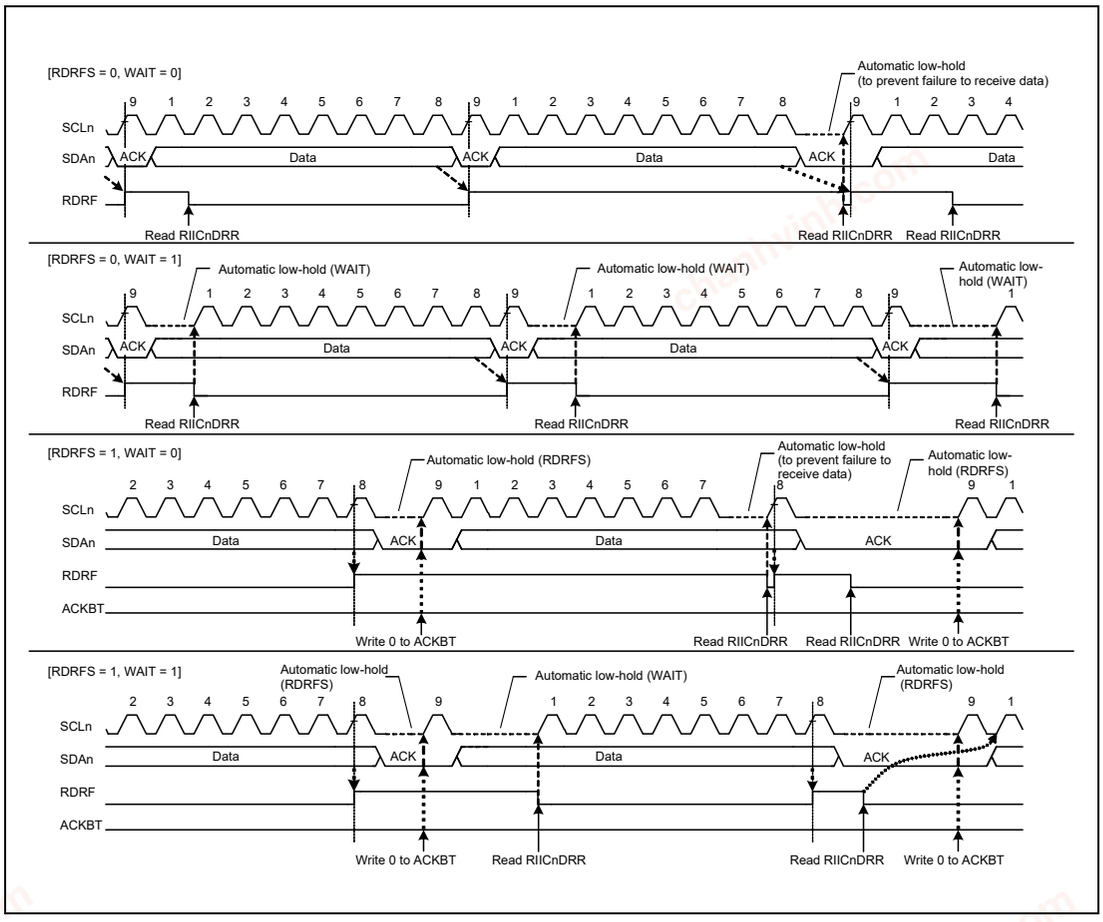


Figure 18.32 Automatic Low-Hold Operation in Receive Mode (Using RDRFS and WAIT Bits)

chanhvinh.com

18.11 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the I^2C bus standard, the RIIC has functions to prevent double-issue of a start condition, to detect arbitration-lost during transmission of NACK, and to detect arbitration-lost in slave transmit mode.

18.11.1 Master Arbitration-Lost Detection (MALE Bit)

The RIIC drives the SDA line low to issue a start condition. However, if the SDA line has already been driven low by another master device issuing a start condition, the RIIC regards its own issuing of a start condition as an error and considers this a loss in arbitration, so priority is given to transfer by the other master device. Similarly, if a request to issue a start condition is made by setting the RIICnCR2.ST bit to 1 while the bus is busy (RIICnCR2.BBSY flag = 1), the RIIC regards this as a double-issuing-of-start-condition error and considers itself to have lost in arbitration, thus preventing a failure of transfer due to issuing of a start condition while transfer is in progress.

When a start condition is issued successfully, if the data for transmission including the address bits (i.e. the internal SDA output level) and the level on the SDA line do not match (the high output as the internal SDA output; i.e. the SDA pin is in the high-impedance state) and the low level is detected on the SDA line, the RIIC loses in arbitration.

After a loss in arbitration of mastership, the RIIC immediately enters slave receive mode. If a slave address (including the general call address) matches its own address at this time, the RIIC continues in slave operation.

A loss in arbitration of mastership is detected when the following conditions are met while the RIICnFER.MALE bit is 1 (master arbitration-lost detection enabled).

[Master arbitration-lost conditions]

- Non-matching of the internal level for output on SDA and the level on the SDA line after a start condition was issued by setting the RIICnCR2.ST bit to 1 while the RIICnCR2.BBSY flag was cleared to 0 (erroneous issuing of a start condition)
- Setting of the RIICnCR2.ST bit to 1 (start condition double-issue error) while the RIICnCR2.BBSY flag is set to 1
- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the SDA line in master transmit mode (RIICnCR2.MST and TRS bits = 11_B)

chanhvinh.com

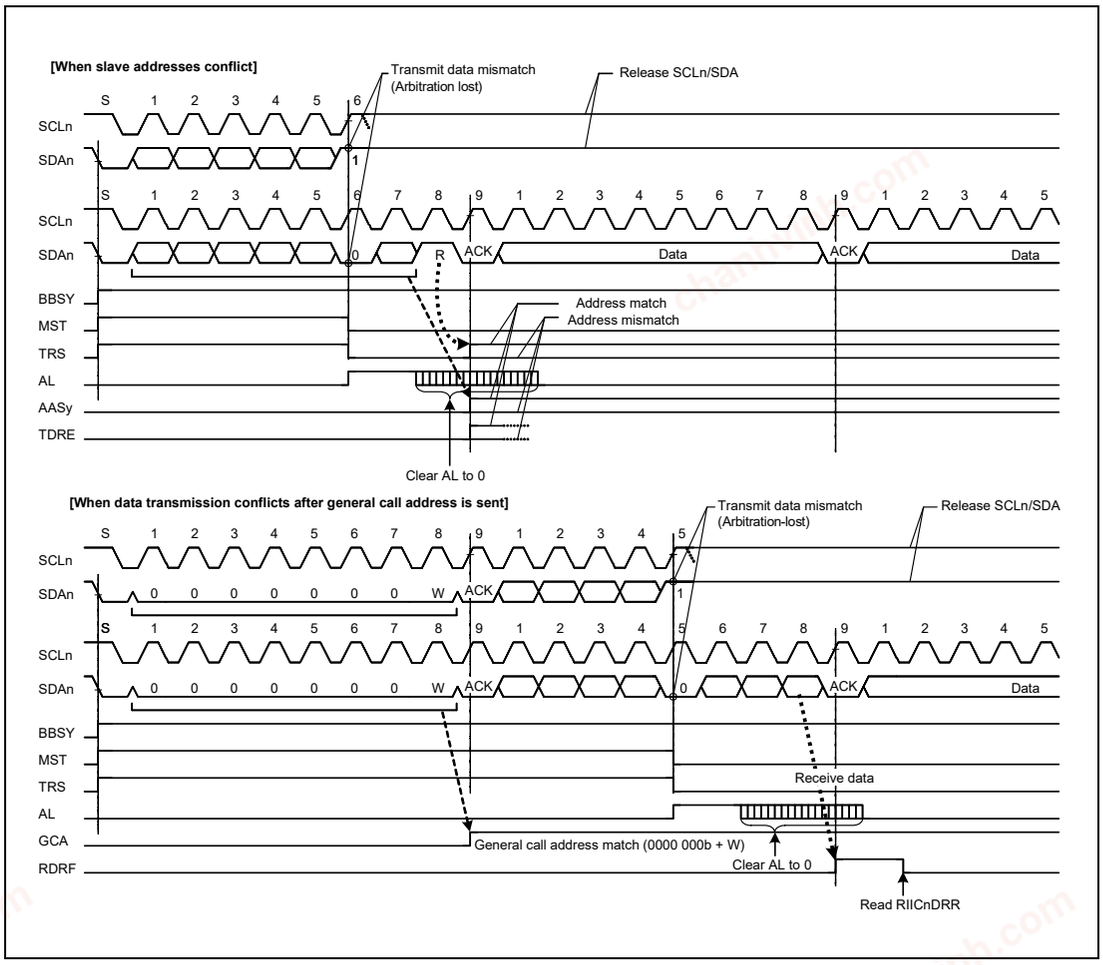


Figure 18.33 Examples of Master Arbitration-Lost Detection (MALE = 1)

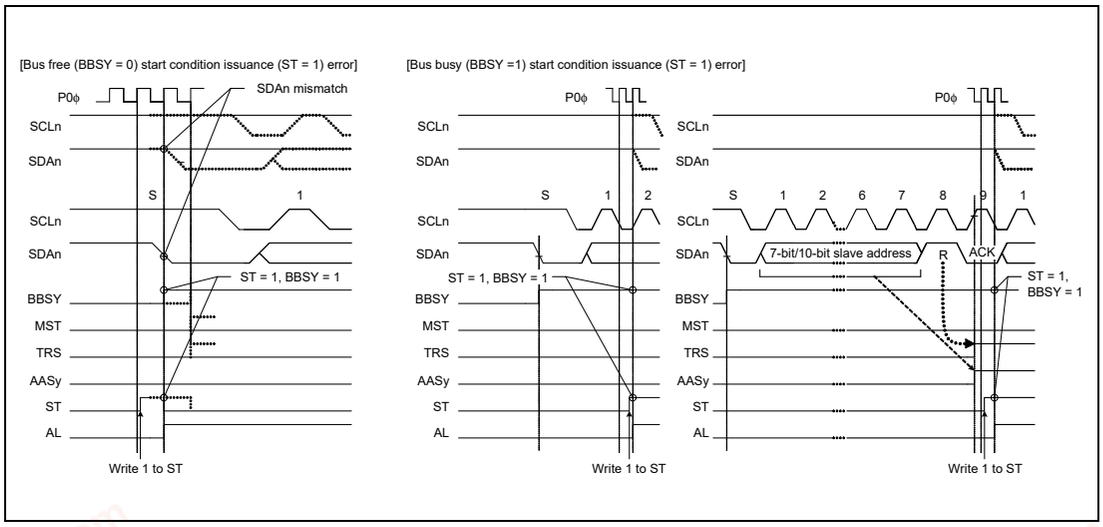


Figure 18.34 Arbitration-Lost when a Start Condition is Issued (MALE = 1)

chanhvinh.com

18.11.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit)

The RIIC has a function to cause arbitration to be lost if the internal SDA output level does not match the level on the SDA line (the high output as the internal SDA output; i.e. the SDA pin is in the high-impedance state) and the low level is detected on the SDA line during transmission of NACK in receive mode. Arbitration is lost due to a conflict of NACK transmission and ACK transmission when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such conflict occurs when multiple master devices send/receive the same information through a single slave device. Figure 18.35 shows an example of arbitration-lost detection during transmission of NACK.

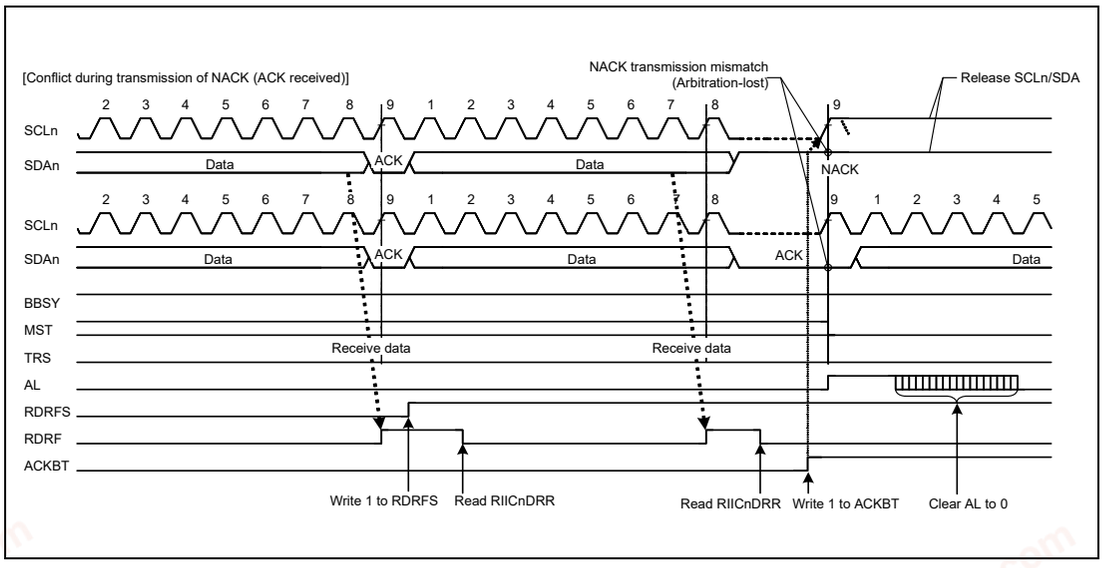


Figure 18.35 Example of Arbitration-Lost Detection during Transmission of NACK (NALE = 1)

The following explains arbitration-lost detection using an example where two master devices (master A and master B) and a single slave device are connected through the bus. In this example, master A receives two bytes of data from the slave device, and master B receives four bytes of data from the slave device.

If master A and master B access the slave device simultaneously, because the slave address is identical, arbitration is not lost in both master A and master B during access to the slave device. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. Here, master A sends NACK when it has received two final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received necessary four bytes of data. At this time, the NACK transmission from master A and the ACK transmission from master B conflict. In general, if a conflict like this occurs, master A cannot detect ACK transmitted by master B and issues a stop condition.

Therefore, the issuance of the stop condition conflicts with the SCL clock output of master B, which disturbs communication.

When the RIIC receives ACK during transmission of NACK, it detects a defeat in conflict with other master devices and causes arbitration to be lost.

If arbitration is lost during transmission of NACK, the RIIC enters slave receive mode. This prevents a stop condition from being issued, preventing a communication failure on the bus.

Similarly, in the ARP command processing of SMBus, the function to detect loss of arbitration during transmission of NACK is also available for eliminating the extra clock cycle processing (such as FF_H

chanhvinh.com

transmission processing) necessary if the UDID (Unique Device Identifier) of assign address does not match in the Get UDID (general) processing after the Assign address command.

The RIIC detects arbitration-lost during transmission of NACK when the following condition is met with the RIICnFER.NALE bit set to 1 (arbitration-lost detection during NACK transmission enabled).

[Condition for arbitration-lost during NACK transmission]

When the internal SDA output level does not match the SDA line (ACK is received) during transmission of NACK (RIICnMR3.ACKBT bit = 1)

18.11.3 Slave Arbitration-Lost Detection (SALE Bit)

The RIIC has a function to cause arbitration to be lost if the data for transmission (i.e. the internal SDA output level) and the level on the SDA line do not match (the high output as the internal SDA output; i.e. the SDA pin is in the highimpedance state) and the low level is detected on the SDA line in slave transmit mode. This arbitration-lost detection function is mainly used when transmitting a UDID (Unique Device Identifier) over an SMBus.

When it loses slave arbitration, the RIIC enters slave receive mode. This function can detect conflicts of data during transmission of UDIDs over an SMBus and eliminates

subsequent redundant processing (processing for the transmission of FF_H).

The RIIC detects slave arbitration-lost when the following condition is met with the RIICnFER.SALE bit set to 1 (slave arbitration-lost detection enabled).

[Condition for slave arbitration-lost]

When transmit data excluding acknowledge (internal SDA output level) does not match the SDA line in slave transmit mode (RIICnCR2.MST and TRS bits = 01_B)

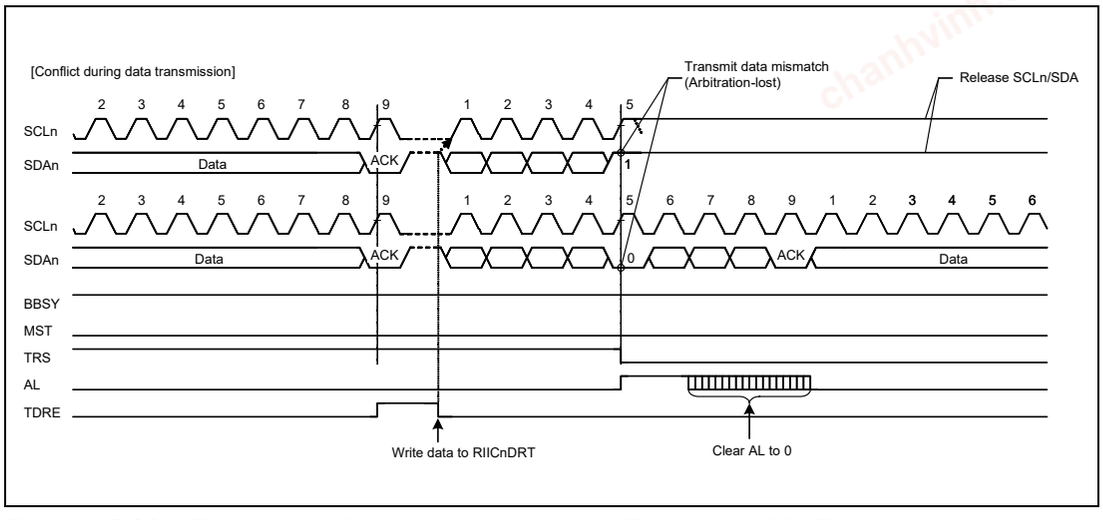


Figure 18.36 Example of Slave Arbitration-Lost Detection (SALE = 1)

chanhvinh.com

18.12 Start Condition/Restart Condition/Stop Condition Issuing Function

18.12.1 Issuing a Start Condition

The RIIC issues a start condition when the RIICnCR2.ST bit is set to 1.

When the ST bit is set to 1, a start condition issuance request is made and the RIIC issues a start condition when the RIICnCR2.BBSY flag is 0 (bus free). When a start condition is issued normally, the RIIC automatically shifts to the master transmit mode.

A start condition is issued in the following sequence.

[Start condition issuance]

- Drive the SDA line low (high level to low level).
- Ensure the time set in RIICnBRH and the start condition hold time.
- Drive the SCL line low (high level to low level).
- Detect low level of the SCL line and ensure the low-level period of SCL line set in RIICnBRL.

18.12.2 Issuing a Restart Condition

The RIIC issues a restart condition when the RIICnCR2.RS bit is set to 1.

When the RS bit is set to 1, a restart condition issuance request is made and the RIIC issues a restart condition when the RIICnCR2.BBSY flag is 1 (bus busy) and the RIICnCR2.MST bit is 1 (master mode).

A restart condition is issued in the following sequence.

[Restart condition issuance]

- Release the SDA line.
- Ensure the low-level period of SCL line set in RIICnBRL.
- Release the SCL line (low level to high level).
- Detect a high level of the SCL line and ensure the time set in RIICnBRL and the restart condition setup time.
- Drive the SDA line low (high level to low level).
- Ensure the time set in RIICnBRH and the restart condition hold time.
- Drive the SCL line low (high level to low level).
- Detect a low level of the SCL line and ensure the low-level period of SCL line set in RIICnBRL.

chanhvinh.com

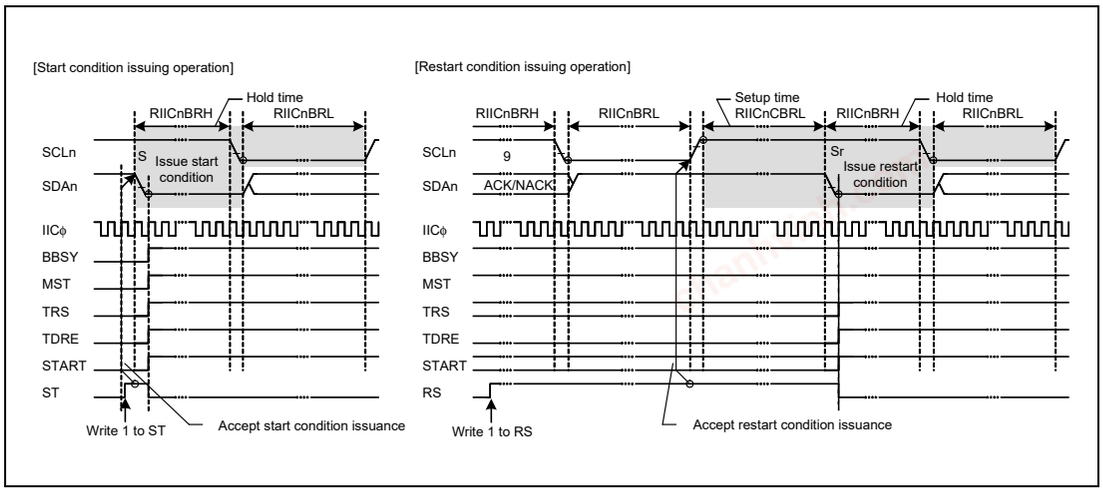


Figure 18.37 Start Condition/Restart Condition Issue Timing (ST and RS Bits)

chanhvinh.com

18.12.3 Issuing a Stop Condition

The RIIC issues a stop condition when the RIICnCR2.SP bit is set to 1.

When the SP bit is set to 1, a stop condition issuance request is made and the RIIC issues a stop condition when the RIICnCR2.BBSY flag is 1 (bus busy) and the RIICnCR2.MST bit is 1 (master mode).

A stop condition is issued in the following sequence.

[Stop condition issuance]

- Drive the SDA line low (high level to low level).
- Ensure the low-level period of SCL line set in RIICnBRL.
- Release the SCL line (low level to high level).
- Detect a high level of the SCL line and ensure the time set in RIICnBRH and the stop condition setup time.
- Release the SDA line (low level to high level).
- Ensure the time set in RIICnBRL and the bus free time.
- Clear the BBSY flag to 0 (to release the bus mastership).

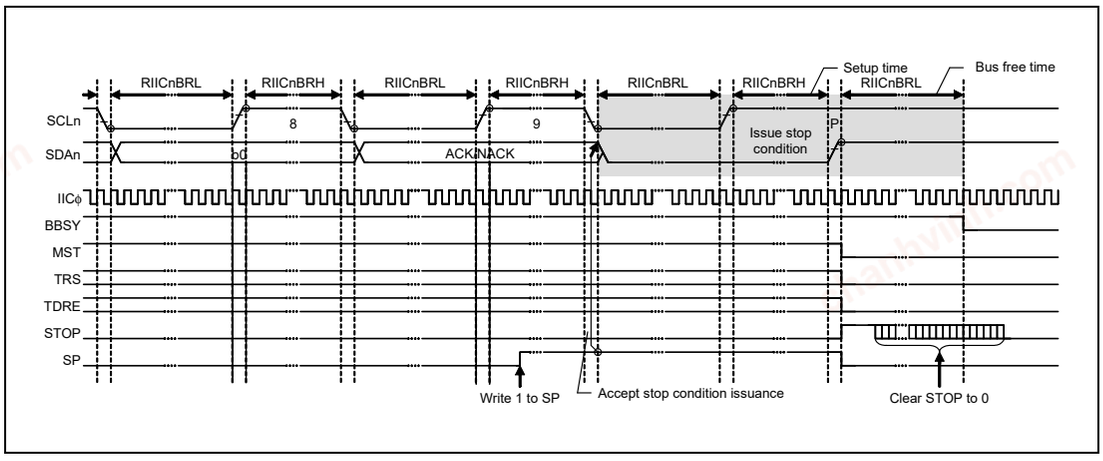


Figure 18.38 Stop Condition Issue Timing (SP Bit)

chanhvinh.com

18.13 Bus Hanging

If the clock signals from the master and slave devices go out of synchronization due to noise or other factors, the I^2C bus might hang with a fixed level on the SCL line and/or SDA line.

As measures against the bus hanging, the RIIC has a timeout function to detect hanging by monitoring the SCL line, a function for the output of an extra SCL clock cycle to release the bus from a hung state due to clock signals being out of synchronization, and the RIIC/internal reset function.

By checking the RIICnCR1.SCLO, SDAO, SCLI, and SDAI bits, it is possible to see whether the RIIC or its partner in communications is placing the low level on the SCL or SDA lines.

18.13.1 Timeout Function

The RIIC has the timeout function to detect an abnormality that the SCL line is held for a certain period of time. The RIIC can detect an abnormal bus state by monitoring that the SCL line is held low or high for a predetermined time.

The timeout function monitors the SCL line state and counts the low-level period or high-level period using the internal counter. The timeout function resets the internal counter each time the SCL line changes (rising or falling), but continues to count unless the SCL line changes. If the internal counter overflows due to no SCL line change, the RIIC can detect the timeout and report the bus abnormality.

This timeout function is enabled when the RIICnFER.TMOE bit is 1. It detects an abnormal bus state in which the SCL line is held low or high in the following cases.

<1> The bus is busy (RIICnCR2.BBSY = 1) in master mode (RIICnCR2.MST = 1).

<2> The bus is busy (RIICnCR2.BBSY = 1) and the RIIC's own slave address matches (RIICnSR1 is not 00_H) in slave mode (RIICnCR2.MST = 0).

<3> The bus is free (RIICnCR2.BBSY = 0) and issuing of a start condition is being requested (RIICnCR2.ST = 1).

The internal counter of the timeout function works using the internal reference clock (IICϕ) set by the RIICnMR1.CKS[2:0] bits as a count source. It functions as a 16-bit counter when long mode is selected (RIICnMR2.TMOS bit = 0) or a 14-bit counter when short mode is selected (TMOS bit = 1).

The SCL line level (low/high or both levels) during which this counter is activated can be selected by the setting of the RIICnMR2.TMOH and TMOL bits. If both TMOL and TMOH bits are cleared to 0, the internal counter does not work.

chanhvinh.com

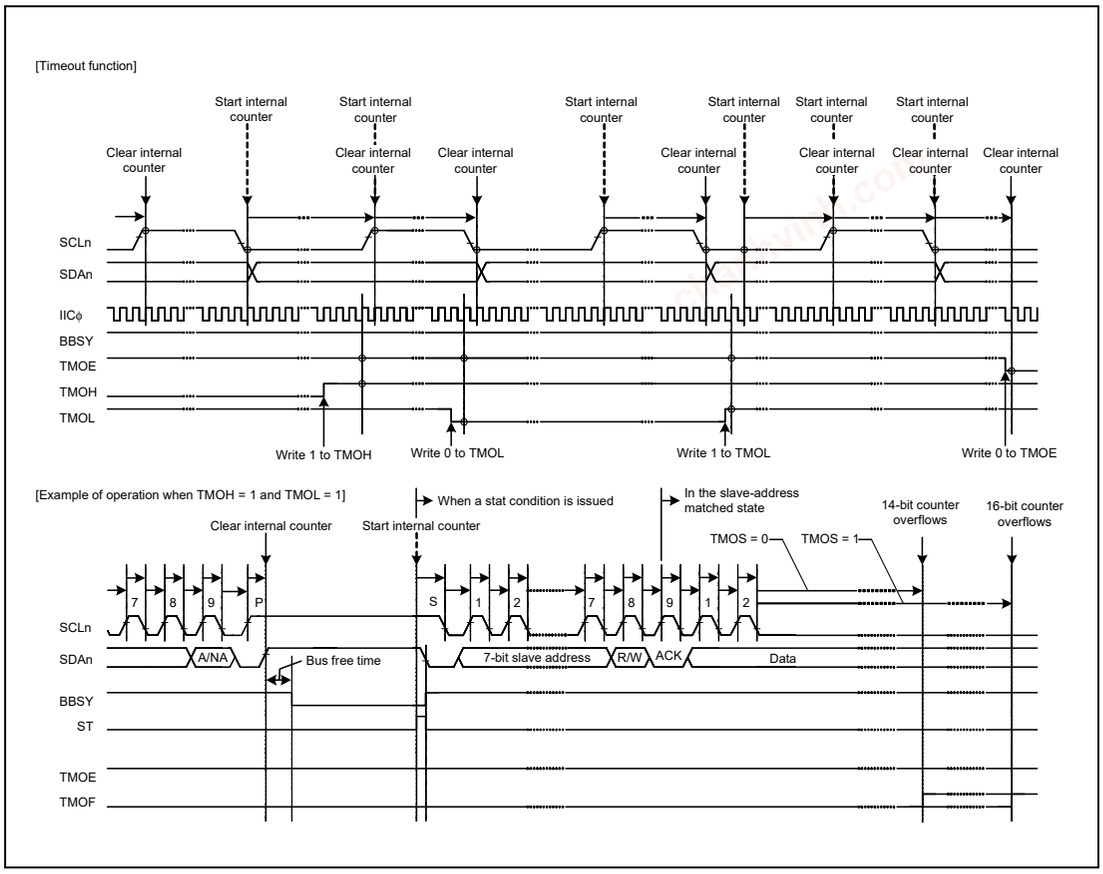


Figure 18.39 Timeout Function (TMOE, TMOS, TMOH, and TMOL Bits)

chanhvinh.com

18.13.2 Extra SCL Clock Cycle Output Function

In master mode, the RIIC module has a facility for the output of extra SCL (clock) cycles to release the SDA line of the slave device from being held at the low level due to the master being out of synchronization with the slave device.

This function is mainly used in master mode to release the SDA line of the slave device from the state of being fixed to the low level by including extra cycles of SCL output from the RIIC with single cycles of the SCL (clock) signal as the unit in the case of a bus error where the RIIC cannot issue a stop condition because the slave device is holding the SDA line at the low level. Do not use this facility in normal situations. Using it when communications are proceeding correctly will lead to malfunctions.

When the RIICnCR1.CLO bit is set to 1 in master mode, a single cycle of the SCL clock at the frequency corresponding to the transfer rate settings (settings of the RIICnMR1.CKS[2:0] bits, and of the RIICnBRH and RIICnBRL registers) is output as an extra clock cycle. After output of this single cycle of the SCL clock, the CLO bit is automatically cleared to 0. In this case, if the BBSY flag is set to 1, the SCL pin continues outputting the low level. If the BBSY flag is set to 0, the SCL pin goes high. In addition, further extra clock cycles can be output consecutively by the software program writing 1 to the CLO bit after having read CLO = 0.

When the RIIC module is in master mode and the slave device is holding the SDA line at the low level because synchronization with the slave device has been lost due to the effects of noise, etc., the output of a stop condition is not possible. The facility for output of an extra cycle of the SCL (clock) signal can be used to output extra cycles of SCL one by one to make the slave device release the SDA line from being held at the low level, thus recovering the bus from an unusable state. Release of the SDA line by the slave device can be monitored by reading the RIICnCR1.SDAI bit. After confirming release of the SDA line by the slave device, complete communications by reissuing the stop condition.

[Output conditions for using the RIICnCR1.CLO bit]

- When the bus is free (RIICnCR2.BBSY flag = 0) or in master mode (RIICnCR2.MST bit = 1 and BBSY flag = 1)
- When the communication device does not hold the SCL line low

Figure 18.40 shows the operation timing of the extra SCL clock cycle output function (CLO bit).

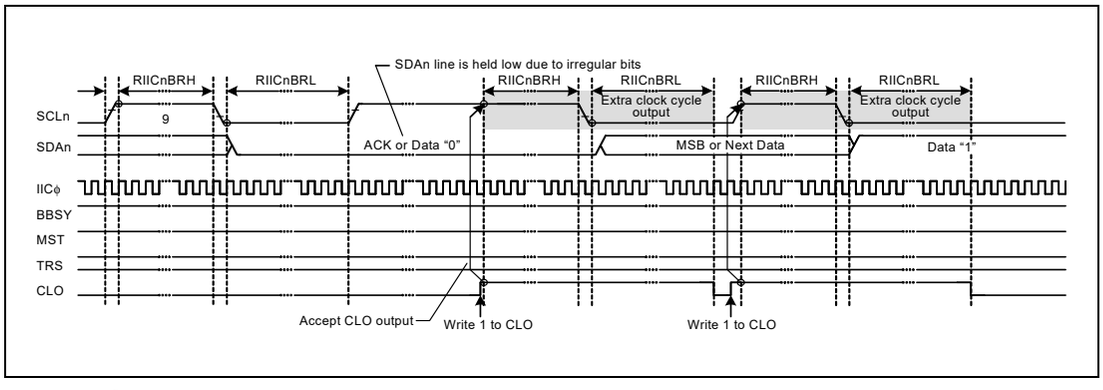


Figure 18.40 Extra SCL Clock Cycle Output Function (CLO Bit)

chanhvinh.com

18.13.3 RIIC Reset and Internal Reset

The RIIC module incorporates a function for resetting itself. There are two types of reset. One is referred to as an RIIC reset; this initializes all registers including the RIICnCR2.BBSY flag. The other is referred to as an internal reset; this releases the RIIC from the slave-address matched state and initializes the internal counter while retaining other settings.

After issuing a reset, be sure to clear the RIICnCR1.IICRST bit to 0.

Both types of reset are effective for release from bus-hung states since both restore the output state of the SCL and SDA pins to the high impedance state.

Issuing a reset during slave operation may lead to a loss of synchronization between the master device clock and the slave device clock, so avoided this where possible. Note that monitoring of the bus state, such as for the presence of a start condition, is not possible during an RIIC reset (RIICnCR1.ICE and IICRST bits = 01_B).

For a detailed description of the RIIC and internal resets, see Section 18.15, Reset Function of RIIC.

chanhvinh.com

18.14 SMBus Operation

The RIIC is available for data communication conforming to the SMBus (Version 2.0). To perform SMBus communication, set the RIICnMR3.SMBS bit to 1. To use the transfer rate within a range of 10 kbps to 100 kbps of the SMBus standard, set the RIICnMR1.CKS[2:0] bits, RIICnCBRH, and RIICnBRL. In addition, determine the values of the RIICnMR2.DLCS bit and the RIICnMR2.SDDL[2:0] bits to meet the data hold time specification of 300 ns or more. If the RIIC is used only as a slave device, the transfer rate setting is not necessary. When the RIIC is used only as a slave device, the transfer rate setting is not necessary, whereas the RIICnBRL needs to be set to a value equal to or longer than the data setup time (250 ns).

For the SMBus device default address ($1100\ 001_B$), use one of the slave address registers L0 to L2 (RIICnSAR0, RIICnSAR1, and RIICnSAR2), and set the corresponding RIICnSARy.FSy bit (7-bit/10-bit address format select) (y = 0 to 2) to 0 (7-bit address format).

When transmitting the UDID (Unique Device Identifier), set the RIICnFER.SALE bit to 1 to enable the slave arbitration lost detection function.

18.14.1 SMBus Timeout Measurement

(1) Measuring timeout of slave device

The following period (timeout interval: $T_{LOW:\ SEXT}$) must be measured for slave devices in SMBus communication.

- From start condition to stop condition

  To measure timeout for slave devices, measure the period from start condition detection to stop condition detection with the internal timer using a start condition detection interrupt (INTRIICSTI) and stop condition detection interrupt (INTRIICSPI) of the RIIC. The measured timeout period must be within the total clock low-level period [slave device] $T_{LOW:\ SEXT}$: 25 ms (max.) of the SMBus standard.
  If the time measured with the internal timer exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (min.) of the SMBus standard, the slave device must release the bus by writing 1 to the RIICnCR1.IICRST bit to issue an internal reset of the RIIC. When an internal reset is issued, the RIIC stops driving the bus for the SCL pin and SDA pin and make the SCL/SDA pin outputs high impedance, which releases the bus.

(2) Measuring timeout of master device

The following periods (timeout interval: $T_{LOW:\ MEXT}$) must be measured for master devices in SMBus communication.

- From start condition to acknowledge bit
- Between acknowledge bits
- From acknowledge bit to stop condition

To measure timeout for master devices, measure these periods with the internal timer using a start condition detection interrupt (INTRIICSTI), stop condition detection interrupt (INTRIICSPI), and transmit end interrupt (INTRIICTEI) or receive data full interrupt (INTRIICRI) of the RIIC. The measured timeout period must be within the total clock low-level extended period [master device] $T_{LOW:\ MEXT}$: 10 ms (max.) of the SMBus standard, and the total of all $T_{LOW:\ MEXT}$ from start condition to stop condition must be within $T_{LOW:\ SEXT}$: 25 ms (max.).

chanhvinh.com

For the ACK receive timing (rising edge of the ninth SMBCLK clock cycle), monitor theRIICnSR2. TEND flag in master transmit mode (master transmitter) and the RIICnSR2.RDRF flag in master receive mode (master receiver). For this reason, perform bytewise transmit operation in master transmit mode, and hold the RIICnMR3.RDRFS bit 0 until the byte just before reception of the final byte in master receive mode. While the RDRFS bit is 0, the RDRF flag is set to 1 at the rising edge of the ninth SMBCLK clock cycle.

If the period measured with the internal timer exceeds the total clock low-level extended period [master device] $T_{LOW:MEXT}$: 10 ms (max.) of the SMBus standard or the total of measured periods exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (min.) of the SMBus standard, the master device must stop the transaction by issuing a stop condition. In master transmit mode, immediately stop the transmit operation (writing data to RIICnDRT).

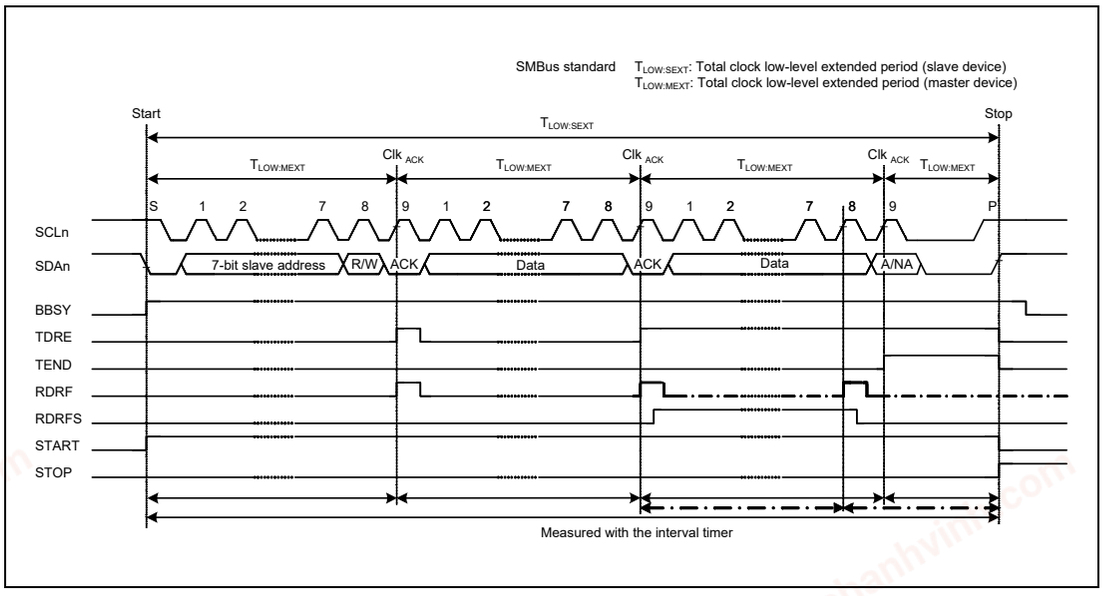


Figure 18.41 SMBus Timeout Measurement

18.14.2 SMBus Host Notification Protocol/Notify ARP Master

In communications over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of (or request the SMBus host for) its own slave address or to request its own slave address from the SMBus host.

For this LSI to operate as an SMBus host (or ARP master), the host address ($0001\ 000_B$) sent from the slave device must be detected as a slave address, so the RIIC has a function for detecting the host address. To detect the host address as a slave address, set the RIICnMR3.SMBS bit and the RIICnSER.HOAE bit to 1. Operation after the host address has been detected is the same as normal slave operation.

chanhvinh.com

18.15 Reset Function of RIIC

The RIIC has chip reset, RIIC reset, and internal reset functions. Table 18.24 lists the scope of each reset and reset conditions.

Table 18.24 RIIC Reset Functions (1/2)

Register		RIIC Reset (ICE = 0, IICRST = 1)	Internal Reset (ICE = 1, IICRST = 1)	Start Condition/ Restart Condition Detection	Stop Condition Detection
RIICnCR1	ICE	0	1	Retained	Retained
	IICRST	1	1	Retained	Retained
	CLO	Initialized	Retained	Retained	Retained
	SOWP	Initialized	Retained	Retained	Retained
	SCLO	Initialized	Initialized	Retained	Retained
	SDAO	Initialized	Initialized	Retained	Retained
	SCLI	Initialized	Retained	Retained	Retained
	SDAI	Initialized	Retained	Retained	Retained
RIICnCR2	BBSY	Initialized	Initialized *1	Operation	Retained
	MST	Initialized	Initialized	Operation (retained)	Initialized
	TRS	Initialized	Initialized	Operation (retained)	Initialized
	SP	Initialized	Initialized	Initialized	Initialized
	RS	Initialized	Initialized	Initialized	Initialized
	ST	Initialized	Initialized	Initialized	Retained
RIICnMR1	CKS[2:0]	Initialized	Retained	Retained	Retained
	BCWP	Initialized	Retained	Retained	Retained
	BC[2:0]	Initialized	Initialized	Initialized	Retained
RIICnMR2		Initialized	Retained	Retained	Retained
RIICnMR3	WAIT	Initialized	Retained	Retained	Retained
	RDRFS	Initialized	Retained	Retained	Retained
	ACKWP	Initialized	Retained	Retained	Retained
	ACKBT	Initialized	Retained	Retained	Initialized
	ACKBR	Initialized	Retained	Retained	Retained
	NF[1:0]	Initialized	Retained	Retained	Retained
RIICnFER		Initialized	Retained	Retained	Retained
RIICnSER		Initialized	Retained	Retained	Retained
RIICnIER		Initialized	Retained	Retained	Retained
RIICnSR1	DID	Initialized	Initialized	Retained	Initialized
	GCA	Initialized	Initialized	Retained	Initialized
	AAS2	Initialized	Initialized	Retained	Initialized
	AAS1	Initialized	Initialized	Retained	Initialized
	AAS0	Initialized	Initialized	Retained	Initialized
RIICnSR2	TDRE	Initialized	Initialized	Retained	Initialized
	TEND	Initialized	Initialized	Retained	Initialized
	RDRF	Initialized	Initialized	Retained	Retained
	NACKF	Initialized	Initialized	Retained	Retained
	STOP	Initialized	Initialized	Retained	Operation
	START	Initialized	Initialized	Operation	Initialized
	AL	Initialized	Initialized	Retained	Retained
	TMOF	Initialized	Initialized	Retained	Retained
RIICnSAR0, RIICnSAR1, RIICnSAR2		Initialized	Retained	Retained	Retained
RIICnBRH, RIICnBRL		Initialized	Retained	Retained	Retained

chanhvinh.com

Table 18.24 RIIC Reset Functions (2/2)

Register	RIIC Reset (ICE = 0, IICRST = 1)	Internal Reset (ICE = 1, IICRST = 1)	Start Condition/ Restart Condition Detection	Stop Condition Detection
RIICnDRT	Initialized	Retained	Retained	Retained
RIICnDRR	Initialized	Retained	Retained	Retained
RIICnDRS	Initialized	Initialized	Retained	Retained

Note 1. When an internal reset is applied while the bus is free after detection of a stop condition, the setting of the BBSY flag is 0 while the bus is free following de-assertion of the internal reset signal.
When an internal reset is applied while the bus is not free, the BBSY flag is not cleared.

chanhvinh.com

19. Serial Sound Interface

The serial sound interface is a module designed to send or receive audio data interface with various devices offering I^2S bus compatibility. It also provides additional modes for other common formats, as well as support for multi-channel mode.

19.1 Features

- Number of channels: Four channels
- Operating mode: Non-compressed mode
  The non-compressed mode supports serial audio streams divided by channels.
- Serves as both a transmitter and a receiver
  Channels 0, 1, and 3 support full-duplex communications.
- Capable of using serial bus format
- Asynchronous transfer takes place between the data buffer and the shift register.
- It is possible to select a dividing ratio for the clock used by the serial bus interface.
- It is possible to control data transmission or reception with DMA transfer and interrupt requests.
  A path is also provided for direct data transfer between this module and the SCUX module.
- Selects the oversampling clock input from among the following pins:
  —AUDIO_CLK (1 to 50 MHz)
  —AUDIO_X1, AUDIO_X2 (when connecting a crystal resonator: 10 to 50 MHz, when used to input external clock: 1 to 50 MHz)
- Includes 8-stage FIFO buffers in transmitter and receiver
- Supports multi-channel mode (TDM mode) in which the SSIWS signal is high only for system word 1 period.
- Supports WS continue mode in which the SSIWS signal is not stopped.
- An interrupt is used to notify the CPU of a change of sampling frequency during transfer.

chanhvinh.com

Figure 19.1 shows a block diagram of this module.

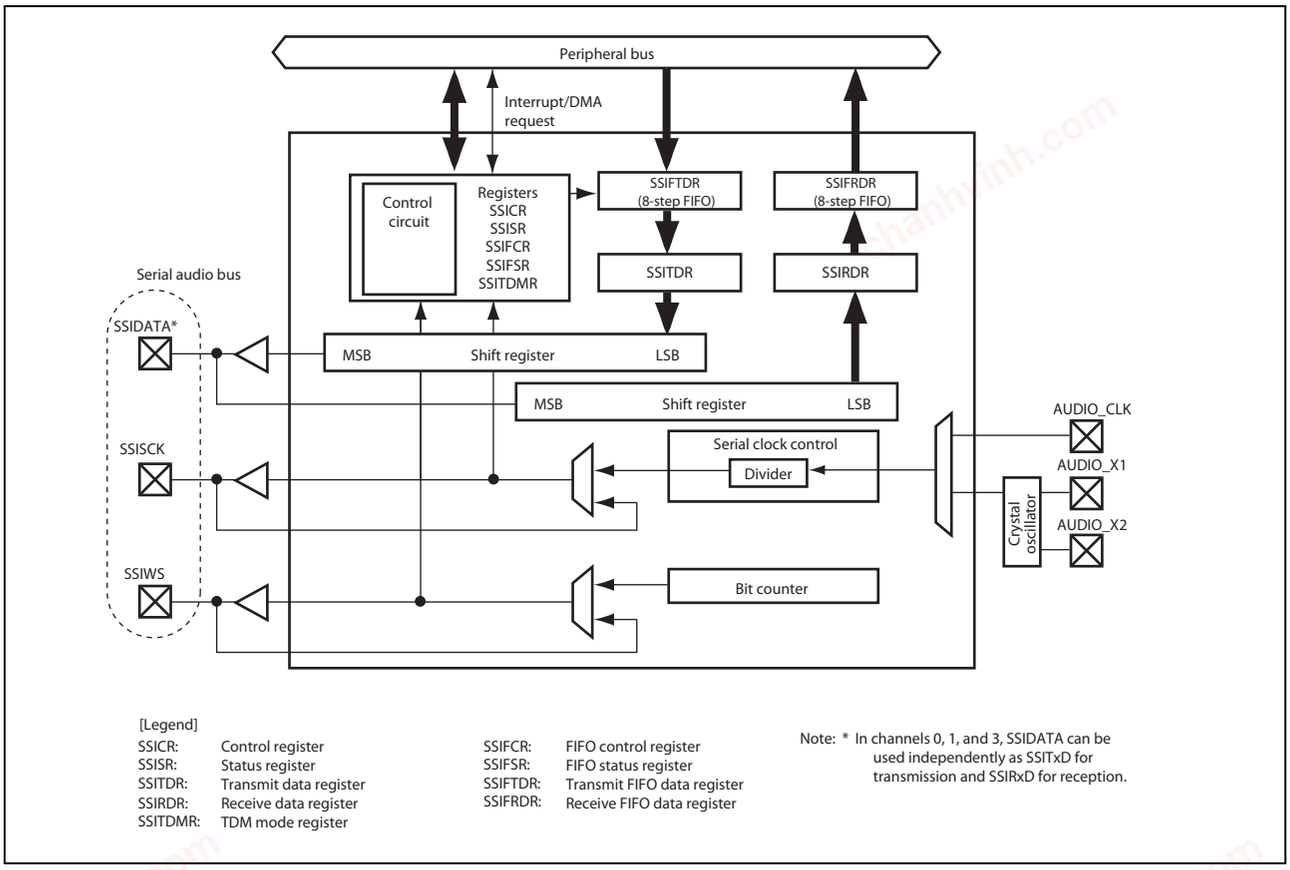


Figure 19.1 Block Diagram of Serial Sound Interface

chanhvinh.com

19.2 Input/Output Pins

Table 19.1 shows the pin assignments relating to this module.

Table 19.1 Pin Assignments

Channel	Pin Name	I/O	Description
0, 1, 3	SSISCK0*1*3, SSISCK1*1*3, SSISCK3*1*3	I/O	Serial bit clock
	SSIWS0*1*3, SSIWS1*1*3, SSIWS3*1*3	I/O	Word selection
	SSITxD0, SSITxD1, SSITxD3	Output	Serial data output
	SSIRxD0*1, SSIRxD1*1, SSIRxD3*1	Input	Serial data input
2	SSISCK2*1*3	I/O	Serial bit clock
	SSIWS2*1*3	I/O	Word selection
	SSIDATA2*1	I/O	Serial data input/output
Common	AUDIO_CLK*2	Input	External clock for audio (input oversampling clock)
	AUDIO_X1	Input	Crystal resonator/external clock for audio (input oversampling clock)
	AUDIO_X2	Output	

Note 1. In slave mode, whether or not to use the noise canceler in the input route can be selected. For details, refer to section 41.3.15, Serial Sound Interface Noise Canceler Control Register (SNCR), under section 41, Ports.

Note 2. When the SSInCKS bit (n = 0 to 3) is set to 1, the MLB_CLK pin is used as the AUDIO_CLK pin. For details, refer to section 37.3.70, SSI Pin Mode Register (SSIPMD_CIM) and section 37.4.4, Pin Connection Specifications of SSIF.

Note 3. Each of SSIF1 to SSIF3 can use SSISCK0 and SSIWS0 of SSIF0 as its own SSISCK and SSIWS. For details, refer to section 37.3.70, SSI Pin Mode Register (SSIPMD_CIM) and section 37.4.4, Pin Connection Specifications of SSIF. When SSIF1 to SSIF3 uses SSIF0 as the master, the noise canceler function cannot be used.

chanhvinh.com

19.3 Register Description

Table 19.2 lists the register configuration. Note that explanation in the text does not refer to the channels.

Table 19.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	Control register 0	SSICR_0	R/W	H'00000000	H'E820B000	32
	Status register 0	SSISR_0	R/W*1	H'02000013	H'E820B004	32
	FIFO control register 0	SSIFCR_0	R/W	H'00000000	H'E820B010	32
	FIFO status register 0	SSIFSR_0	R/(W)*2	H'00010000	H'E820B014	32
	Transmit FIFO data register 0	SSIFTDR_0	W	Undefined	H'E820B018	32
	Receive FIFO data register 0	SSIFRDR_0	R	Undefined	H'E820B01C	32
	TDM mode register 0	SSITDMR_0	R/W	H'00000000	H'E820B020	32
	FC control register 0	SSIFCCR_0	R/W	H'00000000	H'E820B024	32
	FC mode register 0	SSIFCMR_0	R/W	H'00000000	H'E820B028	32
	FC status register 0	SSIFCSR_0	R/(W)*3	H'00000000	H'E820B02C	32
1	Control register 1	SSICR_1	R/W	H'00000000	H'E820B800	32
	Status register 1	SSISR_1	R/W*1	H'02000013	H'E820B804	32
	FIFO control register 1	SSIFCR_1	R/W	H'00000000	H'E820B810	32
	FIFO status register 1	SSIFSR_1	R/(W)*2	H'00010000	H'E820B814	32
	Transmit FIFO data register 1	SSIFTDR_1	W	Undefined	H'E820B818	32
	Receive FIFO data register 1	SSIFRDR_1	R	Undefined	H'E820B81C	32
	TDM mode register 1	SSITDMR_1	R/W	H'00000000	H'E820B820	32
	FC control register 1	SSIFCCR_1	R/W	H'00000000	H'E820B824	32
	FC mode register 1	SSIFCMR_1	R/W	H'00000000	H'E820B828	32
	FC status register 1	SSIFCSR_1	R/(W)*3	H'00000000	H'E820B82C	32
2	Control register 2	SSICR_2	R/W	H'00000000	H'E820C000	32
	Status register 2	SSISR_2	R/W*1	H'02000013	H'E820C004	32
	FIFO control register 2	SSIFCR_2	R/W	H'00000000	H'E820C010	32
	FIFO status register 2	SSIFSR_2	R/(W)*2	H'00010000	H'E820C014	32
	Transmit FIFO data register 2	SSIFTDR_2	W	Undefined	H'E820C018	32
	Receive FIFO data register 2	SSIFRDR_2	R	Undefined	H'E820C01C	32
	TDM mode register 2	SSITDMR_2	R/W	H'00000000	H'E820C020	32
	FC control register 2	SSIFCCR_2	R/W	H'00000000	H'E820C024	32
	FC mode register 2	SSIFCMR_2	R/W	H'00000000	H'E820C028	32
	FC status register 2	SSIFCSR_2	R/(W)*3	H'00000000	H'E820C02C	32
3	Control register 3	SSICR_3	R/W	H'00000000	H'E820C800	32
	Status register 3	SSISR_3	R/W*1	H'02000013	H'E820C804	32
	FIFO control register 3	SSIFCR_3	R/W	H'00000000	H'E820C810	32
	FIFO status register 3	SSIFSR_3	R/(W)*2	H'00010000	H'E820C814	32
	Transmit FIFO data register 3	SSIFTDR_3	W	Undefined	H'E820C818	32
	Receive FIFO data register 3	SSIFRDR_3	R	Undefined	H'E820C81C	32
	TDM mode register 3	SSITDMR_3	R/W	H'00000000	H'E820C820	32
	FC control register 3	SSIFCCR_3	R/W	H'00000000	H'E820C824	32
	FC mode register 3	SSIFCMR_3	R/W	H'00000000	H'E820C828	32
	FC status register 3	SSIFCSR_3	R/(W)*3	H'00000000	H'E820C82C	32

Note 1. Although bits 29 to 26 in these registers can be read from or written to, bits other than these are read-only. For details, refer to section 19.3.2, Status Register (SSISR).

chanhvinh.com

Note 2. To bits 16 and 0 in these registers, only 0 can be written to clear the flags. Other bits are read-only. For details, refer to section 19.3.6, FIFO Status Register (SSIFSR).
Note 3. To bit 24 in these registers, only 0 can be written to clear the flag. Other bits are read-only. For details, refer to section 19.3.12, FC Status Register (SSIFCSR).

chanhvinh.com

19.3.1 Control Register (SSICR)

SSICR is a 32-bit readable/writable register that controls the IRQ, selects the polarity status, and sets operating mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	CKS	TUIEN	TOIEN	RUIEN	ROIEN	IIEN	-	CHNL[1:0]		DWL[2:0]			SWL[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL	CKDV[3:0]				MUEN	-	TEN	REN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved The read value is undefined. The write value should always be 0.
30	CKS	0	R/W	Oversampling Clock Select Selects the clock source for oversampling. 0: AUDIO_X1 input 1: AUDIO_CLK input
29	TUIEN	0	R/W	Transmit Underflow Interrupt Enable 0: Disables an underflow interrupt. 1: Enables an underflow interrupt.
28	TOIEN	0	R/W	Transmit Overflow Interrupt Enable 0: Disables an overflow interrupt. 1: Enables an overflow interrupt.
27	RUIEN	0	R/W	Receive Underflow Interrupt Enable 0: Disables an underflow interrupt. 1: Enables an underflow interrupt.
26	ROIEN	0	R/W	Receive Overflow Interrupt Enable 0: Disables an overflow interrupt. 1: Enables an overflow interrupt.
25	IIEN	0	R/W	Idle Mode Interrupt Enable 0: Disables an idle mode interrupt. 1: Enables an idle mode interrupt.
24	—	0	R	Reserved The read value is undefined. The write value should always be 0.
23, 22	CHNL[1:0]	00	R/W	Channels [When TDM = 0] These bits show the number of channels in each system word. 00: Having one channel per system word 01: Having two channels per system word 10: Having three channels per system word 11: Having four channels per system word [When TDM = 1] These bits show the number of system words in each TDM frame. 00: Setting prohibited 01: Having four system words per TDM frame 10: Having six system words per TDM frame 11: Having eight system words per TDM frame

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
21 to 19	DWL[2:0]	000	R/W	Data Word Length These bits indicate the number of bits in a data word. 000: 8 bits 001: 16 bits 010: 18 bits 011: 20 bits 100: 22 bits 101: 24 bits 110: 32 bits 111: Setting prohibited
18 to 16	SWL[2:0]	000	R/W	System Word Length These bits indicate the number of bits in a system word. 000: 8 bits 001: 16 bits 010: 24 bits 011: 32 bits 100: 48 bits 101: 64 bits 110: 128 bits 111: 256 bits
15	SCKD	0	R/W	Serial Bit Clock Direction 0: Serial bit clock is input, slave mode. 1: Serial bit clock is output, master mode. Note: Only the following settings are allowed: (SCKD, SWSD) = (0, 0) or (1, 1). Other settings are prohibited.
14	SWSD	0	R/W	Serial WS Direction 0: Serial word select is input, slave mode. 1: Serial word select is output, master mode. Note: Only the following settings are allowed: (SCKD, SWSD) = (0,0) or (1,1). Other settings are prohibited.
13	SCKP	0	R/W	Serial Bit Clock Polarity 0: SSIWS and SSIDATA change at the SSISCK falling edge (sampled at the SCK rising edge). 1: SSIWS and SSIDATA change at the SSISCK rising edge (sampled at the SCK falling edge).

	SCKP =0	SCKP = 1
SSIDATA input sampling timing at the time of reception	SSISCK rising edge	SSISCK falling edge
SSIDATA output change timing at the time of transmission	SSISCK falling edge	SSISCK rising edge
SSIWS input sampling timing at the time of slave mode (SWSD = 0)	SSISCK rising edge	SSISCK falling edge
SSIWS output change timing at the time of master mode (SWSD = 1)	SSISCK falling edge	SSISCK rising edge

12	SWSP	0	R/W	Serial WS Polarity [When TDM = 0] 0: SSIWS is low for the 1st channel, high for the 2nd channel. 1: SSIWS is high for the 1st channel, low for the 2nd channel. [When TDM = 1] 0: SSIWS is high only for system word 1 period, low for other periods. 1: Setting prohibited
11	SPDP	0	R/W	Serial Padding Polarity This bit is used to specify the active sense of padding bits. It is also used to specify the logical value to be output from the SSITxD pin when the MUEN bit is set to 0 with transfer disabled in WS continue mode. 0: Padding bits are low. 1: Padding bits are high.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
10	SDTA	0	R/W	Serial Data Alignment 0: Transmitting and receiving in the order of serial data and padding bits 1: Transmitting and receiving in the order of padding bits and serial data
9	PDTA	0	R/W	Parallel Data Alignment When the data word length is 32 bits, this configuration field has no meaning. This bit applies to SSIRDR in receive mode and SSITDR in transmit mode. When data word length is 8 or 16 bits: 0: The lower bits of parallel data (SSITDR, SSIRDR) are transferred prior to the upper bits. 1: The upper bits of parallel data (SSITDR, SSIRDR) are transferred prior to the lower bits. When data word length is 18, 20, 22, or 24 bits: 0: Parallel data (SSITDR, SSIRDR) is left-aligned. 1: Parallel data (SSITDR, SSIRDR) is right-aligned. • PDTA = 0

DWL[2:0]	SSITDR/SSIRDR[31:0]
000	31 24 23 16 15 8 7 0 4th word | 3rd word | 2nd word | 1st word
001	31 16 15 0 2nd word | 1st word
010	31 14 13 0 Valid | Invalid
011	31 12 11 0 Valid | Invalid
100	31 10 9 0 Valid | Invalid
101	31 8 7 0 Valid | Invalid
110	31 0 Valid

• PDTA = 1

DWL[2:0]	SSITDR/SSIRDR[31:0]
000	31 24 23 16 15 8 7 0 1st word | 2nd word | 3rd word | 4th word
001	31 16 15 0 1st word | 2nd word
010	31 18 17 0 Invalid | Valid
011	31 20 19 0 Invalid | Valid
100	31 22 21 0 Invalid | Valid
101	31 24 23 0 Invalid | Valid
110	31 0 Valid

Note: This bit has no meaning in direct transfer between this module and the SCUX. Use the SWAP function of the SCUX.

8	DEL	0	R/W	Serial Data Delay 0: 1 clock cycle delay between SSIWS and SSIDATA 1: No delay between SSIWS and SSIDATA

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	CKDV[3:0]	0000	R/W	Serial Oversampling Clock Division Ratio Sets the ratio between the oversampling clock (AUDIOφ) and the serial bit clock. When the SCKD bit is 0, the setting of these bits is ignored. The serial bit clock is used in the shift register and is supplied from the SSISCK pin. 0000: AUDIOφ 0001: AUDIOφ/2 0010: AUDIOφ/4 0011: AUDIOφ/8 0100: AUDIOφ/16 0101: AUDIOφ/32 0110: AUDIOφ/64 0111: AUDIOφ/128 1000: AUDIOφ/6 1001: AUDIOφ/12 1010: AUDIOφ/24 1011: AUDIOφ/48 1100: AUDIOφ/96 1101: Setting prohibited 1110: Setting prohibited 1111: Setting prohibited
3	MUEN	0	R/W	Mute Enable This bit is used to specify muting of the signals from this module during transmission. It is also used to specify the logical value to be output from the SSITxD pin when the MUEN bit is set to 0 with transfer disabled in WS continue mode. 0: This module is not muted. 1: This module is muted. Note: When this module is muted, serial data to be output is rewritten to 0 but data transmission in the module is not stopped. Write dummy data to the SSIFTDR not to generate a transmit underflow because the number of data in the transmit FIFO is decreasing.
2	—	0	R	Reserved The read value is undefined. The write value should always be 0.
1	TEN	0	R/W	Transmit Enable 0: Disables the transmit operation. 1: Enables the transmit operation. Note: When transmission stop/start is controlled by the SSI control register (SSICTRL_CIM) in the SCUX, set this bit to 0.
0	REN	0	R/W	Receive Enable 0: Disables the receive operation. 1: Enables the receive operation. Note: When reception stop/start is controlled by the SSI control register (SSICTRL_CIM) in the SCUX, set this bit to 0.

chanhvinh.com

19.3.2 Status Register (SSISR)

SSISR consists of status flags indicating the operational status of this module and bits indicating the current channel numbers and word numbers.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	-	-	-	-	-	-	-	-	-
Initial value:	Undefined	Undefined	0	0	0	0	1	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W:	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	TCHNO[1:0]		TSWNO	RCHNO[1:0]		RSWNO	IDST
Initial value:	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	0	0	1	0	0	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: * The bit can be read or written to. Writing 0 initializes the bit, but writing 1 is ignored.

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
29	TUIRQ	0	R/(W)*	Transmit Underflow Error Interrupt Status Flag This status flag indicates that transmit data was supplied at a lower rate than was required. This bit is set to 1 regardless of the value of the TUIEN bit and can be cleared by writing 0 to this bit. If TUIRQ = 1 and TUIEN = 1, an SSI interrupt occurs. If TUIRQ = 1, SSITDR did not have data written to it before it was required for transmission. This will lead to the same data being transmitted once more and a potential corruption of multi-channel data. As a result, this module will output erroneous data. Note: When an underflow error occurs, the current data in the data buffer of this module is transmitted until the next data is written.
28	TOIRQ	0	R/(W)*	Transmit Overflow Error Interrupt Status Flag This status flag indicates that transmit data was supplied at a higher rate than was required. This bit is set to 1 regardless of the value of the TOIEN bit and can be cleared by writing 0 to this bit. If TOIRQ = 1 and TOIEN = 1, an SSI interrupt occurs. If TOIRQ = 1, SSIFTDR had data written to it while the transmit FIFO is full (TDC = H'8). This will lead to the loss of data and a potential corruption of multi-channel data.
27	RUIRQ	0	R/(W)*	Receive Underflow Error Interrupt Status Flag This status flag indicates that receive data was supplied at a lower rate than was required. This bit is set to 1 regardless of the value of the RUIEN bit and can be cleared by writing 0 to this bit. If RUIRQ = 1 and RUIEN = 1, an SSI interrupt occurs. If RUIRQ = 1, SSIFRDR was read while the receive FIFO is empty (RDC = H'0).This can cause invalid receive data to be stored, which may lead to corruption of multi-channel data.
26	ROIRQ	0	R/(W)*	Receive Overflow Error Interrupt Status Flag This status flag indicates that receive data was supplied at a higher rate than was required. This bit is set to 1 regardless of the value of the ROIEN bit and can be cleared by writing 0 to this bit. If ROIRQ = 1 and ROIEN = 1, an SSI interrupt occurs. If ROIRQ = 1, SSIRDR was not read before there was new unread data written to it. This will lead to the loss of data and a potential corruption of multi-channel data. Note: When an overflow error occurs, the current data in the data buffer of this module is overwritten by the next incoming data from the SSI interface.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
25	IIRQ	1	R	Idle Mode Interrupt Status Flag This status flag indicates whether this module is in the idle state. This bit is set to 1 regardless of the value of the IIEN bit to allow polling. The interrupt can be masked by clearing IIEN to 0, but cannot be cleared by writing 0 to this bit. If IIRQ = 1 and IIEN = 1, an SSI interrupt occurs. 0: This module is not in idle state. 1: This module is in idle state.
24 to 7	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
6, 5	TCHNO [1:0]	00	R	Transmit Channel Number These bits show the current channel number. These bits indicate the number of a channel whose data is required to be written to SSITDR. This value will change as the data is copied to the shift register, regardless of whether the data is written to SSITDR. When TDM or CONT is 1, these bits cannot be used.
4	TSWNO	1	R	Transmit Serial Word Number This status bit indicates the current word number. This bit indicates which system word is required to be written to SSITDR. This value will change as the data is copied to the shift register, regardless of whether the data is written to SSITDR. When TDM or CONT is 1, this bit cannot be used.
3, 2	RCHNO [1:0]	00	R	Receive Channel Number These bits show the current channel number. These bits indicate which channel the data in SSIRDR currently represents. This value will change as the data in SSIRDR is updated from the shift register. When TDM or CONT is 1, these bits cannot be used.
1	RSWNO	1	R	Receive Serial Word Number This status bit indicates the current word number. This bit indicates which system word the data in SSIRDR currently represents. This value will change as the data in SSIRDR is updated from the shift register, regardless of whether SSIRDR has been read. When TDM or CONT is 1, this bit cannot be used.
0	IDST	1	R	Idle Mode Status Flag This status flag indicates that the serial bus activity has stopped. This bit is cleared to 0 if the serial bus is currently active while TEN = 1 or REN = 1. This bit is automatically set to 1 if both TEN and REN are cleared to 0 and the current system word communication is completed. Note: If the external device stops the serial bus clock before the current system word is completed, this bit is not set.

Note: * The bit can be read or written to. Writing 0 initializes the bit, but writing 1 is ignored.

chanhvinh.com

19.3.3 Transmit Data Register (SSITDR)

SSITDR is a 32-bit register that stores data to be transmitted. The data for transmission to be stored to SSITDR is automatically transferred from the transmit FIFO data register.

Data written to this register is transferred to the shift register upon transmission request. If the data word length is less than 32 bits, the alignment is determined by the setting of the PDTA control bit in SSICR.

The CPU cannot read or write data from/to SSITDR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

19.3.4 Receive Data Register (SSIRDR)

SSIRDR is a 32-bit register that stores received data. The received data stored in SSIRDR is automatically transferred to the receive FIFO data register.

Data in this register is transferred from the shift register each time data word is received. If the data word length is less than 32 bits, the alignment is determined by the setting of the PDTA control bit in SSICR.

The CPU cannot read or write data from/to SSIRDR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

chanhvinh.com

19.3.5 FIFO Control Register (SSIFCR)

SSIFCR is a 32-bit readable/writable register that specifies the data trigger numbers for the transmit and receive FIFO data registers, and enables or disables FIFO data resets and interrupt requests.
SSIFCR can always be read or written by the CPU.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	TTRG[1:0]		RTRG[1:0]		TIE	RIE	TFRST	RFRST
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7, 6	TTRG[1:0]	00	R/W	Transmit Data Trigger Number When the FIFO is operating for transmission, these bits specify the number of bytes for transmission in the FIFO (trigger number for transmission) at which the TDE flag in the FIFO status register (SSIFSR) will be set. The TDE flag is set to 1 when the number of bytes for transmission in the transmit FIFO data register (SSIFTDR) has fallen to or below the trigger number corresponding to the setting as shown below. 00: 7 (1)* 01: 6 (2)* 10: 4 (4)* 11: 2 (6)* Note: * The values in parentheses are the number of empty stages in SSIFTDR at which the TDE flag is set.
5, 4	RTRG[1:0]	00	R/W	Receive Data Trigger Number When the FIFO is operating for reception, these bits specify the number of received bytes in the FIFO (trigger number for reception) at which the RDF flag in the FIFO status register (SSIFSR) will be set. The RDF flag is set to 1 when the number of received bytes in the receive FIFO data register (SSIFRDR) has risen to or above the trigger number corresponding to the setting as shown below. 00: 1 01: 2 10: 4 11: 6
3	TIE	0	R/W	Transmit Interrupt Enable This bit enables or disables generation of transmit data empty interrupt (TXI) requests in the following situation: when the FIFO is operating for transmission, the data for transmission in the transmit FIFO data register (SSIFTDR) are transferred to the transmit data register (SSITDR) and the number of data bytes in the transmit FIFO data register has become less than the set transmit trigger number, so that the TDE flag in the FIFO status register (SSIFSR) is set to 1. 0: Transmit data empty interrupt (TXI) request is disabled. 1: Transmit data empty interrupt (TXI) request is enabled.* Note: * TXI can be cleared by clearing either the TDE flag (see the description of the TDE bit for details) or TIE bit to 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
2	RIE	0	R/W	Receive Interrupt Enable Enables or disables generation of receive data full interrupt (RXI) requests when the RDF flag in the FIFO status register (SSIFSR) is set to 1 while the FIFO is operating for reception. 0: Receive data full interrupt (RXI) request is disabled. 1: Receive data full interrupt (RXI) request is enabled.* Note: * RXI can be cleared by clearing either the RDF flag (see the description of the RDF bit for details) or RIE bit to 0.
1	TFRST	0	R/W	Transmit FIFO Data Register Reset Invalidates the data in the transmit FIFO data register (SSIFTDR) to reset the FIFO to an empty state. 0: Reset is disabled.* 1: Reset is enabled. Note: * FIFO is reset at a power-on reset.
0	RFRST	0	R/W	Receive FIFO Data Register Reset Invalidates the data in the receive FIFO data register (SSIFRDR) to reset the FIFO to an empty state. 0: Reset is disabled.* 1: Reset is enabled. Note: * FIFO is reset at a power-on reset.

chanhvinh.com

19.3.6 FIFO Status Register (SSIFSR)

SSIFSR contains status flags that indicate the state of operation of the transmit and receive FIFO data registers.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	TDC[3:0]				-	-	-	-	-	-	-	TDE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/(W)*

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	RDC[3:0]				-	-	-	-	-	-	-	RDF
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/(W)*

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27 to 24	TDC[3:0]	0000	R	Number of Data Bytes Stored in SSIFTDR TDC[3:0] = H'0 indicates no data for transmission. TDC[3:0] = H'8 indicates that 32 bytes of data for transmission is stored in SSIFTDR.
23 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	TDE	1	R/(W)*	Transmit Data Empty Indicates that, when the FIFO is operating for transmission, the data for transmission in the transmit FIFO data register (SSIFTDR) is transferred to the transmit data register (SSITDR), the number of data bytes in SSIFTDR has become less than the transmit trigger number specified by TTRG[1:0] in the FIFO control register (SSIFCR), and thus writing of transmit data to SSIFTDR has been enabled. 0: Number of data bytes for transmission in SSIFTDR is greater than the set transmit trigger number. [Clearing conditions] • 0 is written to TDE after data of the number of bytes larger than the set transmit trigger number is written to SSIFTDR. • The direct memory access controller is activated by transmit data empty (TXI) interrupt, and data of the number of bytes larger than the set transmit trigger number is written to SSIFTDR. 1: Number of data bytes for transmission in SSIFTDR is equal to or less than the set transmit trigger number.*1 [Setting conditions] • Power-on reset • Number of transmission data bytes stored in SSIFTDR has become equal to or less than the set transmit trigger number. Note: *1 Since SSIFTDR is an 8-stage FIFO register, the amount of data that can be written to it while TDE = 1 is "8 – transmit trigger number to be specified" bytes at maximum. Writing more data will be ignored. The number of data bytes in SSIFTDR is indicated in the TDC bits in SSIFSR.
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 8	RDC[3:0]	0000	R	Number of Data Bytes Stored in SSIFRDR RDC[3:0] = H'0 indicates no received data. RDC[3:0] = H'8 indicates that 32 bytes of received data is stored in SSIFRDR.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
0	RDF	0	R/(W)*	Receive Data Full Indicates that, when the FIFO is operating for reception, the received data is transferred to the receive FIFO data register (SSIFRDR) and the number of data bytes in SSIFRDR has become greater than the receive trigger number specified by RTRG[1:0] in the FIFO control register (SSIFCR). 0: Number of received data bytes in SSIFRDR is less than the set receive trigger number. [Clearing conditions] • Power-on reset • 0 is written to RDF after the receive FIFO is emptied with writing 1 to RFRST. • 0 is written to RDF after data is read from SSIFRDR until the number of data bytes in SSIFRDR becomes less than the set receive trigger number. • The direct memory access controller is activated by receive data full (RXI) interrupt, and data is read from SSIFRDR until the number of data bytes in SSIFRDR becomes less than the set receive trigger number. 1: Number of received data bytes in SSIFRDR is equal to or greater than the set receive trigger number. [Setting condition] • Data of the number of bytes that is equal to or greater than the set receive trigger number is stored in SSIFRDR.*1 Note: *1 Since SSIFRDR is an 8-stage FIFO register, the amount of data that can be read from it while RDF = 1 is the set receive trigger number of bytes at maximum. Continuing to read data from SSIFRDR after reading all the data will result in undefined data to be read. The number of data bytes in SSIFRDR is indicated in the RDC bits in SSIFSR.

Note: * The bit can be read or written to. Writing 0 initializes the bit, but writing 1 is ignored.

chanhvinh.com

19.3.7 Transmit FIFO Data Register (SSIFTDR)

SSIFTDR is a FIFO register consisting of eight stages of 32-bit registers for storing data to be serially transmitted. On detecting that the transmit data register (SSITDR) is empty, this module transfers the data for transmission written to SSIFTDR to SSITDR to start serial transmission, which can continue until SSIFTDR becomes empty. SSIFTDR can be written to by the CPU at any time.

Note that when SSIFTDR is full of transmit data (32 bytes), the next data cannot be written to it. If writing is attempted, it will be ignored and an overflow occurs.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Note: * Not writable while FIFO is receiving data.

19.3.8 Receive FIFO Data Register (SSIFRDR)

SSIFRDR is a FIFO register consisting of eight stages of 32-bit registers for storing serially received data. When four bytes of data have been received, this module transfers the received data in the receive data register (SSIRDR) to SSIFRDR to complete reception operation. Reception can continue until 32 bytes of data have been stored to SSIFRDR. SSIFRDR can be read but cannot be written to by the CPU. Note that when SSIFRDR is read while it does not hold received data, the value read is undefined and a reception underflow will occur.

After SSIFRDR becomes full of received data, the data received thereafter will be lost and a receive overflow occurs.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

chanhvinh.com

19.3.9 TDM Mode Register (SSITDMR)

SSITDMR is a 32-bit readable/writable register that enables or disables muting of receive data in direct transfer, TDM mode, and WS continue mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RXD MUTE	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	CONT	-	-	-	-	-	-	-	TDM
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17	RXDMUTE	0	R/W	Receive Direct Data Mute Setting When receive data is output directly to the SCUX, the output data is forcibly muted so that there is no signal (output as 0 data). 0: The receive data is output without change. 1: 0 data is output.
16	—	0	R/W	Reserved Always write 0 to this bit.
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	CONT	0	R/W	WS Continue Mode 0: Disables WS continue mode. 1: Enables WS continue mode. Note: This bit can be set only in master mode (SCKD = 1 and SWSD = 1)
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TDM	0	R/W	TDM Mode 0: Disables TDM mode. 1: Enables TDM mode.

chanhvinh.com

19.3.10 FC Control Register (SSIFCCR)

SSIFCCR is a 32-bit readable/writable register that controls frequency change detection.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	FIEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	FCEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	FIEN	0	R/W	Frequency Change Detection Interrupt Enable 0: Disables a frequency change detection interrupt. 1: Enables a frequency change detection interrupt.
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	FCEN	0	R/W	Frequency Change Detection Enable When this bit is set to 1, counting up of cycles of peripheral clock 1 (P1φ) starts at the beginning of the next SSIWS cycle. On the start of each SSIWS cycle, the current counted value is moved to the VALUE bits in the FC Status Register (SSIFCSR). The counter is then cleared to 0 and counting up is resumed. When this bit is set to 0, the counter is cleared to 0 and counting up is stopped. 0: Disables frequency change detection. 1: Enables frequency change detection. Note: Set this bit to 1 after setting the desired values in SSICR, SSIFCMR, and SSITDMR.

chanhvinh.com

19.3.11 FC Mode Register (SSIFCMR)

SSIFCMR sets the maximum and minimum allowable numbers of cycles of the peripheral clock 1 (P1ϕ) for each SSIWS cycle, when frequency change detection is enabled.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	MAXV													
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	MINV													
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
29 to 16	MAXV	0	R/W	Maximum Value Sets the maximum allowable number of cycles of the peripheral clock 1 (P1φ) for each SSIWS cycle, when SSIFCCR.FCEN = 1.
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13 to 0	MINV	0	R/W	Minimum Value Sets the minimum allowable number of cycles of the peripheral clock 1 (P1φ) for each SSIWS cycle, when SSIFCCR.FCEN = 1.

chanhvinh.com

19.3.12 FC Status Register (SSIFCSR)

SSIFCSR consists of the frequency change detection status flag and the bits that indicate the current cycle count of the peripheral clock 1 (P1ϕ).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	FCIRQ	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/(W)*	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	VALUE													
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	FCIRQ	0	R/(W)*	Frequency Change Detection Error Interrupt Status Indicates VALUE > SSIFCMR.MAXV or 0 < VALUE < SSIFCMR.MINV when SSIFCCR.FCEN = 1. This bit is set to 1 regardless of the setting of the FIEN bit in SSIFCCR. Write 0 to clear this flag to 0. When FCIRQ = 1 and SSIFCCR.FIEN = 1, an SSI interrupt is generated.
23 to 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13 to 0	VALUE	0	R	VALUE Indicates the current cycle count of the peripheral clock 1 (P1φ) in an SSIWS cycle, when SSIFCCR.FCEN = 1. Each time the next SSIWS cycle starts, the value is updated. Note: When the SSISCK signal stops, the start of an SSIWS cycle cannot be detected. Consequently, the value is not updated.

Note: * The bit can be read or written to. Writing 0 initializes the bit, but writing 1 is ignored.

chanhvinh.com

19.4 Operation Description

19.4.1 Bus Format

This module can operate as a transmitter or a receiver and can be configured into many serial bus formats in either mode. The bus format can be selected from one of the twelve major modes shown in Table 19.3.

Table 19.3 Bus Format for SSIF Module

	TEN	REN	SCKD	SWSD	TDM	MUEN	IIEN	TOIEN	TUIEN	ROIEN	RUIEN	CONT	SWSP	DEL	PDTA	SDTA	SPDP	SCKP	SWL[2:0]	DWL[2:0]	CHNL[1:0]
Non-Compression Slave Receiver	0	1	0	0	0	Control Bits							Configuration Bits								
Non-Compression Slave Transmitter	1	0	0	0	0																
Non-Compression Slave Transceiver*	1	1	0	0	0																
Non-Compression Master Receiver	0	1	1	1	0																
Non-Compression Master Transmitter	1	0	1	1	0																
Non-Compression Master Transceiver*	1	1	1	1	0																
TDM Slave Receiver	0	1	0	0	1								0	Configuration Bits							
TDM Slave Transmitter	1	0	0	0	1								0								
TDM Slave Transceiver*	1	1	0	0	1								0								
TDM Master Receiver	0	1	1	1	1								0								
TDM Master Transmitter	1	0	1	1	1								0								
TDM Master Transceiver*	1	1	1	1	1								0								

Note: * Set the TEN and REN bits to 1 at the same time when using transceiver mode.

chanhvinh.com

19.4.2 Non-Compressed Modes

The non-compressed modes support all serial audio streams split into channels. It supports the I^2S compatible format as well as many more variants on these modes.

(1) Slave Receiver

This mode allows the module to receive serial data from another device. The clock and word select signal used for the serial data stream is also supplied from an external device. If these signals do not conform to the format specified in the configuration fields of this module, operation is not guaranteed.

(2) Slave Transmitter

This mode allows the module to transmit serial data to another device. The clock and word select signal used for the serial data stream is also supplied from an external device. If these signals do not conform to the format specified in the configuration fields of this module, operation is not guaranteed.

(3) Slave Transceiver

This mode allows serial data transmission and reception between this module and another device. The clock and word select signal used for the serial data stream is also supplied from an external device. If these signals do not conform to the format specified in the configuration fields of this module, operation is not guaranteed.

(4) Master Receiver

This mode allows the module to receive serial data from another device. The clock and word select signals are internally derived from the oversampling clock. The format of these signals is defined in the configuration fields of this module. If the incoming data from another device does not follow the configured format, operation is not guaranteed.

(5) Master Transmitter

This mode allows the module to transmit serial data to another device. The clock and word select signals are internally derived from the oversampling clock. The format of these signals is defined in the configuration fields of this module.

(6) Master Transceiver

This mode allows serial data transmission and reception between this module and another device. The clock and word select signals are internally derived from the oversampling clock. The format of these signals is defined in the configuration fields of this module.

(7) Operation Setting Related to Word Length

All bits related to the SSICR's word length are valid in non-compressed modes. There are many configurations this module supports, but some of the combinations are shown below for the I^2S compatible format, MSB-first and left-aligned format, and MSB-first and right-aligned format.

chanhvinh.com

- I²S Compatible Format

Figure 19.2 and Figure 19.3 show the I²S compatible formats without and with padding, respectively. Padding occurs when the data word length is smaller than the system word length.

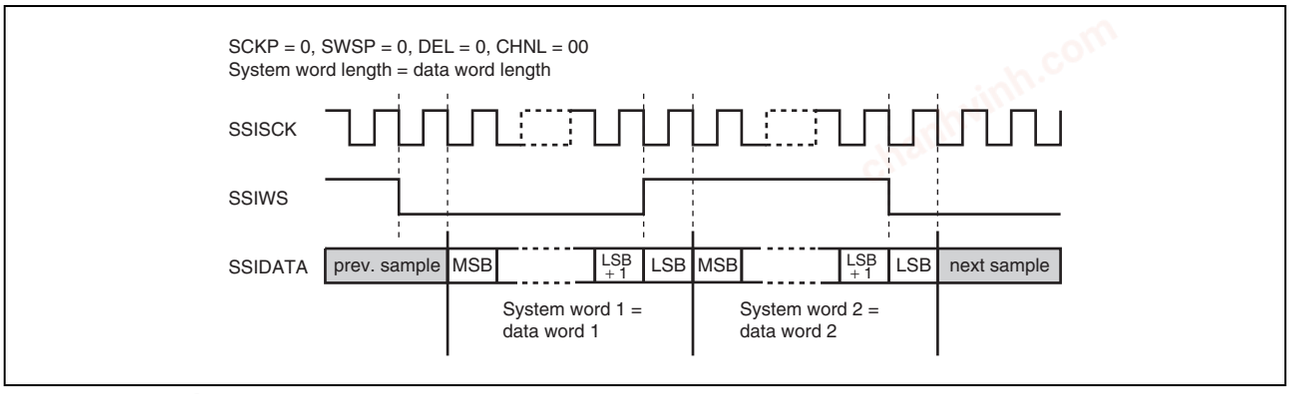


Figure 19.2 I²S Compatible Format (without Padding)

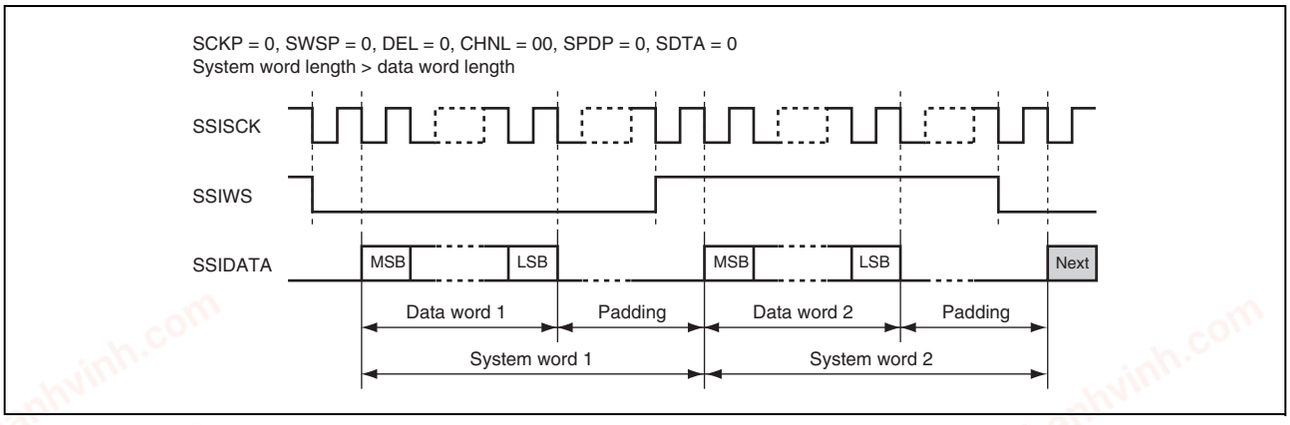


Figure 19.3 I²S Compatible Format (with Padding)

Figure 19.4 shows the MSB-first and left-aligned format and Figure 19.5 shows the MSB-first and right-aligned format.

- MSB-first and Left-aligned Format

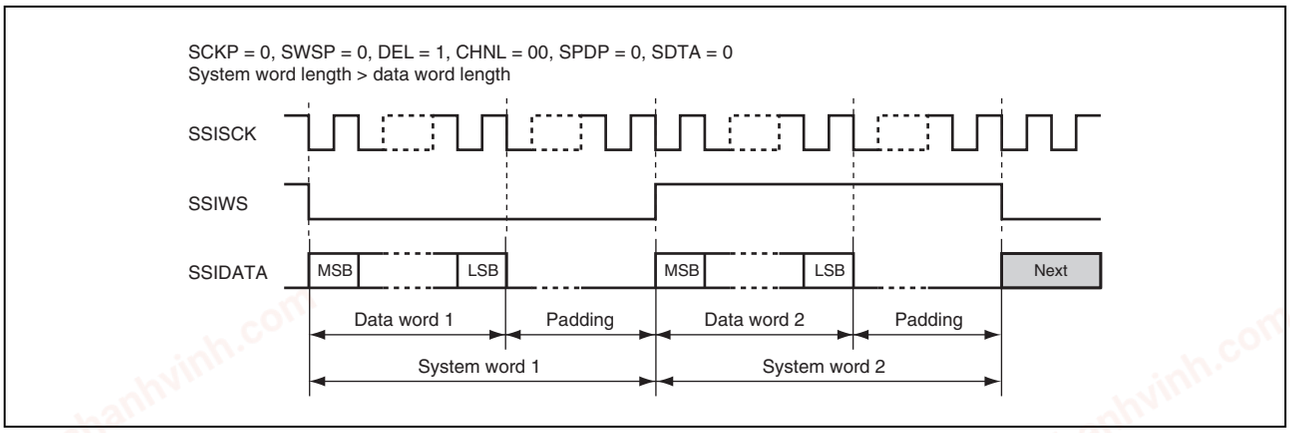


Figure 19.4 MSB-first and Left-aligned Format (Transmitted and Received in the Order of Serial Data and Padding Bits)

chanhvinh.com

- MSB-first and Right-aligned Format

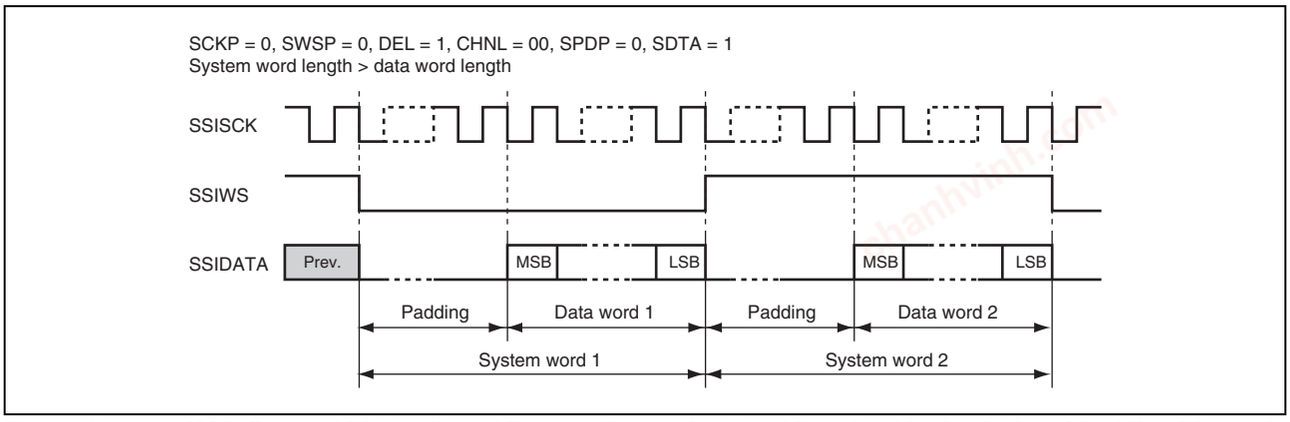


Figure 19.5 MSB-first and Right-aligned Format (Transmitted and Received in the Order of Padding Bits and Serial Data)

chanhvinh.com

(8) Multi-channel Formats

Some devices extend the definition of the I²S bus specification and allow more than 2 channels to be transferred within two system words.

This module supports the transfer of 4, 6, and 8 channels by using the CHNL, SWL and DWL bits only when the system word length (SWL) is greater than or equal to the data word length (DWL) multiplied by channels (CHNL).

Table 19.4 shows the number of padding bits for each of the valid setting. If setting is not valid, "—" is indicated instead of a number.

Table 19.4 The Number of Padding Bits for Each Valid Setting

Padding Bits per System Word		DWL[2:0]	000	001	010	011	100	101	110
CHNL [1:0]	Decoded Channels per System Word	SWL [2:0]	Decoded Word Length	8	16	18	20	22	24	32
00	1	000	8	0	—	—	—	—	—	—
		001	16	8	0	—	—	—	—	—
		010	24	16	8	6	4	2	0	—
		011	32	24	16	14	12	10	8	0
		100	48	40	32	30	28	26	24	16
		101	64	56	48	46	44	42	40	32
		110	128	120	112	110	108	106	104	96
		111	256	248	240	238	236	234	232	224
01	2	000	8	—	—	—	—	—	—	—
		001	16	0	—	—	—	—	—	—
		010	24	8	—	—	—	—	—	—
		011	32	16	0	—	—	—	—	—
		100	48	32	16	12	8	4	0	—
		101	64	48	32	28	24	20	16	0
		110	128	112	96	92	88	84	80	64
		111	256	240	224	220	216	212	208	192
10	3	000	8	—	—	—	—	—	—	—
		001	16	—	—	—	—	—	—	—
		010	24	0	—	—	—	—	—	—
		011	32	8	—	—	—	—	—	—
		100	48	24	0	—	—	—	—	—
		101	64	40	16	10	4	—	—	—
		110	128	104	80	74	68	62	56	32
		111	256	232	208	202	196	190	184	160
11	4	000	8	—	—	—	—	—	—	—
		001	16	—	—	—	—	—	—	—
		010	24	—	—	—	—	—	—	—
		011	32	0	—	—	—	—	—	—
		100	48	16	—	—	—	—	—	—
		101	64	32	0	—	—	—	—	—
		110	128	96	64	56	48	40	32	0
		111	256	224	192	184	176	168	160	128

chanhvinh.com

When this module acts as a transmitter, each word written to SSITDR is transmitted to the serial audio bus in the order they are written. When this module acts as a receiver, each word received by the serial audio bus is read from the SSIRDR register in the order they are received.

Figure 19.6 to Figure 19.8 show how the data on 4, 6, and 8 channels are transferred to the serial audio bus. Note that there are no padding bits in the first example (Figure 19.6), the second example (Figure 19.7) is left-aligned and the third (Figure 19.8) is right-aligned. The other conditions in these examples have been selected arbitrarily.

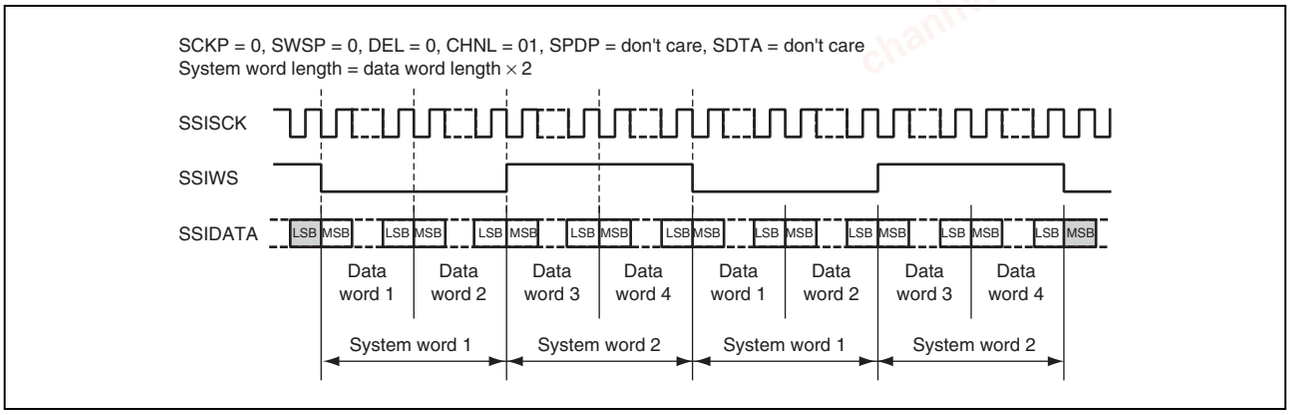


Figure 19.6 Multi-Channel Format (4 Channels Without Padding)

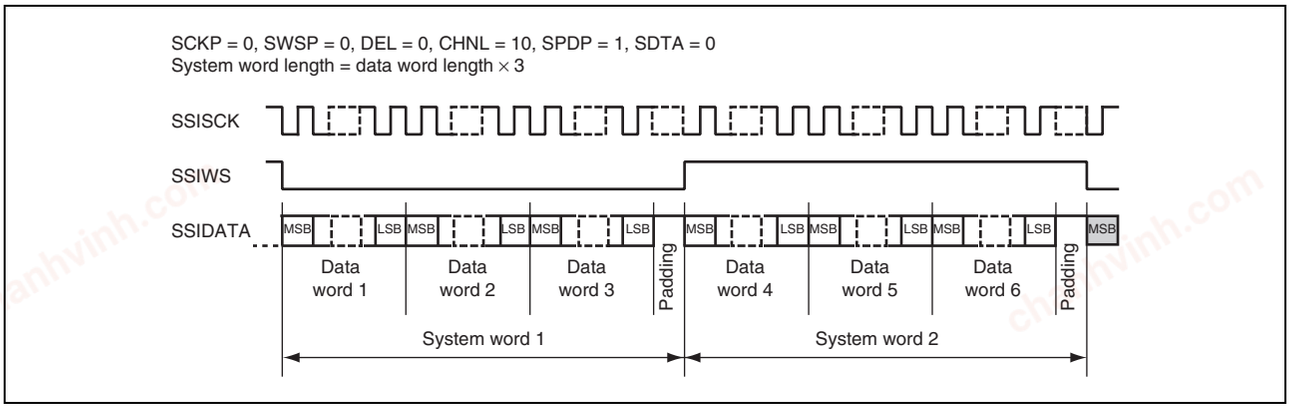


Figure 19.7 Multi-Channel Format (6 Channels with High Padding)

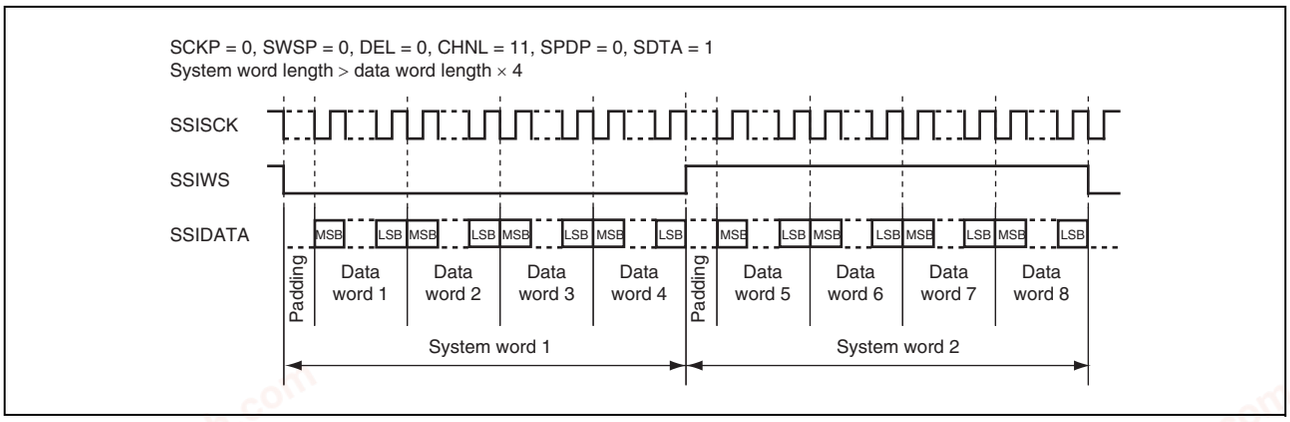


Figure 19.8 Multi-Channel Format (8 Channels; Transmitting and Receiving in the Order of Padding Bits and Serial Data ; with Padding)

chanhvinh.com

(9) Operation Format Configuration Bits

Several more configuration bits in non-compressed mode are shown below. These bits are not mutually exclusive, but some combinations may not be useful.

These configuration bits are described below with reference to the basic sample format in Figure 19.9.

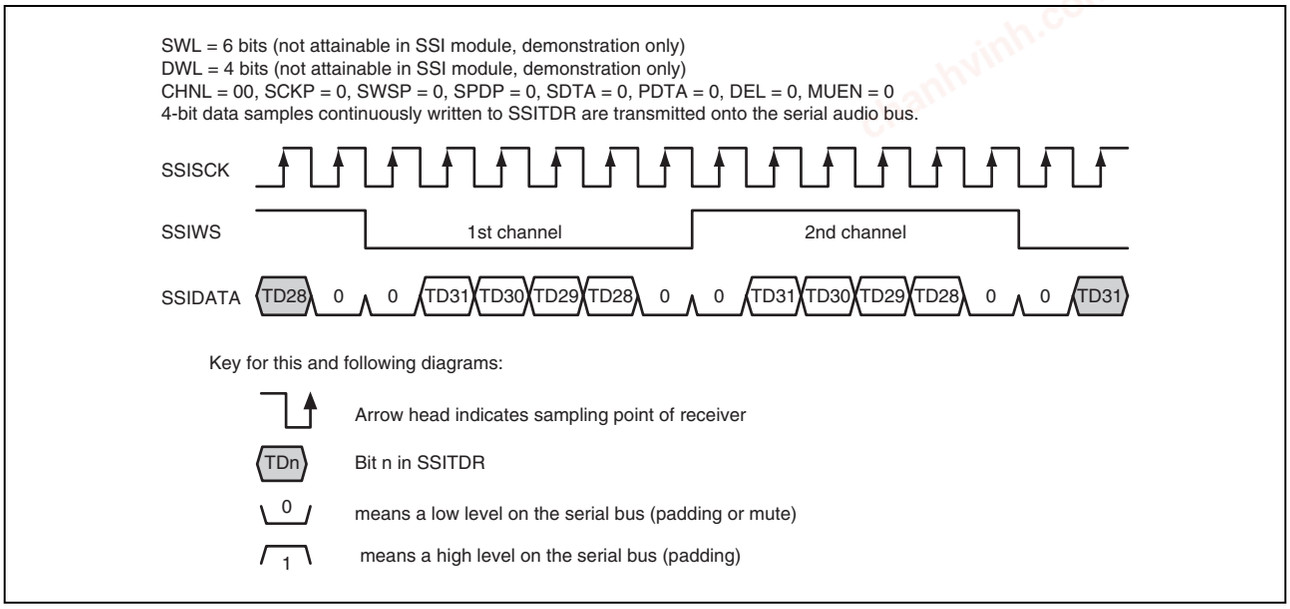


Figure 19.9 Basic Sample Format (Transmit Mode with Example System/Data Word Length)

Figure 19.9 uses a system word length of 6 bits and a data word length of 4 bits. These settings are not possible with this module but are used as an example only for clarification of the other configuration bits.

- Inverted Clock

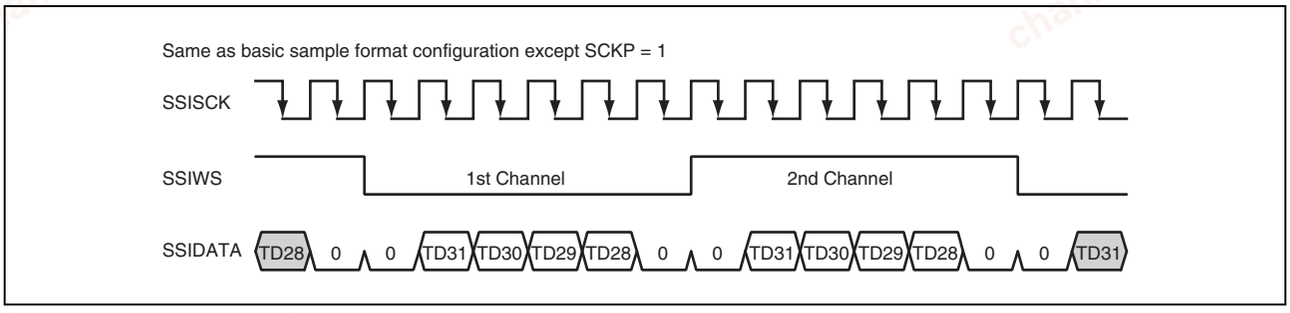


Figure 19.10 Inverted Clock

- Inverted Word Select

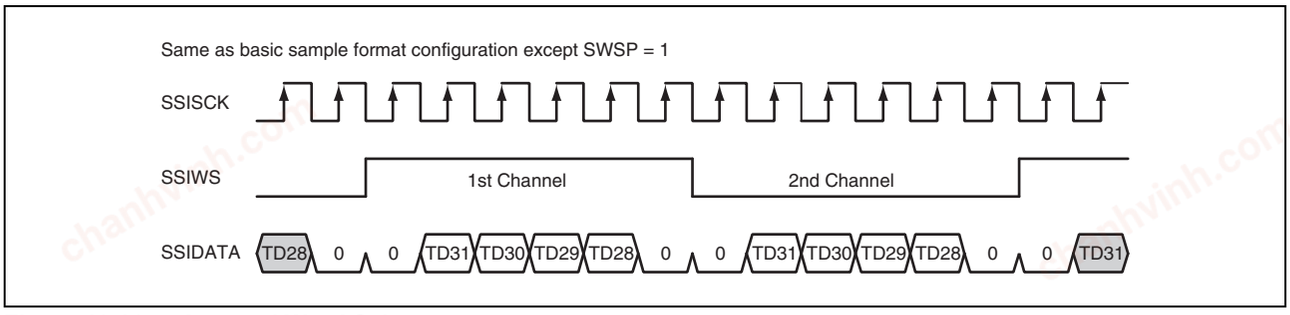


Figure 19.11 Inverted Word Select

chanhvinh.com

- Inverted Padding Polarity

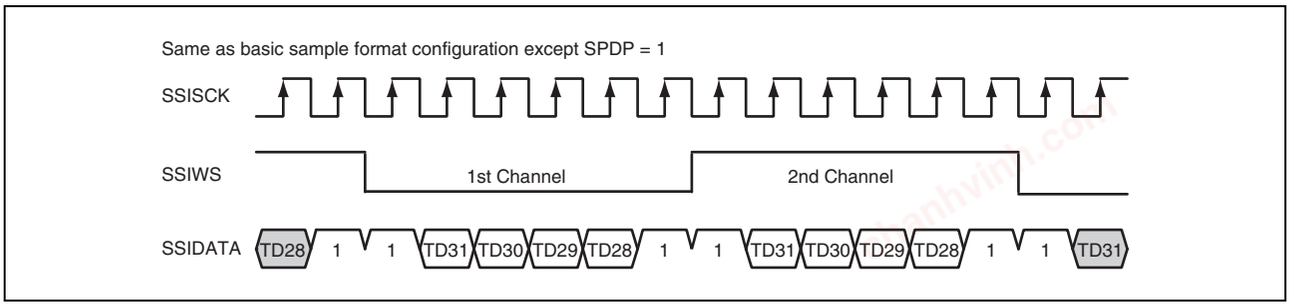


Figure 19.12 Inverted Padding Polarity

- Transmitting and Receiving in the Order of Padding Bits and Serial Data; with Delay

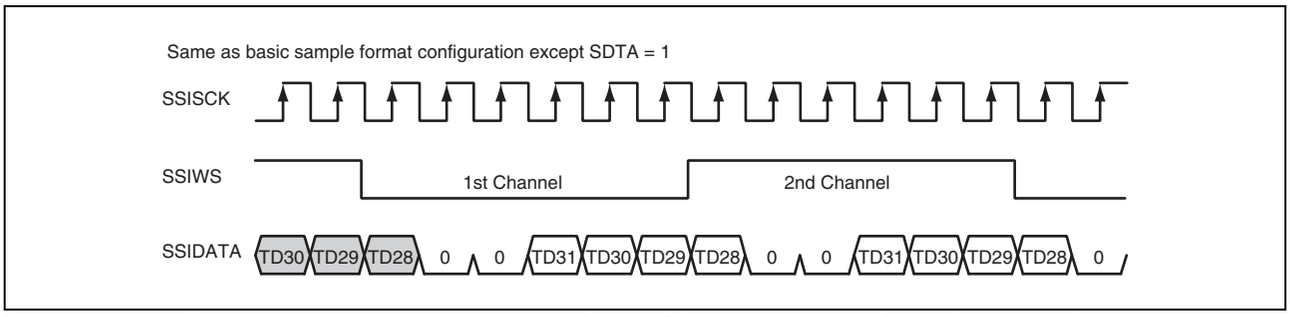


Figure 19.13 Transmitting and Receiving in the Order of Padding Bits and Serial Data; with Delay

- Transmitting and Receiving in the Order of Padding Bits and Serial Data; without Delay

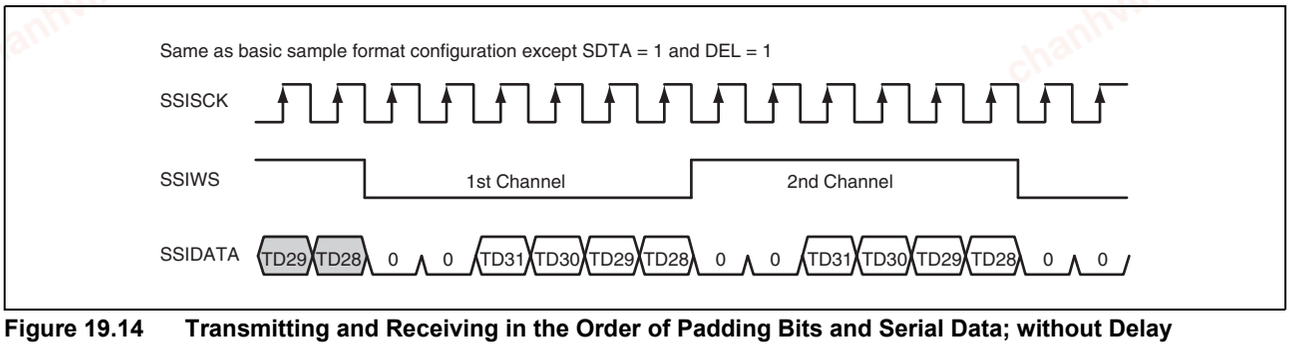


Figure 19.14 Transmitting and Receiving in the Order of Padding Bits and Serial Data; without Delay

- Transmitting and Receiving in the Order of Serial Data and Padding Bits; without Delay

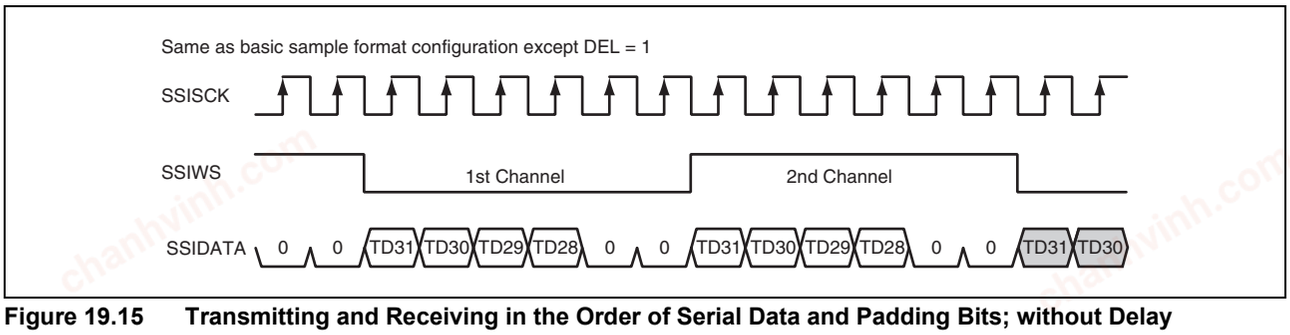


Figure 19.15 Transmitting and Receiving in the Order of Serial Data and Padding Bits; without Delay

chanhvinh.com

- Parallel Right-Aligned with Delay

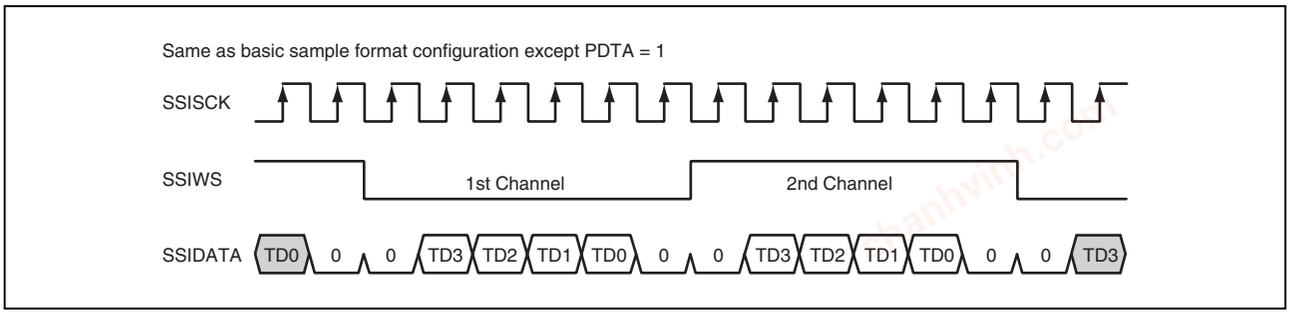


Figure 19.16 Parallel Right-Aligned with Delay

- Mute Enabled

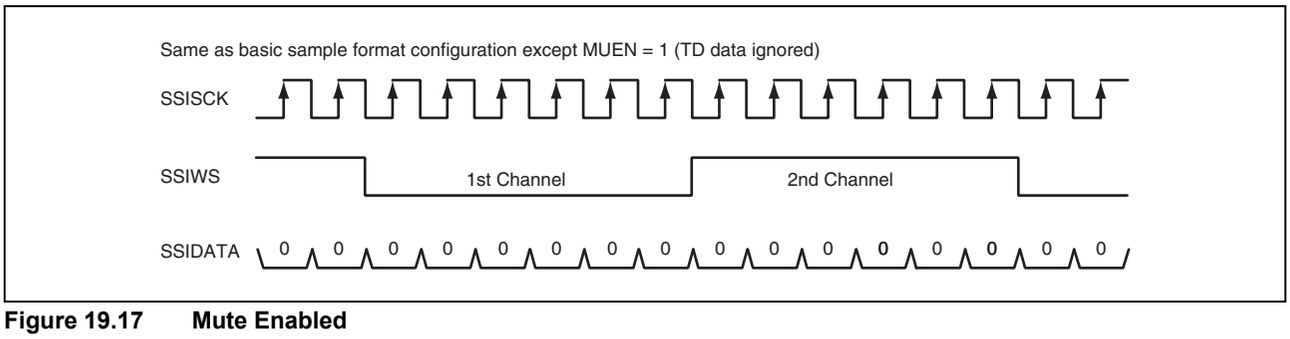


Figure 19.17 Mute Enabled

chanhvinh.com

19.4.3 TDM Mode

TDM mode is provided to enable connection to multi-channel devices for TDM. This mode can be set using the TDM bit in the TDM mode register (SSITDMR). In this mode, the SSIWS signal is high only for system word 1 period and low for the other periods. The pulse produced on the SSIWS signal is defined as SYNC pulse. Note that the SYNC pulse always has the positive polarity (high only for system word 1 period).
Figure 19.18 and Figure 19.19 show the TDM formats without and with padding, respectively.

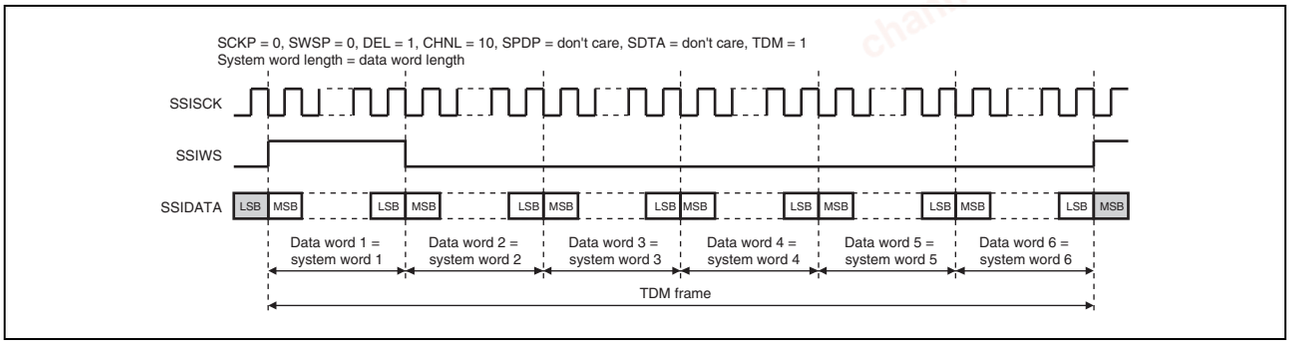


Figure 19.18 TDM Format (6 system words, no padding)

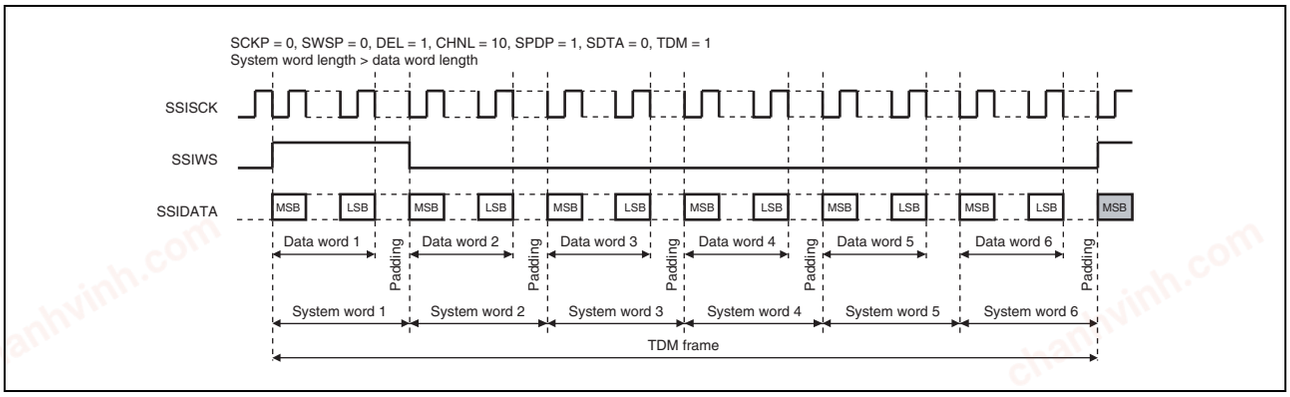


Figure 19.19 TDM Format (6 system words, with padding)

chanhvinh.com

19.4.4 WS Continue Mode

In WS continue mode, the SSIWS signal continues to be output irrespective whether data transfer is enabled or disabled. This mode can be set using the CONT bit in the TDM mode register (SSITDMR). With this mode enabled, the SSIWS signal does not stop but continues operating even if TEN and REN bits in the control register (SSICR) are both set to 0 (transfer disabled). While transfer is disabled, the SSITxD pin outputs 0 if the MUEN bit in SSICR is set to 0, and outputs the value specified by the SPDP bit in SSICR if the MUEN bit is set to 0. With this mode disabled, the SSIWS signal stops if TEN and REN bits are both set to 0.

Figure 19.20 and Figure 19.21 show the operations with WS continue mode enabled and disabled, respectively.

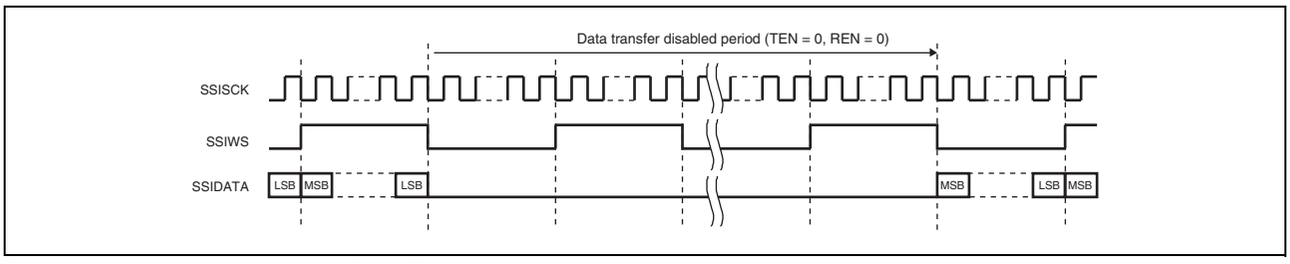


Figure 19.20 WS Continue Mode Enabled

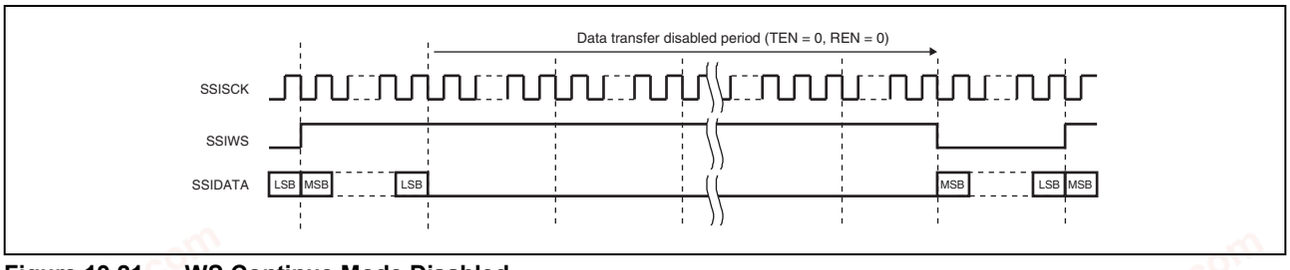


Figure 19.21 WS Continue Mode Disabled

19.4.5 Operation Modes

There are three modes of operation: configuration, enabled and disabled. Figure 19.22 shows how the module enters each of these modes.

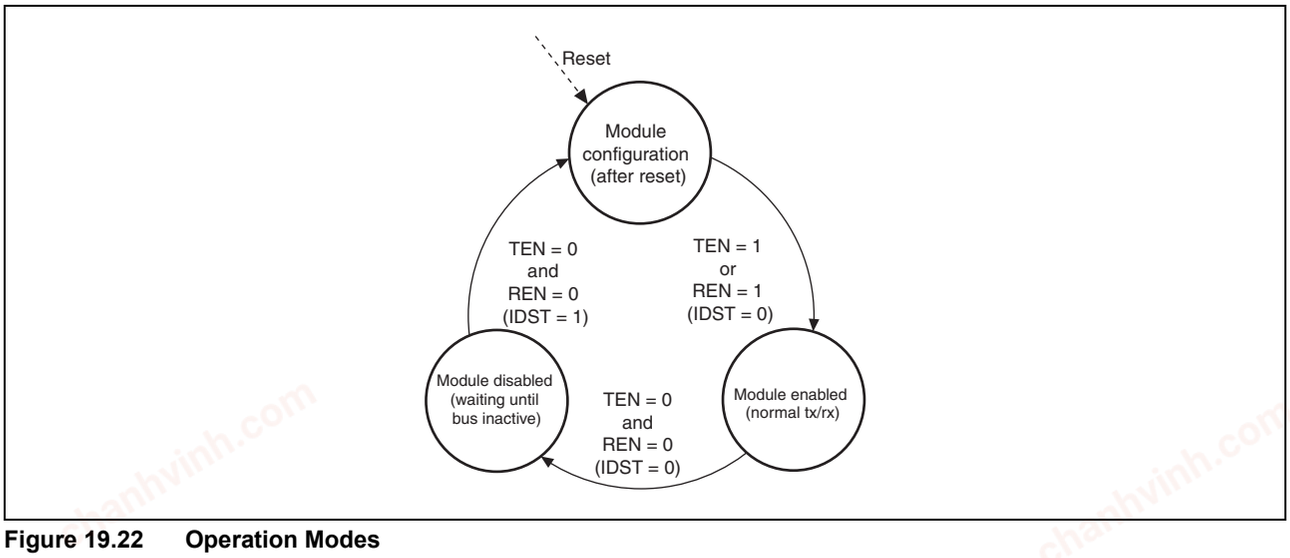


Figure 19.22 Operation Modes

chanhvinh.com

(1) Configuration Mode

This mode is entered after the module is released from reset. All required configuration fields in the control register should be defined in this mode, before this module is enabled by setting the TEN and REN bits.
Setting the TEN and REN bits causes the module to enter the module enabled mode.

(2) Module Enabled Mode

Operation of the module in this mode is dependent on the operation mode selected. For details, refer to section 19.4.6, Transmit Operation, and section 19.4.7, Receive Operation, below.

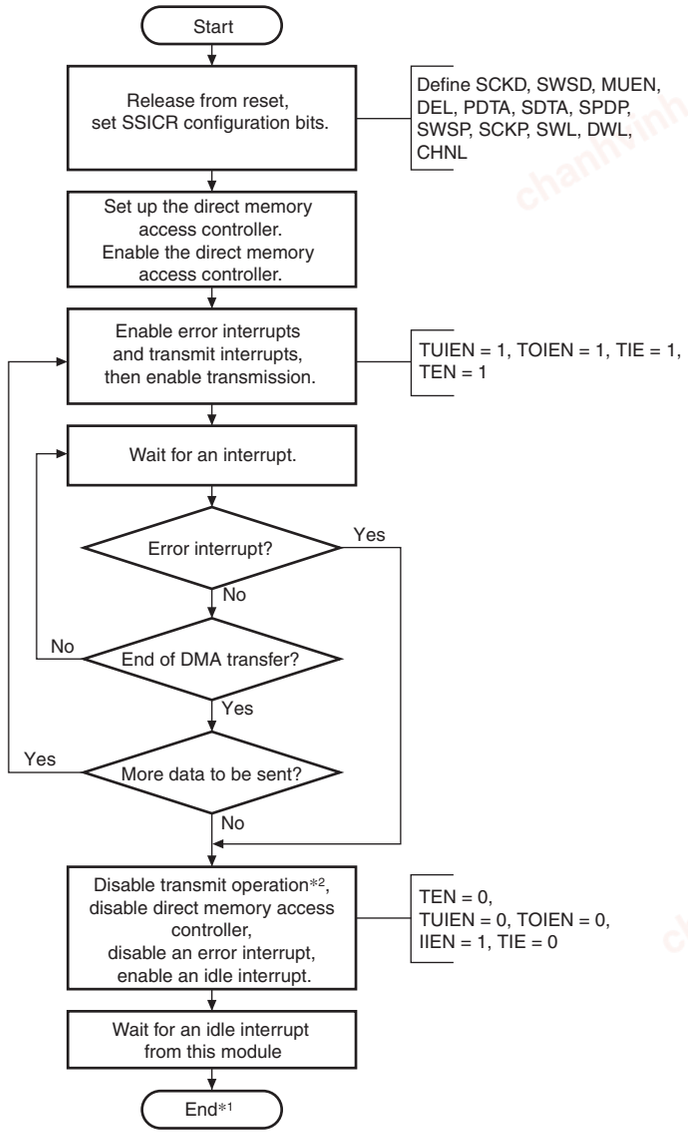
19.4.6 Transmit Operation

Transmission can be controlled either by DMA transfer or interrupt.
DMA control is preferred to reduce the processor load. In DMA control mode, the processor will only receive interrupts if there is an underflow or overflow of data or if the DMA transfer has been completed.
The alternative method is using the interrupts that this module generates to supply data as required.
When disabling this module, the clock* must be kept supplied to this module until the IIRQ bit indicates that the module is in the idle state.
Figure 19.23 shows the transmit operation in DMA control mode, and Figure 19.24 shows the transmit operation in interrupt control mode.

Note: * Input clock from the SSISCK pin when SCKD = 0.
Oversampling clock when SCKD = 1.

chanhvinh.com

(1) Transmission Using Direct Memory Access Controller

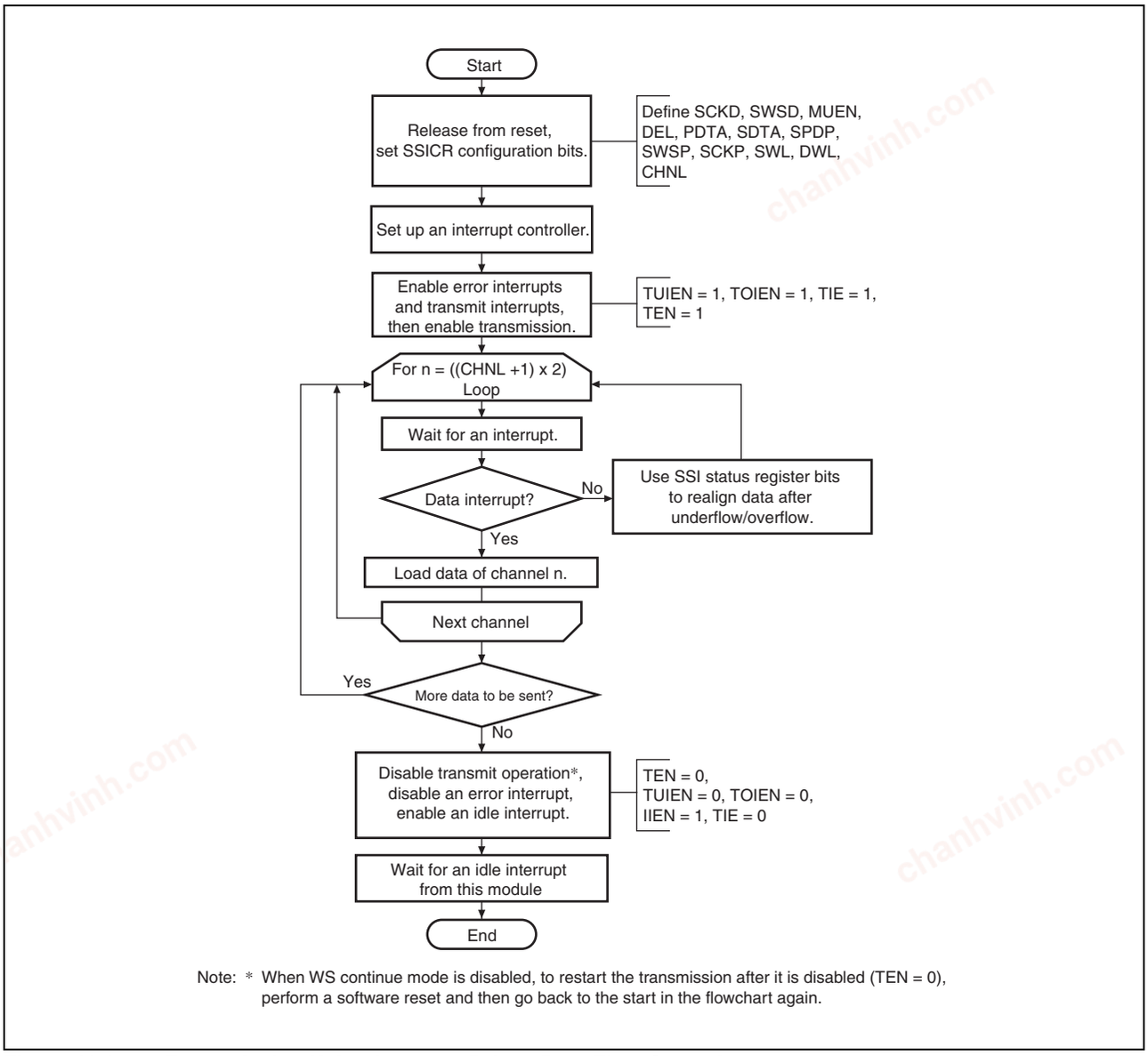


Notes: 1. If an error interrupt (underflow/overflow) occurs, go back to the start in the flowchart again.
2. When WS continue mode is disabled, to restart the transmission after it is disabled (TEN = 0), perform a software reset and then go back to the start in the flowchart again.

Figure 19.23 Transmission Using Direct Memory Access Controller

chanhvinh.com

(2) Transmission Using Interrupt-Driven Data Flow Control



Note: * When WS continue mode is disabled, to restart the transmission after it is disabled (TEN = 0), perform a software reset and then go back to the start in the flowchart again.

Figure 19.24 Transmission Using Interrupt-Driven Data Flow Control

chanhvinh.com

19.4.7 Receive Operation

Like transmission, reception can be controlled either by DMA transfer or interrupt.

Figure 19.25 and Figure 19.26 show the flow of operation.

When disabling this module, the clock* must be kept supplied to this module until the IIRQ bit indicates that the module is in the idle state.

Note: * Input clock from the SSISCK pin when SCKD = 0.
Oversampling clock when SCKD = 1.

(1) Reception Using Direct Memory Access Controller

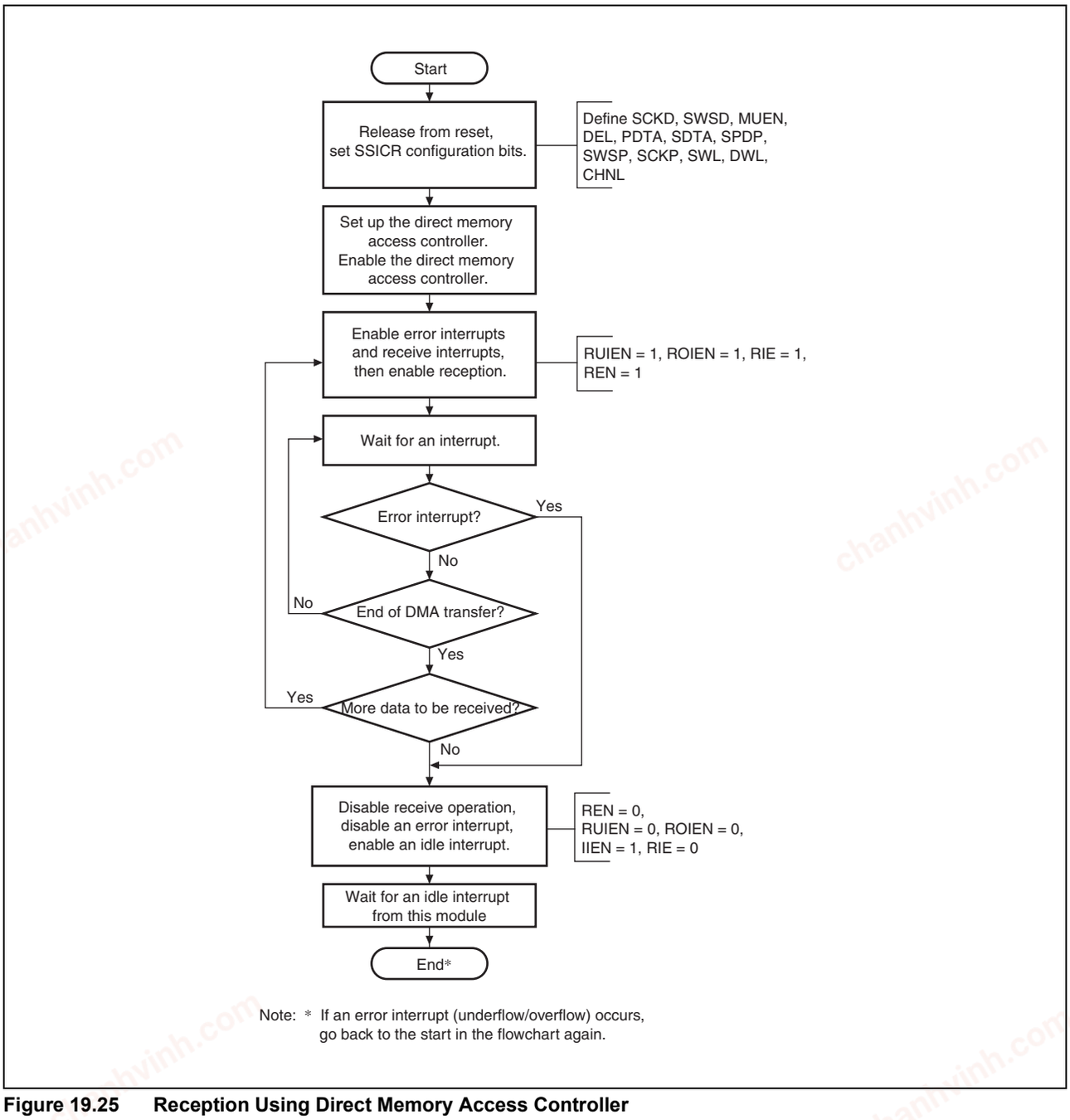


Figure 19.25 Reception Using Direct Memory Access Controller

chanhvinh.com

(2) Reception Using Interrupt-Driven Data Flow Control

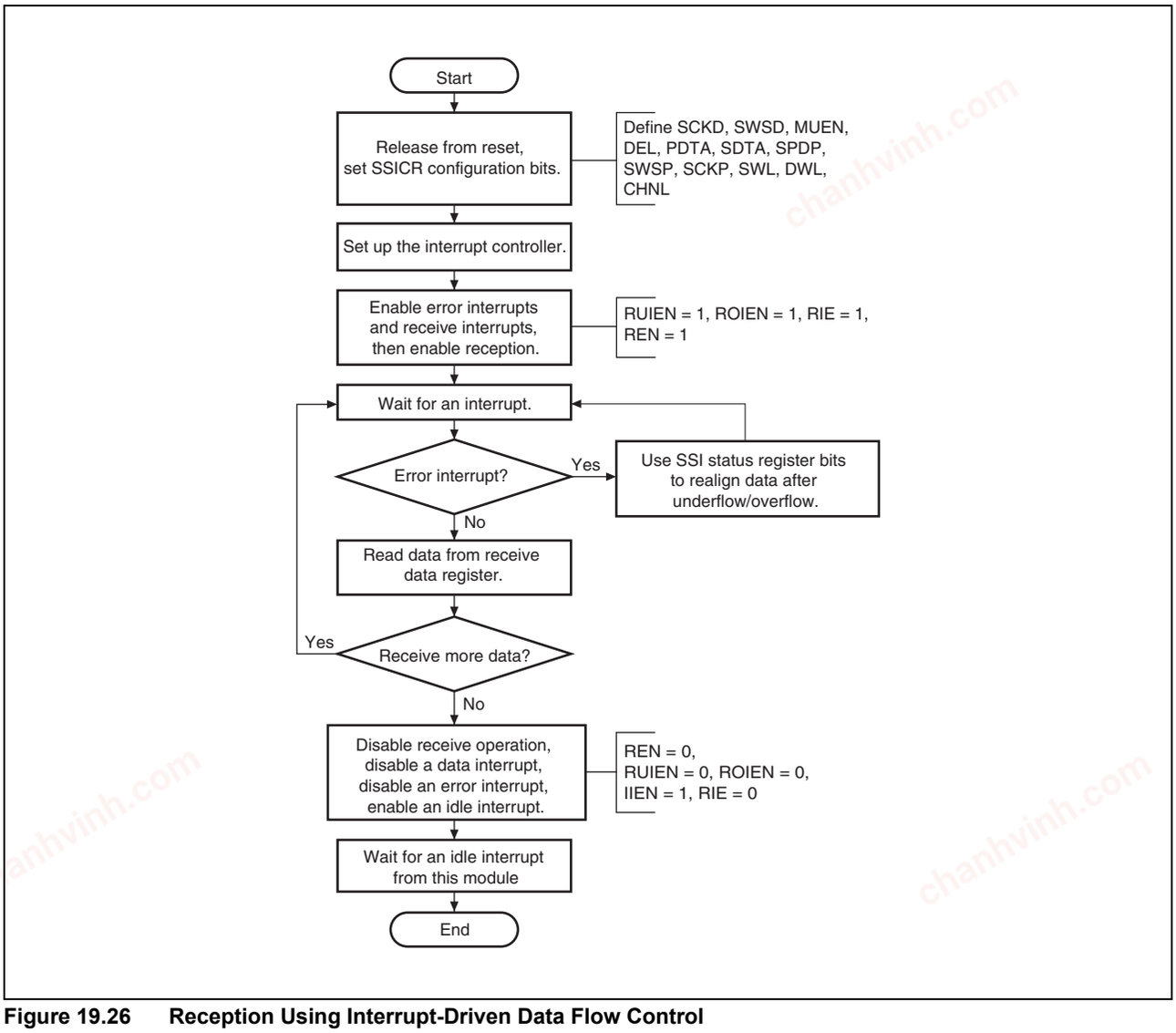


Figure 19.26 Reception Using Interrupt-Driven Data Flow Control

When an underflow or overflow error condition has matched, this module can be recovered to the status before underflow or overflow condition match by using the TCHNO [1:0] and TSWNO bits in transmission and the RCHNO[1:0] and RSWNO bits in reception. When an underflow or overflow occurs, the host CPU can read the channel number and system word number to determine what point the serial audio stream has reached. In the transmitter case, the host CPU can skip forward through the data it wants to transmit until it finds the sample data that matches what this module is expecting to transmit next, and so resynchronize with the audio data stream. In the receiver case the host CPU can store null data to make the number of receive data items consistent until it is ready to store the sample data that this module is indicating will be received next, and so resynchronize with the audio data stream.

chanhvinh.com

19.4.8 Serial Bit Clock Control

This function is used to control and select the clock that is used for the serial bus interface.
If the serial bit clock direction is set to input (SCKD = 0), this module is in clock slave mode and the shift register uses the bit clock that was input to the SSISCK pin.
If the serial bit clock direction is set to output (SCKD = 1), this module is in clock master mode, and the shift register uses the oversampling clock or a divided oversampling clock as the bit clock. The oversampling clock is divided by the ratio specified by the serial oversampling clock division ratio bits (CKDV) in SSICR for use as the bit clock by the shift register.
In either case above, the output of the SSISCK pin is the same as the bit clock.

chanhvinh.com

19.5 Usage Notes

19.5.1 Limitations from Underflow or Overflow during DMA Operation

If an underflow or overflow occurs while the DMA is in operation, the module should be restarted. The transmit and receive buffers in the SSIF consists of 32-bit registers that share the L and R channels. Therefore, data to be transmitted and received at the L channel may sometimes be transmitted and received at the R channel if an underflow or overflow occurs, for example, under the following condition: the control register (SSICR) has a 32-bit setting for both data word length (DWL2 to DWL0) and system word length (SWL2 to SWL0).

If an error interrupt (transmit underflow, transmit overflow, receive underflow, and receive overflow) or setting of a corresponding error status flag (the bits TUIRQ, TOIRQ, RUIRQ, and ROIRQ in SSISR) indicates an error, write 0 to the TEN or REN bit in SSICR to disable DMA transfer requests from this module, thus stopping the operation. Make the setting to stop the direct memory access controller. After this, if reception had been in progress, write 0 to the error status flag bit to clear it, set the direct memory access controller again, and restart the transfer. For transmission, issue a software reset and execute the procedure to start again.

19.5.2 Note on Changing Mode from Master Transceiver to Master Receiver

If a transmit underflow occurs in master transceiver mode while WS continue mode is disabled (SSITDMR.CONT = 0) and the TEN bit in SSICR is set to 0 in order to disable transmit operation, SSIWS output is broken. In order to receive seamlessly after changing mode to master receiver mode, write dummy data to SSIFTDR to suppress transmit underflow.

19.5.3 Limits on TDM mode and WS Continue Mode

If TDM mode or WS continue mode setting is changed, the operation of the SSISCK and SSIWS signals immediately after switching are not guaranteed. If it affects the device to be connected, do not change the setting dynamically.

To temporarily halt and restart transmission while the WS continue mode is enabled (SSITDMR.CONT = 1), after writing to the transmit FIFO data register (SSIFTDR) a multiple of two times, use the transmit underflow error interrupt or the corresponding error status flag (SSISR.TUIRQ) to confirm that an error has occurred, and then write 0 to the TEN bit of the SSISCR register.

Note that after the transmit underflow error, the last value written to SSIFTDR will be repeatedly sent as long as SSISCR.TEN = 1. Therefore, write a dummy value as the last data for transmission or mute the signal by writing 1 to the MUEN bit of the SSISCR register.

To restart transmission, do not apply a software reset; after writing 0 to the error status flag bit to clear it, use the idle mode status flag (SSISR.IDST) to confirm that this module is in the idle state, and then write 1 to the TEN bit of the SSISCR register.

chanhvinh.com

20. Media Local Bus

This module is only incorporated in the RZ/A1L.
This product supports the media local bus as an interface to connect to the MOST controller.

20.1 Features

- 3-pin interface
- A maximum of 50 Mbps of data can be transferred.

For details on the functions and registers, contact Renesas Electronics Corporation's sales office.

Figure 20.1 shows the block diagram.

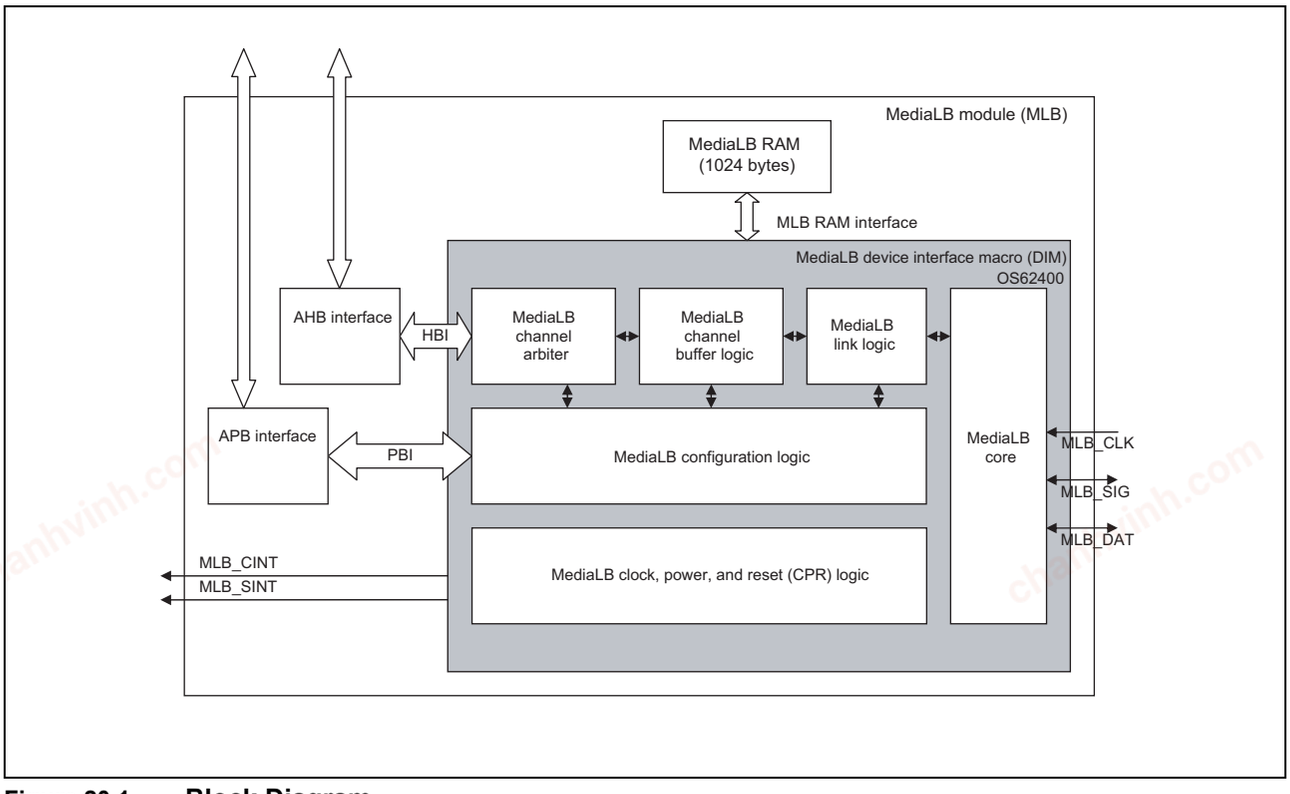


Figure 20.1 Block Diagram

chanhvinh.com

20.2 Input/Output Pins

Table 20.1 shows the pin configuration.

Table 20.1 Pin Configuration

Pin Name	I/O	Description
MLB_CLK	I	MediaLB clock input
MLB_SIG	I/O	MediaLB signal information I/O
MLB_DAT	I/O	MediaLB data I/O

20.3 Register Description

Table 20.2 shows the register configuration.

For details on the registers, contact Renesas Electronics Corporation's sales office.

Table 20.2 Register Configuration

	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Configuration control register	Device control configuration register	DCCR	*1	*1	H'E8034000	32
	System status configuration register	SSCR	*1	*1	H'E8034004	32
	System data configuration register	SDCR	*1	*1	H'E8034008	32
	System mask configuration register	SMCR	*1	*1	H'E803400C	32
	Version control configuration register	VCCR	*1	*1	H'E803401C	32
	Synchronous base address configuration register	SBCR	*1	*1	H'E8034020	32
	Asynchronous base address configuration register	ABCR	*1	*1	H'E8034024	32
	Control base address configuration register	CBCR	*1	*1	H'E8034028	32
	Isochronous base address configuration register	IBCR	*1	*1	H'E803402C	32
	Channel interrupt configuration register	CICR	*1	*1	H'E8034030	32
Channel m*2 configuration register	Channel m entry configuration register	CECRm	*1	*1	H'E8034040 + m x 10	32
	Channel m status configuration register	CSCRm	*1	*1	H'E8034044 + m x 10	32
	Channel m current buffer configuration register	CCBCRm	*1	*1	H'E8034048 + m x 10	32
	Channel m next buffer configuration register	CNBCRm	*1	*1	H'E803404C + m x 10	32
	Local channel m buffer configuration register	LCBCRm	*1	*1	H'E8034280 + m x 4	32

Note 1. Contact Renesas Electronics Corporation's sales office.
Note 2. Each channel of the media local bus is identified by “m” (m = 0 to 30).

chanhvinh.com

21. CAN Interface

This section gives an overall description of the CAN interface (RS-CAN).

The first section describes the features specific to this LSI, including the number of units and the register base addresses. The subsequent sections describe the RS-CAN's functions and registers.

21.1 Overview

21.1.1 Units

This microcontroller incorporates the following number of units of the CAN interface (RS-CAN).

Table 21.1 Units of RS-CAN

	RS-CAN
Number of units	1
Name	RSCAN0

The RS-CAN has two channels.

Table 21.2 Channels of RS-CAN

	RS-CAN
Number of channels	2
Name	CAN0, CAN1

Table 21.3 Index

Index	Meaning	Available Number of CAN Channels and Index
		2 channels
n	Throughout this section, the unit of the RS-CAN is identified by the index "n": for example, RSCANnGCFG is the global configuration register in the RS-CANn unit.	n = 0
m	Throughout this section, the individual channels in the RS-CAN units are identified by the index "m": for example, RSCAN0CmSTS is the channel m status register in the RS-CAN0 unit.	m = 0 or 1
j	The individual receive rule table registers in the RS-CAN units are identified by the index "j": for example, RSCAN0GAFLIDj is the receive rule ID register j in the RS-CAN0 unit.	j = 0 to 15
k	The individual transmit/receive FIFO in the RS-CAN units are identified by the index "k" (k = 0 to channel m × 3 + 2): for example, RSCAN0CFCCk is the transmit/receive FIFO buffer configuration and control register k in the RS-CAN0 unit.	k = 0 to 5
q	The individual receive buffer in the RS-CAN units are identified by the index "q" (q = 0 to channel m × 16 + 15): for example, RSCAN0RMIDq is the receive buffer ID register q in the RS-CAN0 unit.	q = 0 to 31
p	The individual transmit buffers in the RS-CAN units are identified by the index "p" (p = 0 to channel m × 16 + 15): for example, RSCAN0TMCp is the transmit buffer control register p in the RS-CAN0 unit.	p = 0 to 31
y	When the registers other than above are collectively explained, they are identified by the index "y" (y = 0): for example, RSCAN0RMNDy is the receive buffer new data register in the RS-CAN0 unit.	y = 0

chanhvinh.com

21.1.2 Register addresses

RS-CAN base addresses are listed in the following table.

RS-CAN register addresses are given as offsets from the base addresses.

Table 21.4 Register base address

Base Address Name	Base Address
<RSCAN0_base>	E803 $A000_H$

21.1.3 Clock supply

The RS-CAN provides two clock inputs:

Table 21.5 RS-CAN clock supply

Module	RS-CAN clock Clock	Connected to
RSCAN0	clk_xincan	CAN_CLK
	clkc	P1ϕ/2
	pclk	P1ϕ

The operating frequency of the RS-CAN depends on the transfer rate and the number of channels in use. Table 21.6 shows the range of the frequency.

Table 21.6 Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in this LSI

Condition		Range of Operating Frequency		
Transfer Rate	No. of Channels in Use	pclk	clk_xincan*1	clkc*1, *2
1 Mbps	2ch	pclk ≥ 26 MHz	8 MHz ≤ clk_xincan ≤ pclk/2	12.5 MHz ≤ clkc ≤ pclk/2
	1ch	pclk ≥ 18 MHz		
500 kbps	2ch	pclk ≥ 13 MHz	4 MHz ≤ clk_xincan ≤ pclk/2	12.5 MHz ≤ clkc ≤ pclk/2
	1ch	pclk ≥ 8 MHz		
125 kbps	2ch	pclk ≥ 8 MHz	4 MHz ≤ clk_xincan ≤ pclk/2	12.5 MHz ≤ clkc ≤ pclk/2
	1ch			

Note 1. Setting the DCS bit in RSCAN0GCFG enables to select either clk_xincan or clkc. Set clocks less than or equal to pclk/2.

Note 2. Select clk_xincan when pclk < 25 MHz.

chanhvinh.com

21.1.4 Interrupts

The Controller Area Network (RS-CAN) can generate the interrupt requests shown in the following table.

Table 21.7 RS-CAN interrupt requests

Unit Interrupt Name	Outline	Interrupt ID	DMA Trigger Number
	RSCAN0		
INTRCANGERR	CAN global error interrupt	253	—
INTRCANGRECC	CAN receive FIFO interrupt	254	—
	CAN0		
INTRCANmERR (m = 0)	CAN0 error interrupt	256	—
INTRCANmREC (m = 0)	CAN0 transmit/receive FIFO receive completion interrupt	255	—
INTRCANmTRX (m = 0)	CAN0 transmit interrupt	257	—
	CAN1		
INTRCANmERR (m = 1)	CAN1 error interrupt	259	—
INTRCANmREC (m = 1)	CAN1 transmit/receive FIFO receive completion interrupt	258	—
INTRCANmTRX (m = 1)	CAN1 transmit interrupt	260	—

21.1.5 I/O signals

Table 21.8 lists the I/O pins of the RS-CAN module.

Table 21.8 I/O Pins of the RS-CAN Module

Unit Signal Name	Outline	Alternative port pin signal
	CAN0	
CANmRX (m = 0)	CAN0 receive data input	CAN0RX
CANmTX (m = 0)	CAN0 transmit data output	CAN0TX
	CAN1	
CANmRX (m = 1)	CAN1 receive data input	CAN1RX
CANmTX (m = 1)	CAN1 transmit data output	CAN1TX

chanhvinh.com

21.2 Function

This LSI incorporates one unit of the CAN interface (RS-CAN) which consists of two channels (CAN0 and CAN1) of the CAN controller conforming to the ISO11898-1 specifications. Table 21.9 shows the RSCAN module specifications.

Figure 21.1 shows the RS-CAN module block diagram.

Table 21.9 RS-CAN Module Specifications (1/2)

Item	Specification
Number of channels	2
Protocol	ISO11898-1 compliant
Communication speed	• Maximum 1 Mbps $\text{Communication speed (CANm bit time clock)} = \frac{1}{\text{CANm bit time}}$ $\text{CANm bit time} = \text{CANmTq} \times \text{Tq count per bit}$ $\text{CANmTq} = \frac{\text{(BRP[9:0] bits in the RSCAN0CmCFG register + 1)}}{\text{fCAN}}$ m = 0 or 1 Tq: Time quantum fCAN: Frequency of CAN clock (selected by the DCS bit in the RSCAN0GCFG register)
Buffer	160 buffers in total • Individual buffers: 32 buffers (16 buffers × 2 channels) Transmit buffer: 16 buffers per channel Transmit queue: Single queue per channel • Shared buffers: 128 buffers for all channels Receive buffer: 0 to 31 buffers Receive FIFO buffer: 8 FIFO buffers (up to 96 buffers allocatable to each) Transmit/receive FIFO buffer: 3 FIFO buffers per channel (up to 96 buffers allocatable to each)
Reception function	• Receives data frames and remote frames. • Selects ID format (standard ID, extended ID, or both IDs) to be received. • Sets interrupt enable/disable for each FIFO. • Mirror function (CAN mode receives its own transmitted messages.) • Timestamp function (to record message reception time as a 16-bit timer value)
Reception filter function	• Selects receive messages according to 128 receive rules. • Sets the number of receive rules (0 to 128) for each channel. • Acceptance filter processing: Sets ID and mask for each receive rule. • DLC filter processing: Sets DLC check value for each acceptance rule.
Receive message transfer function	• Routing function to transfer receive messages to arbitrary destinations (can be transferred to up to 8 buffers) Transfer destination: Receive buffer, receive FIFO buffer, and/or transmit/receive FIFO buffer • Label addition function Stores label information together with a message in a receive buffer and FIFO buffer.
Transmission function	• Transmits data frames and remote frames. • Selects ID format (standard ID, extended ID, or both IDs) to be transmitted. • Sets interrupt enable/disable for each transmit buffer and transmit/receive FIFO buffer. • Selects ID priority transmission or transmit buffer number priority transmission. • Transmit request can be aborted (possible to confirm with a flag) • One-shot transmission function
Interval transmission function	Transmit messages at intervals (transmit mode or gateway mode of transmit/receive FIFO buffers)
Transmit queue function	Transmits all stored messages according to the ID priority.

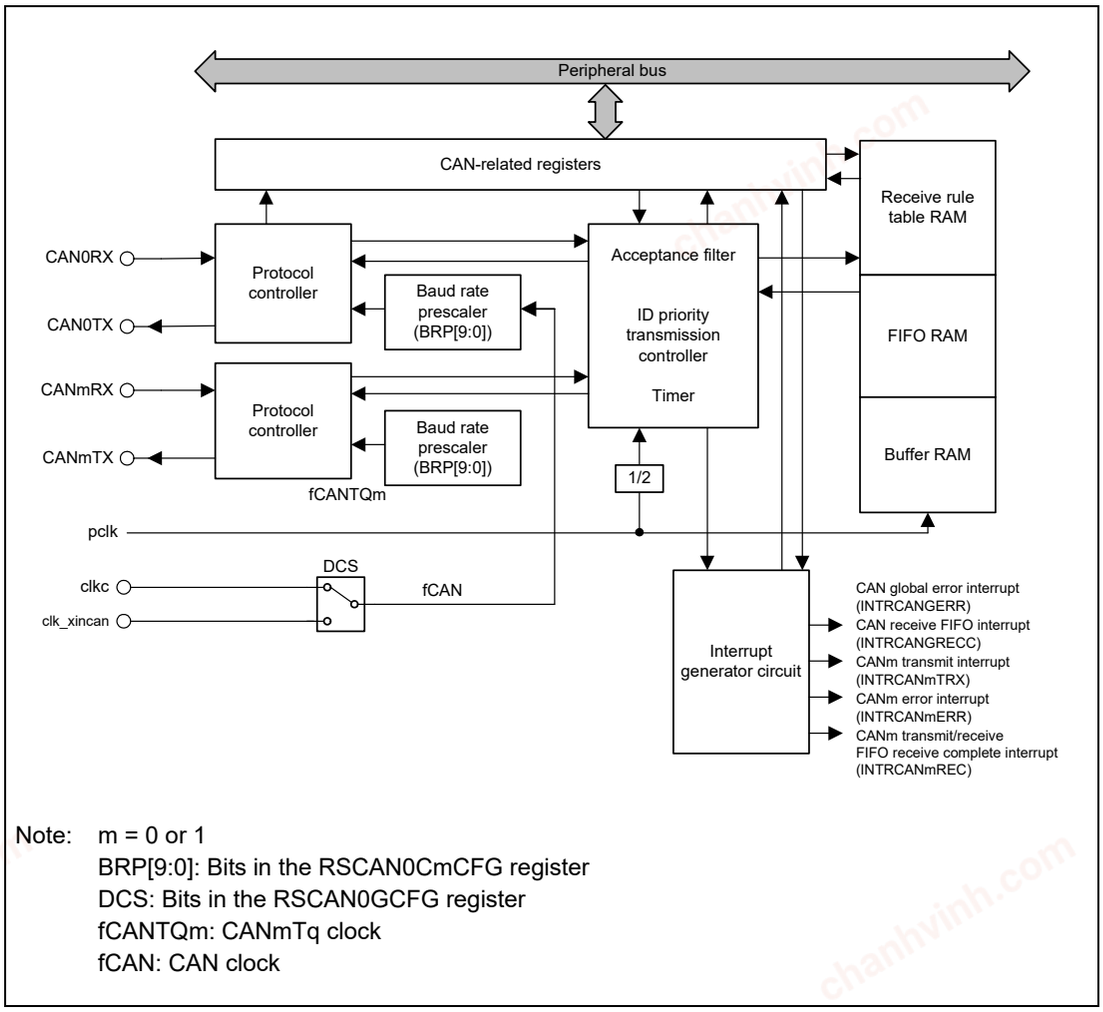
chanhvinh.com

Table 21.9 RS-CAN Module Specifications (2/2)

Item	Specification
Transmit history function	Stores the history information of transmitted messages.
Gateway function	A received message is automatically routed to a different channel.
Bus off recovery mode selection	Selects the method for returning from bus off state. • ISO11898-1 compliant • Automatic entry to channel halt mode at bus-off entry • Automatic entry to channel halt mode at bus-off end • Transition to channel standby mode by program request • Transition to the error-active state by program request
Error status monitoring	• Monitors CAN protocol errors (stuff error, form error, ACK error, CRC error, bit error, ACK delimiter error, and bus dominant lock). • Detects error status transitions (error warning, error passive, bus off entry, and bus off recovery) • Reads the error counter. • Monitors DLC errors.
Interrupt source	8 sources • Global Interrupts [2 sources: common among channels] Receive FIFO interrupt [1 source: common among channels] Global error interrupt [1 source: common among channels] • Channel interrupts [6 sources: 3 sources × number of channels] CANm transmit interrupt [1 source for each channel] – CANm transmit complete interrupt – CANm transmit abort interrupt – CANm transmit/receive FIFO transmit complete interrupt (in transmit mode, gateway mode) – CANm transmit history interrupt – CANm transmit queue interrupt CANm transmit/receive FIFO receive complete interrupt (in receive mode, gateway mode) [1 source for each channel] CANm error interrupt [1 source for each channel] (m = 0 or 1)
CAN stop mode	Reduces power consumption by stopping clock supply to the RS-CAN module.
CAN clock source	Selects the clkc or the clk_xincan. As for the range of operating frequency, refer to Table 21.6.
Test function	Test function for user evaluation • Listen-only mode • Self-test mode 0 (external loopback) • Self-test mode 1 (internal loopback) • Inter-channel communication test

chanhvinh.com

21.2.1 Block Diagram



Note: m = 0 or 1
BRP[9:0]: Bits in the RSCAN0CmCFG register
DCS: Bits in the RSCAN0GCFG register
fCANTQm: CANmTq clock
fCAN: CAN clock

Figure 21.1 RS-CAN Module Block Diagram

chanhvinh.com

21.3 Registers

Table 21.10 lists the registers of the RS-CAN module.n = 0, 1.

Table 21.10 List of RS-CAN Module Registers (1/12)

Register Name	Symbol	After Reset	Address	Access Size
Channel 0 configuration register	RSCAN0C0CFG	0000 0000$_H$	<RSCAN0_base> + 0000$_H$	8, 16, 32
Channel 0 control register	RSCAN0C0CTR	0000 0005$_H$	<RSCAN0_base> + 0004$_H$	8, 16, 32
Channel 0 status register	RSCAN0C0STS	0000 0005$_H$	<RSCAN0_base> + 0008$_H$	8, 16, 32
Channel 0 error flag register	RSCAN0C0ERFL	0000 0000$_H$	<RSCAN0_base> + 000C$_H$	8, 16, 32
Channel 1 configuration register	RSCAN0C1CFG	0000 0000$_H$	<RSCAN0_base> + 0010$_H$	8, 16, 32
Channel 1 control register	RSCAN0C1CTR	0000 0005$_H$	<RSCAN0_base> + 0014$_H$	8, 16, 32
Channel 1 status register	RSCAN0C1STS	0000 0005$_H$	<RSCAN0_base> + 0018$_H$	8, 16, 32
Channel 1 error flag register	RSCAN0C1ERFL	0000 0000$_H$	<RSCAN0_base> + 001C$_H$	8, 16, 32
Global configuration register	RSCAN0GCFG	0000 0000$_H$	<RSCAN0_base> + 0084$_H$	8, 16, 32
Global control register	RSCAN0GCTR	0000 0005$_H$	<RSCAN0_base> + 0088$_H$	8, 16, 32
Global status register	RSCAN0GSTS	0000 000D$_H$	<RSCAN0_base> + 008C$_H$	8, 16, 32
Global error flag register	RSCAN0GERFL	0000 0000$_H$	<RSCAN0_base> + 0090$_H$	8, 16, 32
Global timestamp counter register	RSCAN0GTSC	0000 0000$_H$	<RSCAN0_base> + 0094$_H$	16, 32
Receive rule entry control register	RSCAN0GAFLECTR	0000 0000$_H$	<RSCAN0_base> + 0098$_H$	8, 16, 32
Receive rule configuration register 0	RSCAN0GAFLCFG0	0000 0000$_H$	<RSCAN0_base> + 009C$_H$	8, 16, 32
Receive buffer number register	RSCAN0RMNB	0000 0000$_H$	<RSCAN0_base> + 00A4$_H$	8, 16, 32
Receive buffer new data register 0	RSCAN0RMND0	0000 0000$_H$	<RSCAN0_base> + 00A8$_H$	8, 16, 32
Receive FIFO buffer configuration and control register 0	RSCAN0RFCC0	0000 0000$_H$	<RSCAN0_base> + 00B8$_H$	8, 16, 32
Receive FIFO buffer configuration and control register 1	RSCAN0RFCC1	0000 0000$_H$	<RSCAN0_base> + 00BC$_H$	8, 16, 32
Receive FIFO buffer configuration and control register 2	RSCAN0RFCC2	0000 0000$_H$	<RSCAN0_base> + 00C0$_H$	8, 16, 32
Receive FIFO buffer configuration and control register 3	RSCAN0RFCC3	0000 0000$_H$	<RSCAN0_base> + 00C4$_H$	8, 16, 32
Receive FIFO buffer configuration and control register 4	RSCAN0RFCC4	0000 0000$_H$	<RSCAN0_base> + 00C8$_H$	8, 16, 32
Receive FIFO buffer configuration and control register 5	RSCAN0RFCC5	0000 0000$_H$	<RSCAN0_base> + 00CC$_H$	8, 16, 32
Receive FIFO buffer configuration and control register 6	RSCAN0RFCC6	0000 0000$_H$	<RSCAN0_base> + 00D0$_H$	8, 16, 32
Receive FIFO buffer configuration and control register 7	RSCAN0RFCC7	0000 0000$_H$	<RSCAN0_base> + 00D4$_H$	8, 16, 32
Receive FIFO buffer status register 0	RSCAN0RFSTS0	0000 0001$_H$	<RSCAN0_base> + 00D8$_H$	8, 16, 32
Receive FIFO buffer status register 1	RSCAN0RFSTS1	0000 0001$_H$	<RSCAN0_base> + 00DC$_H$	8, 16, 32
Receive FIFO buffer status register 2	RSCAN0RFSTS2	0000 0001$_H$	<RSCAN0_base> + 00E0$_H$	8, 16, 32
Receive FIFO buffer status register 3	RSCAN0RFSTS3	0000 0001$_H$	<RSCAN0_base> + 00E4$_H$	8, 16, 32
Receive FIFO buffer status register 4	RSCAN0RFSTS4	0000 0001$_H$	<RSCAN0_base> + 00E8$_H$	8, 16, 32
Receive FIFO buffer status register 5	RSCAN0RFSTS5	0000 0001$_H$	<RSCAN0_base> + 00EC$_H$	8, 16, 32
Receive FIFO buffer status register 6	RSCAN0RFSTS6	0000 0001$_H$	<RSCAN0_base> + 00F0$_H$	8, 16, 32
Receive FIFO buffer status register 7	RSCAN0RFSTS7	0000 0001$_H$	<RSCAN0_base> + 00F4$_H$	8, 16, 32
Receive FIFO buffer pointer control register 0	RSCAN0RFPCTR0	—	<RSCAN0_base> + 00F8$_H$	8, 16, 32
Receive FIFO buffer pointer control register 1	RSCAN0RFPCTR1	—	<RSCAN0_base> + 00FC$_H$	8, 16, 32
Receive FIFO buffer pointer control register 2	RSCAN0RFPCTR2	—	<RSCAN0_base> + 0100$_H$	8, 16, 32
Receive FIFO buffer pointer control register 3	RSCAN0RFPCTR3	—	<RSCAN0_base> + 0104$_H$	8, 16, 32
Receive FIFO buffer pointer control register 4	RSCAN0RFPCTR4	—	<RSCAN0_base> + 0108$_H$	8, 16, 32
Receive FIFO buffer pointer control register 5	RSCAN0RFPCTR5	—	<RSCAN0_base> + 010C$_H$	8, 16, 32
Receive FIFO buffer pointer control register 6	RSCAN0RFPCTR6	—	<RSCAN0_base> + 0110$_H$	8, 16, 32
Receive FIFO buffer pointer control register 7	RSCAN0RFPCTR7	—	<RSCAN0_base> + 0114$_H$	8, 16, 32
Transmit/receive FIFO buffer configuration and control register 0	RSCAN0CFCC0	0000 0000$_H$	<RSCAN0_base> + 0118$_H$	8, 16, 32

chanhvinh.com

Table 21.10 List of RS-CAN Module Registers (2/12)

Register Name	Symbol	After Reset	Address	Access Size
Transmit/receive FIFO buffer configuration and control register 1	RSCAN0CFCC1	0000 0000$_H$	<RSCAN0_base> + 011C$_H$	8, 16, 32
Transmit/receive FIFO buffer configuration and control register 2	RSCAN0CFCC2	0000 0000$_H$	<RSCAN0_base> + 0120$_H$	8, 16, 32
Transmit/receive FIFO buffer configuration and control register 3	RSCAN0CFCC3	0000 0000$_H$	<RSCAN0_base> + 0124$_H$	8, 16, 32
Transmit/receive FIFO buffer configuration and control register 4	RSCAN0CFCC4	0000 0000$_H$	<RSCAN0_base> + 0128$_H$	8, 16, 32
Transmit/receive FIFO buffer configuration and control register 5	RSCAN0CFCC5	0000 0000$_H$	<RSCAN0_base> + 012C$_H$	8, 16, 32
Transmit/receive FIFO buffer status register 0	RSCAN0CFSTS0	0000 0001$_H$	<RSCAN0_base> + 0178$_H$	8, 16, 32
Transmit/receive FIFO buffer status register 1	RSCAN0CFSTS1	0000 0001$_H$	<RSCAN0_base> + 017C$_H$	8, 16, 32
Transmit/receive FIFO buffer status register 2	RSCAN0CFSTS2	0000 0001$_H$	<RSCAN0_base> + 0180$_H$	8, 16, 32
Transmit/receive FIFO buffer status register 3	RSCAN0CFSTS3	0000 0001$_H$	<RSCAN0_base> + 0184$_H$	8, 16, 32
Transmit/receive FIFO buffer status register 4	RSCAN0CFSTS4	0000 0001$_H$	<RSCAN0_base> + 0188$_H$	8, 16, 32
Transmit/receive FIFO buffer status register 5	RSCAN0CFSTS5	0000 0001$_H$	<RSCAN0_base> + 018C$_H$	8, 16, 32
Transmit/receive FIFO buffer pointer control register 0	RSCAN0CFPCTR0	—	<RSCAN0_base> + 01D8$_H$	8, 16, 32
Transmit/receive FIFO buffer pointer control register 1	RSCAN0CFPCTR1	—	<RSCAN0_base> + 01DC$_H$	8, 16, 32
Transmit/receive FIFO buffer pointer control register 2	RSCAN0CFPCTR2	—	<RSCAN0_base> + 01E0$_H$	8, 16, 32
Transmit/receive FIFO buffer pointer control register 3	RSCAN0CFPCTR3	—	<RSCAN0_base> + 01E4$_H$	8, 16, 32
Transmit/receive FIFO buffer pointer control register 4	RSCAN0CFPCTR4	—	<RSCAN0_base> + 01E8$_H$	8, 16, 32
Transmit/receive FIFO buffer pointer control register 5	RSCAN0CFPCTR5	—	<RSCAN0_base> + 01EC$_H$	8, 16, 32
FIFO empty status register	RSCAN0FESTS	007F FFFF$_H$	<RSCAN0_base> + 0238$_H$	8, 16, 32
FIFO full status register	RSCAN0FFSTS	0000 0000$_H$	<RSCAN0_base> + 023C$_H$	8, 16, 32
FIFO Msg lost status register	RSCAN0FMSTS	0000 0000$_H$	<RSCAN0_base> + 0240$_H$	8, 16, 32
Receive FIFO buffer interrupt flag status register	RSCAN0RFISTS	0000 0000$_H$	<RSCAN0_base> + 0244$_H$	8, 16, 32
Transmit/receive FIFO buffer receive interrupt flag status register	RSCAN0CFRISTS	0000 0000$_H$	<RSCAN0_base> + 0248$_H$	8, 16, 32
Transmit/receive FIFO buffer transmit interrupt flag status register	RSCAN0CFTISTS	0000 0000$_H$	<RSCAN0_base> + 024C$_H$	8, 16, 32
Transmit buffer control register 0	RSCAN0TMC0	00$_H$	<RSCAN0_base> + 0250$_H$	8
Transmit buffer control register 1	RSCAN0TMC1	00$_H$	<RSCAN0_base> + 0251$_H$	8
Transmit buffer control register 2	RSCAN0TMC2	00$_H$	<RSCAN0_base> + 0252$_H$	8
Transmit buffer control register 3	RSCAN0TMC3	00$_H$	<RSCAN0_base> + 0253$_H$	8
Transmit buffer control register 4	RSCAN0TMC4	00$_H$	<RSCAN0_base> + 0254$_H$	8
Transmit buffer control register 5	RSCAN0TMC5	00$_H$	<RSCAN0_base> + 0255$_H$	8
Transmit buffer control register 6	RSCAN0TMC6	00$_H$	<RSCAN0_base> + 0256$_H$	8
Transmit buffer control register 7	RSCAN0TMC7	00$_H$	<RSCAN0_base> + 0257$_H$	8
Transmit buffer control register 8	RSCAN0TMC8	00$_H$	<RSCAN0_base> + 0258$_H$	8
Transmit buffer control register 9	RSCAN0TMC9	00$_H$	<RSCAN0_base> + 0259$_H$	8
Transmit buffer control register 10	RSCAN0TMC10	00$_H$	<RSCAN0_base> + 025A$_H$	8
Transmit buffer control register 11	RSCAN0TMC11	00$_H$	<RSCAN0_base> + 025B$_H$	8
Transmit buffer control register 12	RSCAN0TMC12	00$_H$	<RSCAN0_base> + 025C$_H$	8
Transmit buffer control register 13	RSCAN0TMC13	00$_H$	<RSCAN0_base> + 025D$_H$	8
Transmit buffer control register 14	RSCAN0TMC14	00$_H$	<RSCAN0_base> + 025E$_H$	8
Transmit buffer control register 15	RSCAN0TMC15	00$_H$	<RSCAN0_base> + 025F$_H$	8
Transmit buffer control register 16	RSCAN0TMC16	00$_H$	<RSCAN0_base> + 0260$_H$	8
Transmit buffer control register 17	RSCAN0TMC17	00$_H$	<RSCAN0_base> + 0261$_H$	8
Transmit buffer control register 18	RSCAN0TMC18	00$_H$	<RSCAN0_base> + 0262$_H$	8
Transmit buffer control register 19	RSCAN0TMC19	00$_H$	<RSCAN0_base> + 0263$_H$	8

chanhvinh.com

Table 21.10 List of RS-CAN Module Registers (3/12)

Register Name	Symbol	After Reset	Address	Access Size
Transmit buffer control register 20	RSCAN0TMC20	00_H	<RSCAN0_base> + 0264_H	8
Transmit buffer control register 21	RSCAN0TMC21	00_H	<RSCAN0_base> + 0265_H	8
Transmit buffer control register 22	RSCAN0TMC22	00_H	<RSCAN0_base> + 0266_H	8
Transmit buffer control register 23	RSCAN0TMC23	00_H	<RSCAN0_base> + 0267_H	8
Transmit buffer control register 24	RSCAN0TMC24	00_H	<RSCAN0_base> + 0268_H	8
Transmit buffer control register 25	RSCAN0TMC25	00_H	<RSCAN0_base> + 0269_H	8
Transmit buffer control register 26	RSCAN0TMC26	00_H	<RSCAN0_base> + $026A_H$	8
Transmit buffer control register 27	RSCAN0TMC27	00_H	<RSCAN0_base> + $026B_H$	8
Transmit buffer control register 28	RSCAN0TMC28	00_H	<RSCAN0_base> + $026C_H$	8
Transmit buffer control register 29	RSCAN0TMC29	00_H	<RSCAN0_base> + $026D_H$	8
Transmit buffer control register 30	RSCAN0TMC30	00_H	<RSCAN0_base> + $026E_H$	8
Transmit buffer control register 31	RSCAN0TMC31	00_H	<RSCAN0_base> + $026F_H$	8
Transmit buffer status register 0	RSCAN0TMSTS0	00_H	<RSCAN0_base> + $02D0_H$	8
Transmit buffer status register 1	RSCAN0TMSTS1	00_H	<RSCAN0_base> + $02D1_H$	8
Transmit buffer status register 2	RSCAN0TMSTS2	00_H	<RSCAN0_base> + $02D2_H$	8
Transmit buffer status register 3	RSCAN0TMSTS3	00_H	<RSCAN0_base> + $02D3_H$	8
Transmit buffer status register 4	RSCAN0TMSTS4	00_H	<RSCAN0_base> + $02D4_H$	8
Transmit buffer status register 5	RSCAN0TMSTS5	00_H	<RSCAN0_base> + $02D5_H$	8
Transmit buffer status register 6	RSCAN0TMSTS6	00_H	<RSCAN0_base> + $02D6_H$	8
Transmit buffer status register 7	RSCAN0TMSTS7	00_H	<RSCAN0_base> + $02D7_H$	8
Transmit buffer status register 8	RSCAN0TMSTS8	00_H	<RSCAN0_base> + $02D8_H$	8
Transmit buffer status register 9	RSCAN0TMSTS9	00_H	<RSCAN0_base> + $02D9_H$	8
Transmit buffer status register 10	RSCAN0TMSTS10	00_H	<RSCAN0_base> + $02DA_H$	8
Transmit buffer status register 11	RSCAN0TMSTS11	00_H	<RSCAN0_base> + $02DB_H$	8
Transmit buffer status register 12	RSCAN0TMSTS12	00_H	<RSCAN0_base> + $02DC_H$	8
Transmit buffer status register 13	RSCAN0TMSTS13	00_H	<RSCAN0_base> + $02DD_H$	8
Transmit buffer status register 14	RSCAN0TMSTS14	00_H	<RSCAN0_base> + $02DE_H$	8
Transmit buffer status register 15	RSCAN0TMSTS15	00_H	<RSCAN0_base> + $02DF_H$	8
Transmit buffer status register 16	RSCAN0TMSTS16	00_H	<RSCAN0_base> + $02E0_H$	8
Transmit buffer status register 17	RSCAN0TMSTS17	00_H	<RSCAN0_base> + $02E1_H$	8
Transmit buffer status register 18	RSCAN0TMSTS18	00_H	<RSCAN0_base> + $02E2_H$	8
Transmit buffer status register 19	RSCAN0TMSTS19	00_H	<RSCAN0_base> + $02E3_H$	8
Transmit buffer status register 20	RSCAN0TMSTS20	00_H	<RSCAN0_base> + $02E4_H$	8
Transmit buffer status register 21	RSCAN0TMSTS21	00_H	<RSCAN0_base> + $02E5_H$	8
Transmit buffer status register 22	RSCAN0TMSTS22	00_H	<RSCAN0_base> + $02E6_H$	8
Transmit buffer status register 23	RSCAN0TMSTS23	00_H	<RSCAN0_base> + $02E7_H$	8
Transmit buffer status register 24	RSCAN0TMSTS24	00_H	<RSCAN0_base> + $02E8_H$	8
Transmit buffer status register 25	RSCAN0TMSTS25	00_H	<RSCAN0_base> + $02E9_H$	8
Transmit buffer status register 26	RSCAN0TMSTS26	00_H	<RSCAN0_base> + $02EA_H$	8
Transmit buffer status register 27	RSCAN0TMSTS27	00_H	<RSCAN0_base> + $02EB_H$	8
Transmit buffer status register 28	RSCAN0TMSTS28	00_H	<RSCAN0_base> + $02EC_H$	8
Transmit buffer status register 29	RSCAN0TMSTS29	00_H	<RSCAN0_base> + $02ED_H$	8
Transmit buffer status register 30	RSCAN0TMSTS30	00_H	<RSCAN0_base> + $02EE_H$	8
Transmit buffer status register 31	RSCAN0TMSTS31	00_H	<RSCAN0_base> + $02EF_H$	8
Transmit buffer transmit request status register 0	RSCAN0TMTRSTS0	$0000\ 0000_H$	<RSCAN0_base> + 0350_H	8, 16, 32
Transmit buffer transmit abort request status register 0	RSCAN0TMTARSTS0	$0000\ 0000_H$	<RSCAN0_base> + 0360_H	8, 16, 32
Transmit buffer transmit complete status register 0	RSCAN0TMTCSTS0	$0000\ 0000_H$	<RSCAN0_base> + 0370_H	8, 16, 32

chanhvinh.com

Table 21.10 List of RS-CAN Module Registers (4/12)

Register Name	Symbol	After Reset	Address	Access Size
Transmit buffer transmit abort status register 0	RSCAN0TMTASTS0	0000 0000$_H$	<RSCAN0_base> + 0380$_H$	8, 16, 32
Transmit buffer interrupt enable configuration register 0	RSCAN0TMIEC0	0000 0000$_H$	<RSCAN0_base> + 0390$_H$	8, 16, 32
Transmit queue configuration and control register 0	RSCAN0TXQCC0	0000 0000$_H$	<RSCAN0_base> + 03A0$_H$	8, 16, 32
Transmit queue configuration and control register 1	RSCAN0TXQCC1	0000 0000$_H$	<RSCAN0_base> + 03A4$_H$	8, 16, 32
Transmit queue status register 0	RSCAN0TXQSTS0	0000 0001$_H$	<RSCAN0_base> + 03C0$_H$	8, 16, 32
Transmit queue status register 1	RSCAN0TXQSTS1	0000 0001$_H$	<RSCAN0_base> + 03C4$_H$	8, 16, 32
Transmit queue pointer control register 0	RSCAN0TXQPCTR0	—	<RSCAN0_base> + 03E0$_H$	8, 16, 32
Transmit queue pointer control register 1	RSCAN0TXQPCTR1	—	<RSCAN0_base> + 03E4$_H$	8, 16, 32
Transmit history configuration and control register 0	RSCAN0THLCC0	0000 0000$_H$	<RSCAN0_base> + 0400$_H$	8, 16, 32
Transmit history configuration and control register 1	RSCAN0THLCC1	0000 0000$_H$	<RSCAN0_base> + 0404$_H$	8, 16, 32
Transmit history status register 0	RSCAN0THLSTS0	0000 0001$_H$	<RSCAN0_base> + 0420$_H$	8, 16, 32
Transmit history status register 1	RSCAN0THLSTS1	0000 0001$_H$	<RSCAN0_base> + 0424$_H$	8, 16, 32
Transmit history pointer control register 0	RSCAN0THLPCTR0	0000 0000$_H$	<RSCAN0_base> + 0440$_H$	8, 16, 32
Transmit history pointer control register 1	RSCAN0THLPCTR1	0000 0000$_H$	<RSCAN0_base> + 0444$_H$	8, 16, 32
Global TX interrupt status register 0	RSCAN0GTINTSTS0	0000 0000$_H$	<RSCAN0_base> + 0460$_H$	8, 16, 32
Global test configuration register	RSCAN0GTSTCFG	0000 0000$_H$	<RSCAN0_base> + 0468$_H$	8, 16, 32
Global test control register	RSCAN0GTSTCTR	0000 0000$_H$	<RSCAN0_base> + 046C$_H$	8, 16, 32
Global lock key register	RSCAN0GLOCKK	—	<RSCAN0_base> + 047C$_H$	16, 32
Receive rule ID register 0	RSCAN0GAFLID0	0000 0000$_H$	<RSCAN0_base> + 0500$_H$	8, 16, 32
Receive rule mask register 0	RSCAN0GAFLM0	0000 0000$_H$	<RSCAN0_base> + 0504$_H$	8, 16, 32
Receive rule pointer 0 register 0	RSCAN0GAFLP00	0000 0000$_H$	<RSCAN0_base> + 0508$_H$	8, 16, 32
Receive rule pointer 1 register 0	RSCAN0GAFLP10	0000 0000$_H$	<RSCAN0_base> + 050C$_H$	8, 16, 32
Receive rule ID register 1	RSCAN0GAFLID1	0000 0000$_H$	<RSCAN0_base> + 0510$_H$	8, 16, 32
Receive rule mask register 1	RSCAN0GAFLM1	0000 0000$_H$	<RSCAN0_base> + 0514$_H$	8, 16, 32
Receive rule pointer 0 register 1	RSCAN0GAFLP01	0000 0000$_H$	<RSCAN0_base> + 0518$_H$	8, 16, 32
Receive rule pointer 1 register 1	RSCAN0GAFLP11	0000 0000$_H$	<RSCAN0_base> + 051C$_H$	8, 16, 32
Receive rule ID register 2	RSCAN0GAFLID2	0000 0000$_H$	<RSCAN0_base> + 0520$_H$	8, 16, 32
Receive rule mask register 2	RSCAN0GAFLM2	0000 0000$_H$	<RSCAN0_base> + 0524$_H$	8, 16, 32
Receive rule pointer 0 register 2	RSCAN0GAFLP02	0000 0000$_H$	<RSCAN0_base> + 0528$_H$	8, 16, 32
Receive rule pointer 1 register 2	RSCAN0GAFLP12	0000 0000$_H$	<RSCAN0_base> + 052C$_H$	8, 16, 32
Receive rule ID register 3	RSCAN0GAFLID3	0000 0000$_H$	<RSCAN0_base> + 0530$_H$	8, 16, 32
Receive rule mask register 3	RSCAN0GAFLM3	0000 0000$_H$	<RSCAN0_base> + 0534$_H$	8, 16, 32
Receive rule pointer 0 register 3	RSCAN0GAFLP03	0000 0000$_H$	<RSCAN0_base> + 0538$_H$	8, 16, 32
Receive rule pointer 1 register 3	RSCAN0GAFLP13	0000 0000$_H$	<RSCAN0_base> + 053C$_H$	8, 16, 32
Receive rule ID register 4	RSCAN0GAFLID4	0000 0000$_H$	<RSCAN0_base> + 0540$_H$	8, 16, 32
Receive rule mask register 4	RSCAN0GAFLM4	0000 0000$_H$	<RSCAN0_base> + 0544$_H$	8, 16, 32
Receive rule pointer 0 register 4	RSCAN0GAFLP04	0000 0000$_H$	<RSCAN0_base> + 0548$_H$	8, 16, 32
Receive rule pointer 1 register 4	RSCAN0GAFLP14	0000 0000$_H$	<RSCAN0_base> + 054C$_H$	8, 16, 32
Receive rule ID register 5	RSCAN0GAFLID5	0000 0000$_H$	<RSCAN0_base> + 0550$_H$	8, 16, 32
Receive rule mask register 5	RSCAN0GAFLM5	0000 0000$_H$	<RSCAN0_base> + 0554$_H$	8, 16, 32
Receive rule pointer 0 register 5	RSCAN0GAFLP05	0000 0000$_H$	<RSCAN0_base> + 0558$_H$	8, 16, 32
Receive rule pointer 1 register 5	RSCAN0GAFLP15	0000 0000$_H$	<RSCAN0_base> + 055C$_H$	8, 16, 32
Receive rule ID register 6	RSCAN0GAFLID6	0000 0000$_H$	<RSCAN0_base> + 0560$_H$	8, 16, 32
Receive rule mask register 6	RSCAN0GAFLM6	0000 0000$_H$	<RSCAN0_base> + 0564$_H$	8, 16, 32
Receive rule pointer 0 register 6	RSCAN0GAFLP06	0000 0000$_H$	<RSCAN0_base> + 0568$_H$	8, 16, 32
Receive rule pointer 1 register 6	RSCAN0GAFLP16	0000 0000$_H$	<RSCAN0_base> + 056C$_H$	8, 16, 32
Receive rule ID register 7	RSCAN0GAFLID7	0000 0000$_H$	<RSCAN0_base> + 0570$_H$	8, 16, 32

chanhvinh.com

Table 21.10 List of RS-CAN Module Registers (5/12)

Register Name	Symbol	After Reset	Address	Access Size
Receive rule mask register 7	RSCAN0GAFLM7	0000 0000$_H$	<RSCAN0_base> + 0574$_H$	8, 16, 32
Receive rule pointer 0 register 7	RSCAN0GAFLP07	0000 0000$_H$	<RSCAN0_base> + 0578$_H$	8, 16, 32
Receive rule pointer 1 register 7	RSCAN0GAFLP17	0000 0000$_H$	<RSCAN0_base> + 057C$_H$	8, 16, 32
Receive rule ID register 8	RSCAN0GAFLID8	0000 0000$_H$	<RSCAN0_base> + 0580$_H$	8, 16, 32
Receive rule mask register 8	RSCAN0GAFLM8	0000 0000$_H$	<RSCAN0_base> + 0584$_H$	8, 16, 32
Receive rule pointer 0 register 8	RSCAN0GAFLP08	0000 0000$_H$	<RSCAN0_base> + 0588$_H$	8, 16, 32
Receive rule pointer 1 register 8	RSCAN0GAFLP18	0000 0000$_H$	<RSCAN0_base> + 058C$_H$	8, 16, 32
Receive rule ID register 9	RSCAN0GAFLID9	0000 0000$_H$	<RSCAN0_base> + 0590$_H$	8, 16, 32
Receive rule mask register 9	RSCAN0GAFLM9	0000 0000$_H$	<RSCAN0_base> + 0594$_H$	8, 16, 32
Receive rule pointer 0 register 9	RSCAN0GAFLP09	0000 0000$_H$	<RSCAN0_base> + 0598$_H$	8, 16, 32
Receive rule pointer 1 register 9	RSCAN0GAFLP19	0000 0000$_H$	<RSCAN0_base> + 059C$_H$	8, 16, 32
Receive rule ID register 10	RSCAN0GAFLID10	0000 0000$_H$	<RSCAN0_base> + 05A0$_H$	8, 16, 32
Receive rule mask register 10	RSCAN0GAFLM10	0000 0000$_H$	<RSCAN0_base> + 05A4$_H$	8, 16, 32
Receive rule pointer 0 register 10	RSCAN0GAFLP010	0000 0000$_H$	<RSCAN0_base> + 05A8$_H$	8, 16, 32
Receive rule pointer 1 register 10	RSCAN0GAFLP110	0000 0000$_H$	<RSCAN0_base> + 05AC$_H$	8, 16, 32
Receive rule ID register 11	RSCAN0GAFLID11	0000 0000$_H$	<RSCAN0_base> + 05B0$_H$	8, 16, 32
Receive rule mask register 11	RSCAN0GAFLM11	0000 0000$_H$	<RSCAN0_base> + 05B4$_H$	8, 16, 32
Receive rule pointer 0 register 11	RSCAN0GAFLP011	0000 0000$_H$	<RSCAN0_base> + 05B8$_H$	8, 16, 32
Receive rule pointer 1 register 11	RSCAN0GAFLP111	0000 0000$_H$	<RSCAN0_base> + 05BC$_H$	8, 16, 32
Receive rule ID register 12	RSCAN0GAFLID12	0000 0000$_H$	<RSCAN0_base> + 05C0$_H$	8, 16, 32
Receive rule mask register 12	RSCAN0GAFLM12	0000 0000$_H$	<RSCAN0_base> + 05C4$_H$	8, 16, 32
Receive rule pointer 0 register 12	RSCAN0GAFLP012	0000 0000$_H$	<RSCAN0_base> + 05C8$_H$	8, 16, 32
Receive rule pointer 1 register 12	RSCAN0GAFLP112	0000 0000$_H$	<RSCAN0_base> + 05CC$_H$	8, 16, 32
Receive rule ID register 13	RSCAN0GAFLID13	0000 0000$_H$	<RSCAN0_base> + 05D0$_H$	8, 16, 32
Receive rule mask register 13	RSCAN0GAFLM13	0000 0000$_H$	<RSCAN0_base> + 05D4$_H$	8, 16, 32
Receive rule pointer 0 register 13	RSCAN0GAFLP013	0000 0000$_H$	<RSCAN0_base> + 05D8$_H$	8, 16, 32
Receive rule pointer 1 register 13	RSCAN0GAFLP113	0000 0000$_H$	<RSCAN0_base> + 05DC$_H$	8, 16, 32
Receive rule ID register 14	RSCAN0GAFLID14	0000 0000$_H$	<RSCAN0_base> + 05E0$_H$	8, 16, 32
Receive rule mask register 14	RSCAN0GAFLM14	0000 0000$_H$	<RSCAN0_base> + 05E4$_H$	8, 16, 32
Receive rule pointer 0 register 14	RSCAN0GAFLP014	0000 0000$_H$	<RSCAN0_base> + 05E8$_H$	8, 16, 32
Receive rule pointer 1 register 14	RSCAN0GAFLP114	0000 0000$_H$	<RSCAN0_base> + 05EC$_H$	8, 16, 32
Receive rule ID register 15	RSCAN0GAFLID15	0000 0000$_H$	<RSCAN0_base> + 05F0$_H$	8, 16, 32
Receive rule mask register 15	RSCAN0GAFLM15	0000 0000$_H$	<RSCAN0_base> + 05F4$_H$	8, 16, 32
Receive rule pointer 0 register 15	RSCAN0GAFLP015	0000 0000$_H$	<RSCAN0_base> + 05F8$_H$	8, 16, 32
Receive rule pointer 1 register 15	RSCAN0GAFLP115	0000 0000$_H$	<RSCAN0_base> + 05FC$_H$	8, 16, 32
Receive buffer ID register 0	RSCAN0RMID0	0000 0000$_H$	<RSCAN0_base> + 0600$_H$	8, 16, 32
Receive buffer pointer register 0	RSCAN0RMPTR0	0000 0000$_H$	<RSCAN0_base> + 0604$_H$	8, 16, 32
Receive buffer data field 0 register 0	RSCAN0RMDF00	0000 0000$_H$	<RSCAN0_base> + 0608$_H$	8, 16, 32
Receive buffer data field 1 register 0	RSCAN0RMDF10	0000 0000$_H$	<RSCAN0_base> + 060C$_H$	8, 16, 32
Receive buffer ID register 1	RSCAN0RMID1	0000 0000$_H$	<RSCAN0_base> + 0610$_H$	8, 16, 32
Receive buffer pointer register 1	RSCAN0RMPTR1	0000 0000$_H$	<RSCAN0_base> + 0614$_H$	8, 16, 32
Receive buffer data field 0 register 1	RSCAN0RMDF01	0000 0000$_H$	<RSCAN0_base> + 0618$_H$	8, 16, 32
Receive buffer data field 1 register 1	RSCAN0RMDF11	0000 0000$_H$	<RSCAN0_base> + 061C$_H$	8, 16, 32
Receive buffer ID register 2	RSCAN0RMID2	0000 0000$_H$	<RSCAN0_base> + 0620$_H$	8, 16, 32
Receive buffer pointer register 2	RSCAN0RMPTR2	0000 0000$_H$	<RSCAN0_base> + 0624$_H$	8, 16, 32
Receive buffer data field 0 register 2	RSCAN0RMDF02	0000 0000$_H$	<RSCAN0_base> + 0628$_H$	8, 16, 32
Receive buffer data field 1 register 2	RSCAN0RMDF12	0000 0000$_H$	<RSCAN0_base> + 062C$_H$	8, 16, 32

chanhvinh.com

Table 21.10 List of RS-CAN Module Registers (6/12)

Register Name	Symbol	After Reset	Address	Access Size
Receive buffer ID register 3	RSCAN0RMID3	0000 0000$_H$	<RSCAN0_base> + 0630$_H$	8, 16, 32
Receive buffer pointer register 3	RSCAN0RMPTR3	0000 0000$_H$	<RSCAN0_base> + 0634$_H$	8, 16, 32
Receive buffer data field 0 register 3	RSCAN0RMDF03	0000 0000$_H$	<RSCAN0_base> + 0638$_H$	8, 16, 32
Receive buffer data field 1 register 3	RSCAN0RMDF13	0000 0000$_H$	<RSCAN0_base> + 063C$_H$	8, 16, 32
Receive buffer ID register 4	RSCAN0RMID4	0000 0000$_H$	<RSCAN0_base> + 0640$_H$	8, 16, 32
Receive buffer pointer register 4	RSCAN0RMPTR4	0000 0000$_H$	<RSCAN0_base> + 0644$_H$	8, 16, 32
Receive buffer data field 0 register 4	RSCAN0RMDF04	0000 0000$_H$	<RSCAN0_base> + 0648$_H$	8, 16, 32
Receive buffer data field 1 register 4	RSCAN0RMDF14	0000 0000$_H$	<RSCAN0_base> + 064C$_H$	8, 16, 32
Receive buffer ID register 5	RSCAN0RMID5	0000 0000$_H$	<RSCAN0_base> + 0650$_H$	8, 16, 32
Receive buffer pointer register 5	RSCAN0RMPTR5	0000 0000$_H$	<RSCAN0_base> + 0654$_H$	8, 16, 32
Receive buffer data field 0 register 5	RSCAN0RMDF05	0000 0000$_H$	<RSCAN0_base> + 0658$_H$	8, 16, 32
Receive buffer data field 1 register 5	RSCAN0RMDF15	0000 0000$_H$	<RSCAN0_base> + 065C$_H$	8, 16, 32
Receive buffer ID register 6	RSCAN0RMID6	0000 0000$_H$	<RSCAN0_base> + 0660$_H$	8, 16, 32
Receive buffer pointer register 6	RSCAN0RMPTR6	0000 0000$_H$	<RSCAN0_base> + 0664$_H$	8, 16, 32
Receive buffer data field 0 register 6	RSCAN0RMDF06	0000 0000$_H$	<RSCAN0_base> + 0668$_H$	8, 16, 32
Receive buffer data field 1 register 6	RSCAN0RMDF16	0000 0000$_H$	<RSCAN0_base> + 066C$_H$	8, 16, 32
Receive buffer ID register 7	RSCAN0RMID7	0000 0000$_H$	<RSCAN0_base> + 0670$_H$	8, 16, 32
Receive buffer pointer register 7	RSCAN0RMPTR7	0000 0000$_H$	<RSCAN0_base> + 0674$_H$	8, 16, 32
Receive buffer data field 0 register 7	RSCAN0RMDF07	0000 0000$_H$	<RSCAN0_base> + 0678$_H$	8, 16, 32
Receive buffer data field 1 register 7	RSCAN0RMDF17	0000 0000$_H$	<RSCAN0_base> + 067C$_H$	8, 16, 32
Receive buffer ID register 8	RSCAN0RMID8	0000 0000$_H$	<RSCAN0_base> + 0680$_H$	8, 16, 32
Receive buffer pointer register 8	RSCAN0RMPTR8	0000 0000$_H$	<RSCAN0_base> + 0684$_H$	8, 16, 32
Receive buffer data field 0 register 8	RSCAN0RMDF08	0000 0000$_H$	<RSCAN0_base> + 0688$_H$	8, 16, 32
Receive buffer data field 1 register 8	RSCAN0RMDF18	0000 0000$_H$	<RSCAN0_base> + 068C$_H$	8, 16, 32
Receive buffer ID register 9	RSCAN0RMID9	0000 0000$_H$	<RSCAN0_base> + 0690$_H$	8, 16, 32
Receive buffer pointer register 9	RSCAN0RMPTR9	0000 0000$_H$	<RSCAN0_base> + 0694$_H$	8, 16, 32
Receive buffer data field 0 register 9	RSCAN0RMDF09	0000 0000$_H$	<RSCAN0_base> + 0698$_H$	8, 16, 32
Receive buffer data field 1 register 9	RSCAN0RMDF19	0000 0000$_H$	<RSCAN0_base> + 069C$_H$	8, 16, 32
Receive buffer ID register 10	RSCAN0RMID10	0000 0000$_H$	<RSCAN0_base> + 06A0$_H$	8, 16, 32
Receive buffer pointer register 10	RSCAN0RMPTR10	0000 0000$_H$	<RSCAN0_base> + 06A4$_H$	8, 16, 32
Receive buffer data field 0 register 10	RSCAN0RMDF010	0000 0000$_H$	<RSCAN0_base> + 06A8$_H$	8, 16, 32
Receive buffer data field 1 register 10	RSCAN0RMDF110	0000 0000$_H$	<RSCAN0_base> + 06AC$_H$	8, 16, 32
Receive buffer ID register 11	RSCAN0RMID11	0000 0000$_H$	<RSCAN0_base> + 06B0$_H$	8, 16, 32
Receive buffer pointer register 11	RSCAN0RMPTR11	0000 0000$_H$	<RSCAN0_base> + 06B4$_H$	8, 16, 32
Receive buffer data field 0 register 11	RSCAN0RMDF011	0000 0000$_H$	<RSCAN0_base> + 06B8$_H$	8, 16, 32
Receive buffer data field 1 register 11	RSCAN0RMDF111	0000 0000$_H$	<RSCAN0_base> + 06BC$_H$	8, 16, 32
Receive buffer ID register 12	RSCAN0RMID12	0000 0000$_H$	<RSCAN0_base> + 06C0$_H$	8, 16, 32
Receive buffer pointer register 12	RSCAN0RMPTR12	0000 0000$_H$	<RSCAN0_base> + 06C4$_H$	8, 16, 32
Receive buffer data field 0 register 12	RSCAN0RMDF012	0000 0000$_H$	<RSCAN0_base> + 06C8$_H$	8, 16, 32
Receive buffer data field 1 register 12	RSCAN0RMDF112	0000 0000$_H$	<RSCAN0_base> + 06CC$_H$	8, 16, 32
Receive buffer ID register 13	RSCAN0RMID13	0000 0000$_H$	<RSCAN0_base> + 06D0$_H$	8, 16, 32
Receive buffer pointer register 13	RSCAN0RMPTR13	0000 0000$_H$	<RSCAN0_base> + 06D4$_H$	8, 16, 32
Receive buffer data field 0 register 13	RSCAN0RMDF013	0000 0000$_H$	<RSCAN0_base> + 06D8$_H$	8, 16, 32
Receive buffer data field 1 register 13	RSCAN0RMDF113	0000 0000$_H$	<RSCAN0_base> + 06DC$_H$	8, 16, 32
Receive buffer ID register 14	RSCAN0RMID14	0000 0000$_H$	<RSCAN0_base> + 06E0$_H$	8, 16, 32
Receive buffer pointer register 14	RSCAN0RMPTR14	0000 0000$_H$	<RSCAN0_base> + 06E4$_H$	8, 16, 32
Receive buffer data field 0 register 14	RSCAN0RMDF014	0000 0000$_H$	<RSCAN0_base> + 06E8$_H$	8, 16, 32

chanhvinh.com

Table 21.10 List of RS-CAN Module Registers (7/12)

Register Name	Symbol	After Reset	Address	Access Size
Receive buffer data field 1 register 14	RSCAN0RMDF114	0000 0000$_H$	<RSCAN0_base> + 06EC$_H$	8, 16, 32
Receive buffer ID register 15	RSCAN0RMID15	0000 0000$_H$	<RSCAN0_base> + 06F0$_H$	8, 16, 32
Receive buffer pointer register 15	RSCAN0RMPTR15	0000 0000$_H$	<RSCAN0_base> + 06F4$_H$	8, 16, 32
Receive buffer data field 0 register 15	RSCAN0RMDF015	0000 0000$_H$	<RSCAN0_base> + 06F8$_H$	8, 16, 32
Receive buffer data field 1 register 15	RSCAN0RMDF115	0000 0000$_H$	<RSCAN0_base> + 06FC$_H$	8, 16, 32
Receive buffer ID register 16	RSCAN0RMID16	0000 0000$_H$	<RSCAN0_base> + 0700$_H$	8, 16, 32
Receive buffer pointer register 16	RSCAN0RMPTR16	0000 0000$_H$	<RSCAN0_base> + 0704$_H$	8, 16, 32
Receive buffer data field 0 register 16	RSCAN0RMDF016	0000 0000$_H$	<RSCAN0_base> + 0708$_H$	8, 16, 32
Receive buffer data field 1 register16	RSCAN0RMDF116	0000 0000$_H$	<RSCAN0_base> + 070C$_H$	8, 16, 32
Receive buffer ID register 17	RSCAN0RMID17	0000 0000$_H$	<RSCAN0_base> + 0710$_H$	8, 16, 32
Receive buffer pointer register 17	RSCAN0RMPTR17	0000 0000$_H$	<RSCAN0_base> + 0714$_H$	8, 16, 32
Receive buffer data field 0 register 17	RSCAN0RMDF017	0000 0000$_H$	<RSCAN0_base> + 0718$_H$	8, 16, 32
Receive buffer data field 1 register 17	RSCAN0RMDF117	0000 0000$_H$	<RSCAN0_base> + 071C$_H$	8, 16, 32
Receive buffer ID register 18	RSCAN0RMID18	0000 0000$_H$	<RSCAN0_base> + 0720$_H$	8, 16, 32
Receive buffer pointer register 18	RSCAN0RMPTR18	0000 0000$_H$	<RSCAN0_base> + 0724$_H$	8, 16, 32
Receive buffer data field 0 register 18	RSCAN0RMDF018	0000 0000$_H$	<RSCAN0_base> + 0728$_H$	8, 16, 32
Receive buffer data field 1 register 18	RSCAN0RMDF118	0000 0000$_H$	<RSCAN0_base> + 072C$_H$	8, 16, 32
Receive buffer ID register 19	RSCAN0RMID19	0000 0000$_H$	<RSCAN0_base> + 0730$_H$	8, 16, 32
Receive buffer pointer register 19	RSCAN0RMPTR19	0000 0000$_H$	<RSCAN0_base> + 0734$_H$	8, 16, 32
Receive buffer data field 0 register 19	RSCAN0RMDF019	0000 0000$_H$	<RSCAN0_base> + 0738$_H$	8, 16, 32
Receive buffer data field 1 register 19	RSCAN0RMDF119	0000 0000$_H$	<RSCAN0_base> + 073C$_H$	8, 16, 32
Receive buffer ID register 20	RSCAN0RMID20	0000 0000$_H$	<RSCAN0_base> + 0740$_H$	8, 16, 32
Receive buffer pointer register 20	RSCAN0RMPTR20	0000 0000$_H$	<RSCAN0_base> + 0744$_H$	8, 16, 32
Receive buffer data field 0 register 20	RSCAN0RMDF020	0000 0000$_H$	<RSCAN0_base> + 0748$_H$	8, 16, 32
Receive buffer data field 1 register 20	RSCAN0RMDF120	0000 0000$_H$	<RSCAN0_base> + 074C$_H$	8, 16, 32
Receive buffer ID register 21	RSCAN0RMID21	0000 0000$_H$	<RSCAN0_base> + 0750$_H$	8, 16, 32
Receive buffer pointer register 21	RSCAN0RMPTR21	0000 0000$_H$	<RSCAN0_base> + 0754$_H$	8, 16, 32
Receive buffer data field 0 register 21	RSCAN0RMDF021	0000 0000$_H$	<RSCAN0_base> + 0758$_H$	8, 16, 32
Receive buffer data field 1 register 21	RSCAN0RMDF121	0000 0000$_H$	<RSCAN0_base> + 075C$_H$	8, 16, 32
Receive buffer ID register 22	RSCAN0RMID22	0000 0000$_H$	<RSCAN0_base> + 0760$_H$	8, 16, 32
Receive buffer pointer register 22	RSCAN0RMPTR22	0000 0000$_H$	<RSCAN0_base> + 0764$_H$	8, 16, 32
Receive buffer data field 0 register 22	RSCAN0RMDF022	0000 0000$_H$	<RSCAN0_base> + 0768$_H$	8, 16, 32
Receive buffer data field 1 register 22	RSCAN0RMDF122	0000 0000$_H$	<RSCAN0_base> + 076C$_H$	8, 16, 32
Receive buffer ID register 23	RSCAN0RMID23	0000 0000$_H$	<RSCAN0_base> + 0770$_H$	8, 16, 32
Receive buffer pointer register 23	RSCAN0RMPTR23	0000 0000$_H$	<RSCAN0_base> + 0774$_H$	8, 16, 32
Receive buffer data field 0 register 23	RSCAN0RMDF023	0000 0000$_H$	<RSCAN0_base> + 0778$_H$	8, 16, 32
Receive buffer data field 1 register 23	RSCAN0RMDF123	0000 0000$_H$	<RSCAN0_base> + 077C$_H$	8, 16, 32
Receive buffer ID register 24	RSCAN0RMID24	0000 0000$_H$	<RSCAN0_base> + 0780$_H$	8, 16, 32
Receive buffer pointer register 24	RSCAN0RMPTR24	0000 0000$_H$	<RSCAN0_base> + 0784$_H$	8, 16, 32
Receive buffer data field 0 register 24	RSCAN0RMDF024	0000 0000$_H$	<RSCAN0_base> + 0788$_H$	8, 16, 32
Receive buffer data field 1 register 24	RSCAN0RMDF124	0000 0000$_H$	<RSCAN0_base> + 078C$_H$	8, 16, 32
Receive buffer ID register 25	RSCAN0RMID25	0000 0000$_H$	<RSCAN0_base> + 0790$_H$	8, 16, 32
Receive buffer pointer register 25	RSCAN0RMPTR25	0000 0000$_H$	<RSCAN0_base> + 0794$_H$	8, 16, 32
Receive buffer data field 0 register 25	RSCAN0RMDF025	0000 0000$_H$	<RSCAN0_base> + 0798$_H$	8, 16, 32
Receive buffer data field 1 register 25	RSCAN0RMDF125	0000 0000$_H$	<RSCAN0_base> + 079C$_H$	8, 16, 32
Receive buffer ID register 26	RSCAN0RMID26	0000 0000$_H$	<RSCAN0_base> + 07A0$_H$	8, 16, 32
Receive buffer pointer register 26	RSCAN0RMPTR26	0000 0000$_H$	<RSCAN0_base> + 07A4$_H$	8, 16, 32

chanhvinh.com

Table 21.10 List of RS-CAN Module Registers (8/12)

Register Name	Symbol	After Reset	Address	Access Size
Receive buffer data field 0 register 26	RSCAN0RMDF026	0000 0000$_H$	<RSCAN0_base> + 07A8$_H$	8, 16, 32
Receive buffer data field 1 register 26	RSCAN0RMDF126	0000 0000$_H$	<RSCAN0_base> + 07AC$_H$	8, 16, 32
Receive buffer ID register 27	RSCAN0RMID27	0000 0000$_H$	<RSCAN0_base> + 07B0$_H$	8, 16, 32
Receive buffer pointer register 27	RSCAN0RMPTR27	0000 0000$_H$	<RSCAN0_base> + 07B4$_H$	8, 16, 32
Receive buffer data field 0 register 27	RSCAN0RMDF027	0000 0000$_H$	<RSCAN0_base> + 07B8$_H$	8, 16, 32
Receive buffer data field 1 register 27	RSCAN0RMDF127	0000 0000$_H$	<RSCAN0_base> + 07BC$_H$	8, 16, 32
Receive buffer ID register 28	RSCAN0RMID28	0000 0000$_H$	<RSCAN0_base> + 07C0$_H$	8, 16, 32
Receive buffer pointer register 28	RSCAN0RMPTR28	0000 0000$_H$	<RSCAN0_base> + 07C4$_H$	8, 16, 32
Receive buffer data field 0 register 28	RSCAN0RMDF028	0000 0000$_H$	<RSCAN0_base> + 07C8$_H$	8, 16, 32
Receive buffer data field 1 register 28	RSCAN0RMDF128	0000 0000$_H$	<RSCAN0_base> + 07CC$_H$	8, 16, 32
Receive buffer ID register 29	RSCAN0RMID29	0000 0000$_H$	<RSCAN0_base> + 07D0$_H$	8, 16, 32
Receive buffer pointer register 29	RSCAN0RMPTR29	0000 0000$_H$	<RSCAN0_base> + 07D4$_H$	8, 16, 32
Receive buffer data field 0 register 29	RSCAN0RMDF029	0000 0000$_H$	<RSCAN0_base> + 07D8$_H$	8, 16, 32
Receive buffer data field 1 register 29	RSCAN0RMDF129	0000 0000$_H$	<RSCAN0_base> + 07DC$_H$	8, 16, 32
Receive buffer ID register 30	RSCAN0RMID30	0000 0000$_H$	<RSCAN0_base> + 07E0$_H$	8, 16, 32
Receive buffer pointer register 30	RSCAN0RMPTR30	0000 0000$_H$	<RSCAN0_base> + 07E4$_H$	8, 16, 32
Receive buffer data field 0 register 30	RSCAN0RMDF030	0000 0000$_H$	<RSCAN0_base> + 07E8$_H$	8, 16, 32
Receive buffer data field 1 register 30	RSCAN0RMDF130	0000 0000$_H$	<RSCAN0_base> + 07EC$_H$	8, 16, 32
Receive buffer ID register 31	RSCAN0RMID31	0000 0000$_H$	<RSCAN0_base> + 07F0$_H$	8, 16, 32
Receive buffer pointer register 31	RSCAN0RMPTR31	0000 0000$_H$	<RSCAN0_base> + 07F4$_H$	8, 16, 32
Receive buffer data field 0 register 31	RSCAN0RMDF031	0000 0000$_H$	<RSCAN0_base> + 07F8$_H$	8, 16, 32
Receive buffer data field 1 register 31	RSCAN0RMDF131	0000 0000$_H$	<RSCAN0_base> + 07FC$_H$	8, 16, 32
Receive FIFO buffer access ID register 0	RSCAN0RFID0	0000 0000$_H$	<RSCAN0_base> + 0E00$_H$	8, 16, 32
Receive FIFO buffer access pointer register 0	RSCAN0RFPTR0	0000 0000$_H$	<RSCAN0_base> + 0E04$_H$	8, 16, 32
Receive FIFO buffer access data field 0 register 0	RSCAN0RFDF00	0000 0000$_H$	<RSCAN0_base> + 0E08$_H$	8, 16, 32
Receive FIFO buffer access data field 1 register 0	RSCAN0RFDF10	0000 0000$_H$	<RSCAN0_base> + 0E0C$_H$	8, 16, 32
Receive FIFO buffer access ID register 1	RSCAN0RFID1	0000 0000$_H$	<RSCAN0_base> + 0E10$_H$	8, 16, 32
Receive FIFO buffer access pointer register 1	RSCAN0RFPTR1	0000 0000$_H$	<RSCAN0_base> + 0E14$_H$	8, 16, 32
Receive FIFO buffer access data field 0 register 1	RSCAN0RFDF01	0000 0000$_H$	<RSCAN0_base> + 0E18$_H$	8, 16, 32
Receive FIFO buffer access data field 1 register 1	RSCAN0RFDF11	0000 0000$_H$	<RSCAN0_base> + 0E1C$_H$	8, 16, 32
Receive FIFO buffer access ID register 2	RSCAN0RFID2	0000 0000$_H$	<RSCAN0_base> + 0E20$_H$	8, 16, 32
Receive FIFO buffer access pointer register 2	RSCAN0RFPTR2	0000 0000$_H$	<RSCAN0_base> + 0E24$_H$	8, 16, 32
Receive FIFO buffer access data field 0 register 2	RSCAN0RFDF02	0000 0000$_H$	<RSCAN0_base> + 0E28$_H$	8, 16, 32
Receive FIFO buffer access data field 1 register 2	RSCAN0RFDF12	0000 0000$_H$	<RSCAN0_base> + 0E2C$_H$	8, 16, 32
Receive FIFO buffer access ID register 3	RSCAN0RFID3	0000 0000$_H$	<RSCAN0_base> + 0E30$_H$	8, 16, 32
Receive FIFO buffer access pointer register 3	RSCAN0RFPTR3	0000 0000$_H$	<RSCAN0_base> + 0E34$_H$	8, 16, 32
Receive FIFO buffer access data field 0 register 3	RSCAN0RFDF03	0000 0000$_H$	<RSCAN0_base> + 0E38$_H$	8, 16, 32
Receive FIFO buffer access data field 1 register 3	RSCAN0RFDF13	0000 0000$_H$	<RSCAN0_base> + 0E3C$_H$	8, 16, 32
Receive FIFO buffer access ID register 4	RSCAN0RFID4	0000 0000$_H$	<RSCAN0_base> + 0E40$_H$	8, 16, 32
Receive FIFO buffer access pointer register 4	RSCAN0RFPTR4	0000 0000$_H$	<RSCAN0_base> + 0E44$_H$	8, 16, 32
Receive FIFO buffer access data field 0 register 4	RSCAN0RFDF04	0000 0000$_H$	<RSCAN0_base> + 0E48$_H$	8, 16, 32
Receive FIFO buffer access data field 1 register 4	RSCAN0RFDF14	0000 0000$_H$	<RSCAN0_base> + 0E4C$_H$	8, 16, 32
Receive FIFO buffer access ID register 5	RSCAN0RFID5	0000 0000$_H$	<RSCAN0_base> + 0E50$_H$	8, 16, 32
Receive FIFO buffer access pointer register 5	RSCAN0RFPTR5	0000 0000$_H$	<RSCAN0_base> + 0E54$_H$	8, 16, 32
Receive FIFO buffer access data field 0 register 5	RSCAN0RFDF05	0000 0000$_H$	<RSCAN0_base> + 0E58$_H$	8, 16, 32
Receive FIFO buffer access data field 1 register 5	RSCAN0RFDF15	0000 0000$_H$	<RSCAN0_base> + 0E5C$_H$	8, 16, 32
Receive FIFO buffer access ID register 6	RSCAN0RFID6	0000 0000$_H$	<RSCAN0_base> + 0E60$_H$	8, 16, 32

chanhvinh.com

Table 21.10 List of RS-CAN Module Registers (9/12)

Register Name	Symbol	After Reset	Address	Access Size
Receive FIFO buffer access pointer register 6	RSCAN0RFPTR6	0000 0000$_H$	<RSCAN0_base> + 0E64$_H$	8, 16, 32
Receive FIFO buffer access data field 0 register 6	RSCAN0RFDF06	0000 0000$_H$	<RSCAN0_base> + 0E68$_H$	8, 16, 32
Receive FIFO buffer access data field 1 register 6	RSCAN0RFDF16	0000 0000$_H$	<RSCAN0_base> + 0E6C$_H$	8, 16, 32
Receive FIFO buffer access ID register 7	RSCAN0RFID7	0000 0000$_H$	<RSCAN0_base> + 0E70$_H$	8, 16, 32
Receive FIFO buffer access pointer register 7	RSCAN0RFPTR7	0000 0000$_H$	<RSCAN0_base> + 0E74$_H$	8, 16, 32
Receive FIFO buffer access data field 0 register 7	RSCAN0RFDF07	0000 0000$_H$	<RSCAN0_base> + 0E78$_H$	8, 16, 32
Receive FIFO buffer access data field 1 register 7	RSCAN0RFDF17	0000 0000$_H$	<RSCAN0_base> + 0E7C$_H$	8, 16, 32
Transmit/receive FIFO buffer access ID register 0	RSCAN0CFID0	0000 0000$_H$	<RSCAN0_base> + 0E80$_H$	8, 16, 32
Transmit/receive FIFO buffer access pointer register 0	RSCAN0CFPTR0	0000 0000$_H$	<RSCAN0_base> + 0E84$_H$	8, 16, 32
Transmit/receive FIFO buffer access data field 0 register 0	RSCAN0CFDF00	0000 0000$_H$	<RSCAN0_base> + 0E88$_H$	8, 16, 32
Transmit/receive FIFO buffer access data field 1 register 0	RSCAN0CFDF10	0000 0000$_H$	<RSCAN0_base> + 0E8C$_H$	8, 16, 32
Transmit/receive FIFO buffer access ID register 1	RSCAN0CFID1	0000 0000$_H$	<RSCAN0_base> + 0E90$_H$	8, 16, 32
Transmit/receive FIFO buffer access pointer register 1	RSCAN0CFPTR1	0000 0000$_H$	<RSCAN0_base> + 0E94$_H$	8, 16, 32
Transmit/receive FIFO buffer access data field 0 register 1	RSCAN0CFDF01	0000 0000$_H$	<RSCAN0_base> + 0E98$_H$	8, 16, 32
Transmit/receive FIFO buffer access data field 1 register 1	RSCAN0CFDF11	0000 0000$_H$	<RSCAN0_base> + 0E9C$_H$	8, 16, 32
Transmit/receive FIFO buffer access ID register 2	RSCAN0CFID2	0000 0000$_H$	<RSCAN0_base> + 0EA0$_H$	8, 16, 32
Transmit/receive FIFO buffer access pointer register 2	RSCAN0CFPTR2	0000 0000$_H$	<RSCAN0_base> + 0EA4$_H$	8, 16, 32
Transmit/receive FIFO buffer access data field 0 register 2	RSCAN0CFDF02	0000 0000$_H$	<RSCAN0_base> + 0EA8$_H$	8, 16, 32
Transmit/receive FIFO buffer access data field 1 register 2	RSCAN0CFDF12	0000 0000$_H$	<RSCAN0_base> + 0EAC$_H$	8, 16, 32
Transmit/receive FIFO buffer access ID register 3	RSCAN0CFID3	0000 0000$_H$	<RSCAN0_base> + 0EB0$_H$	8, 16, 32
Transmit/receive FIFO buffer access pointer register 3	RSCAN0CFPTR3	0000 0000$_H$	<RSCAN0_base> + 0EB4$_H$	8, 16, 32
Transmit/receive FIFO buffer access data field 0 register 3	RSCAN0CFDF03	0000 0000$_H$	<RSCAN0_base> + 0EB8$_H$	8, 16, 32
Transmit/receive FIFO buffer access data field 1 register 3	RSCAN0CFDF13	0000 0000$_H$	<RSCAN0_base> + 0EBC$_H$	8, 16, 32
Transmit/receive FIFO buffer access ID register 4	RSCAN0CFID4	0000 0000$_H$	<RSCAN0_base> + 0EC0$_H$	8, 16, 32
Transmit/receive FIFO buffer access pointer register 4	RSCAN0CFPTR4	0000 0000$_H$	<RSCAN0_base> + 0EC4$_H$	8, 16, 32
Transmit/receive FIFO buffer access data field 0 register 4	RSCAN0CFDF04	0000 0000$_H$	<RSCAN0_base> + 0EC8$_H$	8, 16, 32
Transmit/receive FIFO buffer access data field 1 register 4	RSCAN0CFDF14	0000 0000$_H$	<RSCAN0_base> + 0ECC$_H$	8, 16, 32
Transmit/receive FIFO buffer access ID register 5	RSCAN0CFID5	0000 0000$_H$	<RSCAN0_base> + 0ED0$_H$	8, 16, 32
Transmit/receive FIFO buffer access pointer register 5	RSCAN0CFPTR5	0000 0000$_H$	<RSCAN0_base> + 0ED4$_H$	8, 16, 32
Transmit/receive FIFO buffer access data field 0 register 5	RSCAN0CFDF05	0000 0000$_H$	<RSCAN0_base> + 0ED8$_H$	8, 16, 32
Transmit/receive FIFO buffer access data field 1 register 5	RSCAN0CFDF15	0000 0000$_H$	<RSCAN0_base> + 0EDC$_H$	8, 16, 32
Transmit buffer ID register 0	RSCAN0TMID0	0000 0000$_H$	<RSCAN0_base> + 1000$_H$	8, 16, 32
Transmit buffer pointer register 0	RSCAN0TMPTR0	0000 0000$_H$	<RSCAN0_base> + 1004$_H$	8, 16, 32
Transmit buffer data field 0 register 0	RSCAN0TMDF00	0000 0000$_H$	<RSCAN0_base> + 1008$_H$	8, 16, 32
Transmit buffer data field 1 register 0	RSCAN0TMDF10	0000 0000$_H$	<RSCAN0_base> + 100C$_H$	8, 16, 32
Transmit buffer ID register 1	RSCAN0TMID1	0000 0000$_H$	<RSCAN0_base> + 1010$_H$	8, 16, 32
Transmit buffer pointer register 1	RSCAN0TMPTR1	0000 0000$_H$	<RSCAN0_base> + 1014$_H$	8, 16, 32
Transmit buffer data field 0 register 1	RSCAN0TMDF01	0000 0000$_H$	<RSCAN0_base> + 1018$_H$	8, 16, 32
Transmit buffer data field 1 register 1	RSCAN0TMDF11	0000 0000$_H$	<RSCAN0_base> + 101C$_H$	8, 16, 32
Transmit buffer ID register 2	RSCAN0TMID2	0000 0000$_H$	<RSCAN0_base> + 1020$_H$	8, 16, 32
Transmit buffer pointer register 2	RSCAN0TMPTR2	0000 0000$_H$	<RSCAN0_base> + 1024$_H$	8, 16, 32
Transmit buffer data field 0 register 2	RSCAN0TMDF02	0000 0000$_H$	<RSCAN0_base> + 1028$_H$	8, 16, 32
Transmit buffer data field 1 register 2	RSCAN0TMDF12	0000 0000$_H$	<RSCAN0_base> + 102C$_H$	8, 16, 32
Transmit buffer ID register 3	RSCAN0TMID3	0000 0000$_H$	<RSCAN0_base> + 1030$_H$	8, 16, 32
Transmit buffer pointer register 3	RSCAN0TMPTR3	0000 0000$_H$	<RSCAN0_base> + 1034$_H$	8, 16, 32
Transmit buffer data field 0 register 3	RSCAN0TMDF03	0000 0000$_H$	<RSCAN0_base> + 1038$_H$	8, 16, 32
Transmit buffer data field 1 register 3	RSCAN0TMDF13	0000 0000$_H$	<RSCAN0_base> + 103C$_H$	8, 16, 32

chanhvinh.com

Table 21.10 List of RS-CAN Module Registers (10/12)

Register Name	Symbol	After Reset	Address	Access Size
Transmit buffer ID register 4	RSCAN0TMID4	0000 0000$_H$	<RSCAN0_base> + 1040$_H$	8, 16, 32
Transmit buffer pointer register 4	RSCAN0TMPTR4	0000 0000$_H$	<RSCAN0_base> + 1044$_H$	8, 16, 32
Transmit buffer data field 0 register 4	RSCAN0TMDF04	0000 0000$_H$	<RSCAN0_base> + 1048$_H$	8, 16, 32
Transmit buffer data field 1 register 4	RSCAN0TMDF14	0000 0000$_H$	<RSCAN0_base> + 104C$_H$	8, 16, 32
Transmit buffer ID register 5	RSCAN0TMID5	0000 0000$_H$	<RSCAN0_base> + 1050$_H$	8, 16, 32
Transmit buffer pointer register 5	RSCAN0TMPTR5	0000 0000$_H$	<RSCAN0_base> + 1054$_H$	8, 16, 32
Transmit buffer data field 0 register 5	RSCAN0TMDF05	0000 0000$_H$	<RSCAN0_base> + 1058$_H$	8, 16, 32
Transmit buffer data field 1 register 5	RSCAN0TMDF15	0000 0000$_H$	<RSCAN0_base> + 105C$_H$	8, 16, 32
Transmit buffer ID register 6	RSCAN0TMID6	0000 0000$_H$	<RSCAN0_base> + 1060$_H$	8, 16, 32
Transmit buffer pointer register 6	RSCAN0TMPTR6	0000 0000$_H$	<RSCAN0_base> + 1064$_H$	8, 16, 32
Transmit buffer data field 0 register 6	RSCAN0TMDF06	0000 0000$_H$	<RSCAN0_base> + 1068$_H$	8, 16, 32
Transmit buffer data field 1 register 6	RSCAN0TMDF16	0000 0000$_H$	<RSCAN0_base> + 106C$_H$	8, 16, 32
Transmit buffer ID register 7	RSCAN0TMID7	0000 0000$_H$	<RSCAN0_base> + 1070$_H$	8, 16, 32
Transmit buffer pointer register 7	RSCAN0TMPTR7	0000 0000$_H$	<RSCAN0_base> + 1074$_H$	8, 16, 32
Transmit buffer data field 0 register 7	RSCAN0TMDF07	0000 0000$_H$	<RSCAN0_base> + 1078$_H$	8, 16, 32
Transmit buffer data field 1 register 7	RSCAN0TMDF17	0000 0000$_H$	<RSCAN0_base> + 107C$_H$	8, 16, 32
Transmit buffer ID register 8	RSCAN0TMID8	0000 0000$_H$	<RSCAN0_base> + 1080$_H$	8, 16, 32
Transmit buffer pointer register 8	RSCAN0TMPTR8	0000 0000$_H$	<RSCAN0_base> + 1084$_H$	8, 16, 32
Transmit buffer data field 0 register 8	RSCAN0TMDF08	0000 0000$_H$	<RSCAN0_base> + 1088$_H$	8, 16, 32
Transmit buffer data field 1 register 8	RSCAN0TMDF18	0000 0000$_H$	<RSCAN0_base> + 108C$_H$	8, 16, 32
Transmit buffer ID register 9	RSCAN0TMID9	0000 0000$_H$	<RSCAN0_base> + 1090$_H$	8, 16, 32
Transmit buffer pointer register 9	RSCAN0TMPTR9	0000 0000$_H$	<RSCAN0_base> + 1094$_H$	8, 16, 32
Transmit buffer data field 0 register 9	RSCAN0TMDF09	0000 0000$_H$	<RSCAN0_base> + 1098$_H$	8, 16, 32
Transmit buffer data field 1 register 9	RSCAN0TMDF19	0000 0000$_H$	<RSCAN0_base> + 109C$_H$	8, 16, 32
Transmit buffer ID register 10	RSCAN0TMID10	0000 0000$_H$	<RSCAN0_base> + 10A0$_H$	8, 16, 32
Transmit buffer pointer register 10	RSCAN0TMPTR10	0000 0000$_H$	<RSCAN0_base> + 10A4$_H$	8, 16, 32
Transmit buffer data field 0 register 10	RSCAN0TMDF010	0000 0000$_H$	<RSCAN0_base> + 10A8$_H$	8, 16, 32
Transmit buffer data field 1 register 10	RSCAN0TMDF110	0000 0000$_H$	<RSCAN0_base> + 10AC$_H$	8, 16, 32
Transmit buffer ID register 11	RSCAN0TMID11	0000 0000$_H$	<RSCAN0_base> + 10B0$_H$	8, 16, 32
Transmit buffer pointer register 11	RSCAN0TMPTR11	0000 0000$_H$	<RSCAN0_base> + 10B4$_H$	8, 16, 32
Transmit buffer data field 0 register 11	RSCAN0TMDF011	0000 0000$_H$	<RSCAN0_base> + 10B8$_H$	8, 16, 32
Transmit buffer data field 1 register 11	RSCAN0TMDF111	0000 0000$_H$	<RSCAN0_base> + 10BC$_H$	8, 16, 32
Transmit buffer ID register 12	RSCAN0TMID12	0000 0000$_H$	<RSCAN0_base> + 10C0$_H$	8, 16, 32
Transmit buffer pointer register 12	RSCAN0TMPTR12	0000 0000$_H$	<RSCAN0_base> + 10C4$_H$	8, 16, 32
Transmit buffer data field 0 register 12	RSCAN0TMDF012	0000 0000$_H$	<RSCAN0_base> + 10C8$_H$	8, 16, 32
Transmit buffer data field 1 register 12	RSCAN0TMDF112	0000 0000$_H$	<RSCAN0_base> + 10CC$_H$	8, 16, 32
Transmit buffer ID register 13	RSCAN0TMID13	0000 0000$_H$	<RSCAN0_base> + 10D0$_H$	8, 16, 32
Transmit buffer pointer register 13	RSCAN0TMPTR13	0000 0000$_H$	<RSCAN0_base> + 10D4$_H$	8, 16, 32
Transmit buffer data field 0 register 13	RSCAN0TMDF013	0000 0000$_H$	<RSCAN0_base> + 10D8$_H$	8, 16, 32
Transmit buffer data field 1 register 13	RSCAN0TMDF113	0000 0000$_H$	<RSCAN0_base> + 10DC$_H$	8, 16, 32
Transmit buffer ID register 14	RSCAN0TMID14	0000 0000$_H$	<RSCAN0_base> + 10E0$_H$	8, 16, 32
Transmit buffer pointer register 14	RSCAN0TMPTR14	0000 0000$_H$	<RSCAN0_base> + 10E4$_H$	8, 16, 32
Transmit buffer data field 0 register 14	RSCAN0TMDF014	0000 0000$_H$	<RSCAN0_base> + 10E8$_H$	8, 16, 32
Transmit buffer data field 1 register 14	RSCAN0TMDF114	0000 0000$_H$	<RSCAN0_base> + 10EC$_H$	8, 16, 32
Transmit buffer ID register 15	RSCAN0TMID15	0000 0000$_H$	<RSCAN0_base> + 10F0$_H$	8, 16, 32
Transmit buffer pointer register 15	RSCAN0TMPTR15	0000 0000$_H$	<RSCAN0_base> + 10F4$_H$	8, 16, 32
Transmit buffer data field 0 register 15	RSCAN0TMDF015	0000 0000$_H$	<RSCAN0_base> + 10F8$_H$	8, 16, 32

chanhvinh.com

Table 21.10 List of RS-CAN Module Registers (11/12)

Register Name	Symbol	After Reset	Address	Access Size
Transmit buffer data field 1 register 15	RSCAN0TMDF115	0000 0000$_H$	<RSCAN0_base> + 10FC$_H$	8, 16, 32
Transmit buffer ID register 16	RSCAN0TMID16	0000 0000$_H$	<RSCAN0_base> + 1100$_H$	8, 16, 32
Transmit buffer pointer register 16	RSCAN0TMPTR16	0000 0000$_H$	<RSCAN0_base> + 1104$_H$	8, 16, 32
Transmit buffer data field 0 register 16	RSCAN0TMDF016	0000 0000$_H$	<RSCAN0_base> + 1108$_H$	8, 16, 32
Transmit buffer data field 1 register 16	RSCAN0TMDF116	0000 0000$_H$	<RSCAN0_base> + 110C$_H$	8, 16, 32
Transmit buffer ID register 17	RSCAN0TMID17	0000 0000$_H$	<RSCAN0_base> + 1110$_H$	8, 16, 32
Transmit buffer pointer register 17	RSCAN0TMPTR17	0000 0000$_H$	<RSCAN0_base> + 1114$_H$	8, 16, 32
Transmit buffer data field 0 register 17	RSCAN0TMDF017	0000 0000$_H$	<RSCAN0_base> + 1118$_H$	8, 16, 32
Transmit buffer data field 1 register 17	RSCAN0TMDF117	0000 0000$_H$	<RSCAN0_base> + 111C$_H$	8, 16, 32
Transmit buffer ID register 18	RSCAN0TMID18	0000 0000$_H$	<RSCAN0_base> + 1120$_H$	8, 16, 32
Transmit buffer pointer register 18	RSCAN0TMPTR18	0000 0000$_H$	<RSCAN0_base> + 1124$_H$	8, 16, 32
Transmit buffer data field 0 register 18	RSCAN0TMDF018	0000 0000$_H$	<RSCAN0_base> + 1128$_H$	8, 16, 32
Transmit buffer data field 1 register 18	RSCAN0TMDF118	0000 0000$_H$	<RSCAN0_base> + 112C$_H$	8, 16, 32
Transmit buffer ID register 19	RSCAN0TMID19	0000 0000$_H$	<RSCAN0_base> + 1130$_H$	8, 16, 32
Transmit buffer pointer register 19	RSCAN0TMPTR19	0000 0000$_H$	<RSCAN0_base> + 1134$_H$	8, 16, 32
Transmit buffer data field 0 register 19	RSCAN0TMDF019	0000 0000$_H$	<RSCAN0_base> + 1138$_H$	8, 16, 32
Transmit buffer data field 1 register 19	RSCAN0TMDF119	0000 0000$_H$	<RSCAN0_base> + 113C$_H$	8, 16, 32
Transmit buffer ID register 20	RSCAN0TMID20	0000 0000$_H$	<RSCAN0_base> + 1140$_H$	8, 16, 32
Transmit buffer pointer register 20	RSCAN0TMPTR20	0000 0000$_H$	<RSCAN0_base> + 1144$_H$	8, 16, 32
Transmit buffer data field 0 register 20	RSCAN0TMDF020	0000 0000$_H$	<RSCAN0_base> + 1148$_H$	8, 16, 32
Transmit buffer data field 1 register 20	RSCAN0TMDF120	0000 0000$_H$	<RSCAN0_base> + 114C$_H$	8, 16, 32
Transmit buffer ID register 21	RSCAN0TMID21	0000 0000$_H$	<RSCAN0_base> + 1150$_H$	8, 16, 32
Transmit buffer pointer register 21	RSCAN0TMPTR21	0000 0000$_H$	<RSCAN0_base> + 1154$_H$	8, 16, 32
Transmit buffer data field 0 register 21	RSCAN0TMDF021	0000 0000$_H$	<RSCAN0_base> + 1158$_H$	8, 16, 32
Transmit buffer data field 1 register 21	RSCAN0TMDF121	0000 0000$_H$	<RSCAN0_base> + 115C$_H$	8, 16, 32
Transmit buffer ID register 22	RSCAN0TMID22	0000 0000$_H$	<RSCAN0_base> + 1160$_H$	8, 16, 32
Transmit buffer pointer register 22	RSCAN0TMPTR22	0000 0000$_H$	<RSCAN0_base> + 1164$_H$	8, 16, 32
Transmit buffer data field 0 register 22	RSCAN0TMDF022	0000 0000$_H$	<RSCAN0_base> + 1168$_H$	8, 16, 32
Transmit buffer data field 1 register 22	RSCAN0TMDF122	0000 0000$_H$	<RSCAN0_base> + 116C$_H$	8, 16, 32
Transmit buffer ID register 23	RSCAN0TMID23	0000 0000$_H$	<RSCAN0_base> + 1170$_H$	8, 16, 32
Transmit buffer pointer register 23	RSCAN0TMPTR23	0000 0000$_H$	<RSCAN0_base> + 1174$_H$	8, 16, 32
Transmit buffer data field 0 register 23	RSCAN0TMDF023	0000 0000$_H$	<RSCAN0_base> + 1178$_H$	8, 16, 32
Transmit buffer data field 1 register 23	RSCAN0TMDF123	0000 0000$_H$	<RSCAN0_base> + 117C$_H$	8, 16, 32
Transmit buffer ID register 24	RSCAN0TMID24	0000 0000$_H$	<RSCAN0_base> + 1180$_H$	8, 16, 32
Transmit buffer pointer register 24	RSCAN0TMPTR24	0000 0000$_H$	<RSCAN0_base> + 1184$_H$	8, 16, 32
Transmit buffer data field 0 register 24	RSCAN0TMDF024	0000 0000$_H$	<RSCAN0_base> + 1188$_H$	8, 16, 32
Transmit buffer data field 1 register 24	RSCAN0TMDF124	0000 0000$_H$	<RSCAN0_base> + 118C$_H$	8, 16, 32
Transmit buffer ID register 25	RSCAN0TMID25	0000 0000$_H$	<RSCAN0_base> + 1190$_H$	8, 16, 32
Transmit buffer pointer register 25	RSCAN0TMPTR25	0000 0000$_H$	<RSCAN0_base> + 1194$_H$	8, 16, 32
Transmit buffer data field 0 register 25	RSCAN0TMDF025	0000 0000$_H$	<RSCAN0_base> + 1198$_H$	8, 16, 32
Transmit buffer data field 1 register 25	RSCAN0TMDF125	0000 0000$_H$	<RSCAN0_base> + 119C$_H$	8, 16, 32
Transmit buffer ID register 26	RSCAN0TMID26	0000 0000$_H$	<RSCAN0_base> + 11A0$_H$	8, 16, 32
Transmit buffer pointer register 26	RSCAN0TMPTR26	0000 0000$_H$	<RSCAN0_base> + 11A4$_H$	8, 16, 32
Transmit buffer data field 0 register 26	RSCAN0TMDF026	0000 0000$_H$	<RSCAN0_base> + 11A8$_H$	8, 16, 32
Transmit buffer data field 1 register 26	RSCAN0TMDF126	0000 0000$_H$	<RSCAN0_base> + 11AC$_H$	8, 16, 32
Transmit buffer ID register 27	RSCAN0TMID27	0000 0000$_H$	<RSCAN0_base> + 11B0$_H$	8, 16, 32
Transmit buffer pointer register 27	RSCAN0TMPTR27	0000 0000$_H$	<RSCAN0_base> + 11B4$_H$	8, 16, 32

chanhvinh.com

Table 21.10 List of RS-CAN Module Registers (12/12)

Register Name	Symbol	After Reset	Address	Access Size
Transmit buffer data field 0 register 27	RSCAN0TMDF027	0000 0000$_H$	<RSCAN0_base> + 11B8$_H$	8, 16, 32
Transmit buffer data field 1 register 27	RSCAN0TMDF127	0000 0000$_H$	<RSCAN0_base> + 11BC$_H$	8, 16, 32
Transmit buffer ID register 28	RSCAN0TMID28	0000 0000$_H$	<RSCAN0_base> + 11C0$_H$	8, 16, 32
Transmit buffer pointer register 28	RSCAN0TMPTR28	0000 0000$_H$	<RSCAN0_base> + 11C4$_H$	8, 16, 32
Transmit buffer data field 0 register 28	RSCAN0TMDF028	0000 0000$_H$	<RSCAN0_base> + 11C8$_H$	8, 16, 32
Transmit buffer data field 1 register 28	RSCAN0TMDF128	0000 0000$_H$	<RSCAN0_base> + 11CC$_H$	8, 16, 32
Transmit buffer ID register 29	RSCAN0TMID29	0000 0000$_H$	<RSCAN0_base> + 11D0$_H$	8, 16, 32
Transmit buffer pointer register 29	RSCAN0TMPTR29	0000 0000$_H$	<RSCAN0_base> + 11D4$_H$	8, 16, 32
Transmit buffer data field 0 register 29	RSCAN0TMDF029	0000 0000$_H$	<RSCAN0_base> + 11D8$_H$	8, 16, 32
Transmit buffer data field 1 register 29	RSCAN0TMDF129	0000 0000$_H$	<RSCAN0_base> + 11DC$_H$	8, 16, 32
Transmit buffer ID register 30	RSCAN0TMID30	0000 0000$_H$	<RSCAN0_base> + 11E0$_H$	8, 16, 32
Transmit buffer pointer register 30	RSCAN0TMPTR30	0000 0000$_H$	<RSCAN0_base> + 11E4$_H$	8, 16, 32
Transmit buffer data field 0 register 30	RSCAN0TMDF030	0000 0000$_H$	<RSCAN0_base> + 11E8$_H$	8, 16, 32
Transmit buffer data field 1 register 30	RSCAN0TMDF130	0000 0000$_H$	<RSCAN0_base> + 11EC$_H$	8, 16, 32
Transmit buffer ID register 31	RSCAN0TMID31	0000 0000$_H$	<RSCAN0_base> + 11F0$_H$	8, 16, 32
Transmit buffer pointer register 31	RSCAN0TMPTR31	0000 0000$_H$	<RSCAN0_base> + 11F4$_H$	8, 16, 32
Transmit buffer data field 0 register 31	RSCAN0TMDF031	0000 0000$_H$	<RSCAN0_base> + 11F8$_H$	8, 16, 32
Transmit buffer data field 1 register 31	RSCAN0TMDF131	0000 0000$_H$	<RSCAN0_base> + 11FC$_H$	8, 16, 32
Transmit history access register 0	RSCAN0THLACC0	0000 0000$_H$	<RSCAN0_base> + 1800$_H$	8, 16, 32
Transmit history access register 1	RSCAN0THLACC1	0000 0000$_H$	<RSCAN0_base> + 1804$_H$	8, 16, 32

Table 21.11 Transmit Buffer p Allocated to Each Channel

	CANm
Transmit buffer p	Transmit buffer 16 × m + 0
	Transmit buffer 16 × m + 1
	Transmit buffer 16 × m + 2
	Transmit buffer 16 × m + 3
	Transmit buffer 16 × m + 4
	Transmit buffer 16 × m + 5
	Transmit buffer 16 × m + 6
	Transmit buffer 16 × m + 7
	Transmit buffer 16 × m + 8
	Transmit buffer 16 × m + 9
	Transmit buffer 16 × m + 10
	Transmit buffer 16 × m + 11
	Transmit buffer 16 × m + 12
	Transmit buffer 16 × m + 13
	Transmit buffer 16 × m + 14
	Transmit buffer 16 × m + 15

chanhvinh.com

Table 21.12 Transmit/Receive FIFO Buffer k Allocated to Each Channel

	CANm
Transmit/receive FIFO buffer k	Transmit/receive FIFO buffer 3 × m + 0
	Transmit/receive FIFO buffer 3 × m + 1
	Transmit/receive FIFO buffer 3 × m + 2

Table 21.13 Transmit Buffer p Linked to the Transmit/Receive FIFO Buffer by the Setting of Bits CFTML[3:0]

Setting of Bits CFTML[3:0]	Transmit Buffer p Linked to the Transmit/Receive FIFO Buffer
0000_B	Transmit buffer 16 × m + 0
0001_B	Transmit buffer 16 × m + 1
0010_B	Transmit buffer 16 × m + 2
0011_B	Transmit buffer 16 × m + 3
0100_B	Transmit buffer 16 × m + 4
0101_B	Transmit buffer 16 × m + 5
0110_B	Transmit buffer 16 × m + 6
0111_B	Transmit buffer 16 × m + 7
1000_B	Transmit buffer 16 × m + 8
1001_B	Transmit buffer 16 × m + 9
1010_B	Transmit buffer 16 × m + 10
1011_B	Transmit buffer 16 × m + 11
1100_B	Transmit buffer 16 × m + 12
1101_B	Transmit buffer 16 × m + 13
1110_B	Transmit buffer 16 × m + 14
1111_B	Transmit buffer 16 × m + 15

Table 21.14 Transmit Buffer p Allocated to the Transmit Queue of Each Channel

Setting of Bits TXQDC[3:0]	Transmit Buffer p Allocated to the Transmit Queue
0000_B	Setting prohibited
0001_B	Setting prohibited
0010_B	Transmit buffer 16 × m + 15 to 16 × m + 13
0011_B	Transmit buffer 16 × m + 15 to 16 × m + 12
0100_B	Transmit buffer 16 × m + 15 to 16 × m + 11
0101_B	Transmit buffer 16 × m + 15 to 16 × m + 10
0110_B	Transmit buffer 16 × m + 15 to 16 × m + 9
0111_B	Transmit buffer 16 × m + 15 to 16 × m + 8
1000_B	Transmit buffer 16 × m + 15 to 16 × m + 7
1001_B	Transmit buffer 16 × m + 15 to 16 × m + 6
1010_B	Transmit buffer 16 × m + 15 to 16 × m + 5
1011_B	Transmit buffer 16 × m + 15 to 16 × m + 4
1100_B	Transmit buffer 16 × m + 15 to 16 × m + 3
1101_B	Transmit buffer 16 × m + 15 to 16 × m + 2
1110_B	Transmit buffer 16 × m + 15 to 16 × m + 1
1111_B	Transmit buffer 16 × m + 15 to 16 × m + 0

chanhvinh.com

21.3.1 RSCAN0CmCFG — Channel Configuration Register (m = 0 or 1)

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0000_H + (m * 0010_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	SJW[1:0]		—	TSEG2[2:0]			TSEG1[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	BRP[9:0]									
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.15 RSCAN0CmCFG register contents

Bit Position	Bit Name	Function
31 to 26	Reserved	These bits are always read as 0. The write value should always be 0.
25, 24	SJW[1:0]	Resynchronization Jump Width Control b25 b24 0 0: 1 Tq 0 1: 2 Tq 1 0: 3 Tq 1 1: 4 Tq
23	Reserved	This bit is always read as 0. The write value should always be 0.
22 to 20	TSEG2[2:0]	Time Segment 2 Control b22 b21 b20 0 0 0: Setting prohibited 0 0 1: 2 Tq 0 1 0: 3 Tq 0 1 1: 4 Tq 1 0 0: 5 Tq 1 0 1: 6 Tq 1 1 0: 7 Tq 1 1 1: 8 Tq
19 to 16	TSEG1[3:0]	Time Segment 1 Control b19 b18 b17 b16 0 0 0 0: Setting prohibited 0 0 0 1: Setting prohibited 0 0 1 0: Setting prohibited 0 0 1 1: 4 Tq 0 1 0 0: 5 Tq 0 1 0 1: 6 Tq 0 1 1 0: 7 Tq 0 1 1 1: 8 Tq 1 0 0 0: 9 Tq 1 0 0 1: 10 Tq 1 0 1 0: 11 Tq 1 0 1 1: 12 Tq 1 1 0 0: 13 Tq 1 1 0 1: 14 Tq 1 1 1 0: 15 Tq 1 1 1 1: 16 Tq
15 to 10	Reserved	These bits are always read as 0. The write value should always be 0.
9 to 0	BRP[9:0]	Prescaler Division Ratio Set When these bits are set to P (0 to 1023), the baud rate prescaler divides fCAN by P + 1.

chanhvinh.com

Modify the RSCAN0CmCFG register in channel reset mode or channel halt mode. Set this register before requesting a transition to channel communication mode or channel wait mode. For a description of the bit timing parameters and settings, see Section 21.10.1, Initial Settings.

SJW[1:0] Bits

These bits are used to specify a Tq value for the resynchronization jump width. Allowed values are 1 Tq to 4 Tq, inclusive.
Set a value less than or equal to the value of the TSEG2 bits.

TSEG2[2:0] Bits

These bits are used to specify a Tq value for the length of phase buffer segment 2 (PHASE_SEG2). Allowed values are 2 Tq to 8 Tq, inclusive.
Set a value smaller than the value of the TSEG1 bits.

TSEG1[3:0] Bits

These bits are used to specify a Tq value for the total length of the propagation time segment (PROP_SEG) and phase buffer segment 1 (PHASE_SEG1).
Allowed values are 4 Tq to 16 Tq, inclusive.

BRP[9:0] Bits

The CANmTq (fCANTQm) clock is calculated by dividing the CAN clock (fCAN) by the baud rate prescaler, ((BRP[9:0]) + 1). One clock cycle of the CANmTq clock is 1 Time Quantum (Tq).

chanhvinh.com

21.3.2 RSCAN0CmCTR — Channel Control Register (m = 0 or 1)

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0004_H + (m * 0010_H)

Initial value: 0000 0005_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	CTMS[1:0]		CTME	ERRD	BOM[1:0]		—	—	—	—	TAIE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ALIE	BLIE	OLIE	BORIE	BOEIE	EPIE	EWIE	BEIE	—	—	—	—	RTBO	CSLPR	CHMDC[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 21.16 RSCAN0CmCTR register contents (1/2)

Bit Position	Bit Name	Function
31 to 27	Reserved	These bits are always read as 0. The write value should always be 0.
26, 25	CTMS[1:0]	Communication Test Mode Select b26 b25 0 0: Standard test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (external loopback mode) 1 1: Self-test mode 1 (internal loopback mode)
24	CTME	Communication Test Mode Enable 0: Communication test mode is disabled. 1: Communication test mode is enabled.
23	ERRD	Error Display Mode Select 0: Error flags are displayed only for the first error information after bits 14 to 8 in RSCAN0CmERFL are all cleared. 1: Error flags for all error information are displayed.
22, 21	BOM[1:0]	Bus Off Recovery Mode Select b22 b21 0 0: ISO11898-1 compliant 0 1: Entry to channel halt mode automatically at bus-off entry 1 0: Entry to channel halt mode automatically at bus-off end 1 1: Entry to channel halt mode (during bus-off recovery period) by program request
20 to 17	Reserved	These bits are always read as 0. The write value should always be 0.
16	TAIE	Transmit Abort Interrupt Enable 0: Transmit abort interrupt is disabled. 1: Transmit abort interrupt is enabled.
15	ALIE	Arbitration Lost Interrupt Enable 0: Arbitration lost interrupt is disabled. 1: Arbitration lost interrupt is enabled.
14	BLIE	Bus Lock Interrupt Enable 0: Bus lock interrupt is disabled. 1: Bus lock interrupt is enabled.
13	OLIE	Overload Frame Transmit Interrupt Enable 0: Overload frame transmit interrupt is disabled. 1: Overload frame transmit interrupt is enabled.
12	BORIE	Bus Off Recovery Interrupt Enable 0: Bus off recovery interrupt is disabled. 1: Bus off recovery interrupt is enabled.

chanhvinh.com

Table 21.16 RSCAN0CmCTR register contents (2/2)

Bit Position	Bit Name	Function
11	BOEIE	Bus Off Entry Interrupt Enable 0: Bus off entry interrupt is disabled. 1: Bus off entry interrupt is enabled.
10	EPIE	Error Passive Interrupt Enable 0: Error passive interrupt is disabled. 1: Error passive interrupt is enabled.
9	EWIE	Error Warning Interrupt Enable 0: Error warning interrupt is disabled. 1: Error warning interrupt is enabled.
8	BEIE	Bus Error Interrupt Enable 0: Bus error interrupt is disabled. 1: Bus error interrupt is enabled.
7 to 4	Reserved	These bits are always read as 0. The write value should always be 0.
3	RTBO	Forcible Return from Bus-off When this bit is set to 1, forcible return from the bus off state is made. This bit is always read as 0.
2	CSLPR	Channel Stop Mode 0: Other than channel stop mode 1: Channel stop mode
1, 0	CHMDC[1:0]	Mode Select b1 b0 0 0: Channel communication mode 0 1: Channel reset mode 1 0: Channel halt mode 1 1: Setting prohibited

CTMS[1:0] Bits

These bits are used to select a communication test mode. Modify these bits in channel halt mode only. These bits are set to 0 in channel reset mode.

CTME Bit

Setting this bit to 1 enables communication test mode. Modify these bits in channel halt mode. This bit is set to 0 in channel reset mode.

ERRD Bit

This bit is used to control the display mode of bits 14 to 8 in the RSCAN0CmERFL register.
When this bit is clear to 0, only the flags of the first error are set to 1. If two or more errors occur in the first error, all the flags of the detected errors are set to 1.

When this bit is set to 1, all the flags of errors that have occurred are set to 1 regardless of the error occurrence order.

Modify this bit only in channel reset mode or channel halt mode.

chanhvinh.com

BOM[1:0] Bits

These bits are used to select the bus off recovery mode of the RS-CAN module.

When the BOM[1:0] bits are set to 00_B, return from the bus off state to the error active state is compliant with the CAN specifications. That is, the RS-CAN module reenters the CAN communication (error active state) after 11 consecutive recessive bits are detected 128 times. A bus off recovery interrupt request is generated at the time of return from the bus off state. Even if the CHMDC[1:0] bits are set to 10_B (channel halt mode) before recessive bits are detected 128 times, the RS-CAN module does not transition to channel halt mode until recessive bits are detected 128 times.

When the RS-CAN module reaches the bus off state when the BOM[1:0] bits are set to 01_B, the CHMDC[1:0] bits in the RSCAN0CmCTR register (m = 0 or 1) are set to 10_B and the RS-CAN module transitions to channel halt mode. No bus off recovery interrupt request is generated at the time of return from the bus off state and the TEC[7:0] and REC[7:0] bits in the RSCAN0CmSTS register are cleared to 00_H.

When the RS-CAN module reaches the bus off state when the BOM[1:0] bits are set to 10_B, the CHMDC[1:0] bits are set to 10_B and the RS-CAN module transitions to channel halt mode after return from the bus off state (11 consecutive recessive bits are detected 128 times). A bus off recovery interrupt request is generated at the time of return from the bus off state and the TEC[7:0] and REC[7:0] bits are cleared to 00_H.

When the BOM[1:0] bits are set to 11_B and the CHMDC[1:0] bits are set to 10_B while the RS-CAN module is in the bus off state, the RS-CAN module transitions to channel halt mode. No bus off recovery interrupt request is generated at the time of return from the bus off state and the TEC[7:0] and REC[7:0] bits are cleared to 00_H. However, if 11 consecutive recessive bits are detected 128 times and the RS-CAN module has recovered to the error active state from the bus off state before the CHMDC[1:0] bits are set to 10_B, a bus off recovery interrupt request is generated.

If the CPU requests a transition to channel reset mode at the same time as the RS-CAN module transition to channel halt mode (at bus off entry when the BOM[1:0] bits are 01_B or at bus off end when the BOM[1:0] bits are 10_B), the CPU's request takes precedence. Modify the BOM bits only in channel reset mode.

TAIE Bit

When transmit abort of the transmit buffer is completed with the TAIE bit set to 1, an interrupt request is generated. Modify this bit only in channel reset mode.

ALIE Bit

When the ALF flag in the RSCAN0CmERFL register is set to 1 with the ALIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BLIE Bit

When the BLF flag in the RSCAN0CmERFL register is set to 1 with the BLIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

OLIE Bit

When the OVLF flag in the RSCAN0CmERFL register is set to 1 with the OLIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

chanhvinh.com

BORIE Bit

When the BORF flag in the RSCAN0CmERFL register is set to 1 with the BORIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BOEIE Bit

When the BOEF flag in the RSCAN0CmERFL register is set to 1 with the BOEIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

EPIE Bit

When the EPF flag in the RSCAN0CmERFL register is set to 1 with the EPIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

EWIE Bit

When the EWF flag in the RSCAN0CmERFL register is set to 1 with the EWIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BEIE Bit

When the BEF flag in the RSCAN0CmERFL register is set to 1 with the BEIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

RTBO Bit

Setting this bit to 1 in the bus off state forcibly returns the state from the bus off state to the error active state. This bit is automatically cleared to 0. Setting this bit to 1 clears the TEC[7:0] and REC[7:0] bits in the RSCAN0CmSTS register to 00_H and also clears the BOSTS flag in the RSCAN0CmSTS register to 0 (not in bus off state). The other registers remain unchanged. No bus off recovery interrupt request is generated upon return from the bus off state in this case. Use this bit only when the BOM[1:0] bits in the RSCAN0CmCTR register are 00_B (ISO11898-1 compliant).

A delay of up to 1 CAN bit time occurs after the RTBO bit is set to 1 until the RSCAN module transitions to the error active state. Set this bit to 1 in channel communication mode.

CSLPR Bit

Setting this bit to 1 places the channel into channel stop mode.

Clearing this bit to 0 makes the channel exit channel stop mode.

Modify this bit from 0 to 1 only in channel reset mode.

CHMDC[1:0] Bits

These bits are used to select a channel mode (channel communication mode, channel reset mode, or channel halt mode). For details, see Section 21.5.2, Channel Modes. Setting the CSLPR bit to 1 in channel reset mode allows transition to channel stop mode. Do not set the CHMDC[1:0] bits to 11_B. When the CAN module has automatically transitioned to channel halt mode based on the setting of the BOM[1:0] bits, the CHMDC[1:0] bits automatically become 10_B.

chanhvinh.com

21.3.3 RSCAN0CmSTS — Channel Status Register (m = 0 or 1)

Access: Can be read in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0008_H + (m * 0010_H)

Initial value: 0000 0005_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TEC[7:0]								REC[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	COMSTS	RECSTS	TRMSTS	BOSTS	EPSTS	CSLPSTS	CHLTSTS	CRSTSTS
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.17 RSCAN0CmSTS register contents

Bit Position	Bit Name	Function
31 to 24	TEC[7:0]	The transmit error counter (TEC) can be read.
23 to 16	REC[7:0]	The receive error counter (REC) can be read.
15 to 8	Reserved	These bits are always read as 0.
7	COMSTS	Communication Status Flag 0: Communication is not ready. 1: Communication is ready.
6	RECSTS	Receive Status Flag 0: Bus idle, in transmission or bus off state 1: In reception
5	TRMSTS	Transmit Status Flag 0: Bus idle or in reception 1: In transmission or bus off state
4	BOSTS	Bus Off Status Flag 0: Not in bus off state 1: In bus off state
3	EPSTS	Error Passive Status Flag 0: Not in error passive state 1: In error passive state
2	CSLPSTS	Channel Stop Status Flag 0: Not in channel stop mode 1: In channel stop mode
1	CHLTSTS	Channel Halt Status Flag 0: Not in channel halt mode 1: In channel halt mode
0	CRSTSTS	Channel Reset Status Flag 0: Not in channel reset mode 1: In channel reset mode

TEC[7:0] Bits

These bits contain the transmit error counter value. For transmit error counter increment/decrement conditions, see the CAN specification (ISO11898-1).

These bits are cleared to 0 in channel reset mode.

chanhvinh.com

REC[7:0] Bits

These bits contain the receive error counter value. For receive error counter increment/decrement conditions, see the CAN specifications (ISO11898-1).

These bits are cleared to 0 in channel reset mode.

COMSTS Flag

This bit indicates that communication is ready.

This flag becomes 1 when the CAN module has detected 11 consecutive recessive bits after it has transitioned from channel reset mode or channel halt mode to channel communication mode. This flag is cleared to 0 in channel reset mode or channel halt mode.

RECSTS Flag

This flag is set to 1 when reception has started, and is cleared to 0 when the bus has become idle or transmission has started.

TRMSTS Flag

This flag is set to 1 when transmission has started, and is cleared to 0 when the bus has become idle or reception has started. This flag remains 1 in the bus off state.

BOSTS Flag

This flag is set to 1 when the bus off state (TEC[7:0] > 255) is entered. It is cleared to 0 when the CAN module has exited the bus off state.

EPSTS Flag

This flag is set to 1 when the RS-CAN module has entered the error passive state ((128 ≤ TEC[7:0] ≤ 255) or (128 ≤ REC[7:0])), It is cleared to 0 when the RS-CAN module has exited the error passive state or has entered channel reset mode.

CSLPSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel stop mode, and is cleared to 0 when the CAN module has returned from channel stop mode.

CHLTSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel halt mode, and is cleared to 0 when the CAN module has returned from channel halt mode.

CRSTSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel reset mode, and is cleared to 0 when the CAN module has transitioned to channel communication mode or channel halt mode. This flag remains 1 when the CAN module transitions from channel reset mode to channel stop mode.

chanhvinh.com

21.3.4 RSCAN0CmERFL — Channel Error Flag Register (m = 0 or 1)

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + $000C_H$ + (m * 0010_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	CRCREG[14:0]														
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ADERR	B0ERR	B1ERR	CERR	AERR	FERR	SERR	ALF	BLF	OVLF	BORF	BOEF	EPF	EWF	BEF
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 21.18 RSCAN0CmERFL register contents (1/2)

Bit Position	Bit Name	Function
31	Reserved	This bit is always read as 0. The write value should always be 0.
30 to 16	CRCREG[14:0]	CRC Calculation Data A CRC value calculated based on the transmit message or receive message is indicated.
15	Reserved	This bit is always read as 0. The write value should always be 0.
14	ADERR	ACK Delimiter Error Flag 0: No ACK delimiter error is detected. 1: ACK delimiter error is detected.
13	B0ERR	Dominant Bit Error Flag 0: No dominant bit error is detected. 1: Dominant bit error is detected.
12	B1ERR	Recessive Bit Error Flag 0: No recessive bit error is detected. 1: Recessive bit error is detected.
11	CERR	CRC Error Flag 0: No CRC error is detected. 1: CRC error is detected.
10	AERR	ACK Error Flag 0: No ACK error is detected. 1: ACK error is detected.
9	FERR	Form Error Flag 0: No form error is detected. 1: Form error is detected.
8	SERR	Stuff Error Flag 0: No stuff error is detected. 1: Stuff error is detected.
7	ALF	Arbitration-lost Flag 0: No arbitration-lost is detected. 1: Arbitration-lost is detected.
6	BLF	Dominant Lock Flag 0: No dominant lock is detected. 1: Dominant lock is detected.

chanhvinh.com

Table 21.18 RSCAN0CmERFL register contents (2/2)

Bit Position	Bit Name	Function
5	OVLF	Overload Flag 0: No overload is detected. 1: Overload is detected.
4	BORF	Bus Off Recovery Flag 0: No bus off recovery is detected. 1: Bus off recovery is detected.
3	BOEF	Bus Off Entry Flag 0: No bus off entry is detected. 1: Bus off entry is detected.
2	EPF	Error Passive Flag 0: No error passive is detected. 1: Error passive is detected.
1	EWF	Error Warning Flag 0: No error warning is detected. 1: Error warning is detected.
0	BEF	Protocol Error Flag 0: No protocol error is detected. 1: Protocol error is detected.

See the CAN specification (ISO11898-1) for a description of error occurrence conditions. To clear each flag of this register, the program must write a 0 to the corresponding bit. These flags cannot be set to 1 by the program. If any of these error occurs at the same time that the program writes 0 to the flag, the flag is still set to 1. The channel reset mode transition clears all of these flags to 0. If the ERRD bit in the RSCAN0CmCTR register is set to 0 (ie, only the flags from the first error event are displayed) and an error related to bits 14 to 8 of RSCAN0CmERFL is detected, the flag bits are only set by the error event if bits 14 to 8 were all 0 at the time the error occurred.

CRCREG[14:0] Flag

When the CTME bit in the RSCAN0CmCTR register is set to 1 (communication test mode is enabled), the CRC value calculated based on the transmit or receive message can be read. When the CTME bit is set to 0 (communication test mode is disabled), these bits are always read as 0.

ADERR Flag

This flag is set to 1 when a form error has been detected in the ACK delimiter during transmission.

B0ERR Flag

This flag is set to 1 when a recessive bit has been detected though a dominant bit was transmitted.

B1ERR Flag

This flag is set to 1 when a dominant bit has been detected though a recessive bit was transmitted.

CERR Flag

This flag is set to 1 when a CRC error has been detected.

AERR Flag

This flag is set to 1 when an ACK error has been detected.

chanhvinh.com

FERR Flag

This flag is set to 1 when a form error has been detected.

SERR Flag

This flag is set to 1 when a stuff error has been detected.

ALF Flag

This flag is set to 1 when an arbitration-lost has been detected.

BLF Flag

This flag is set to 1 when 32 consecutive dominant bits have been detected on the CAN bus in channel communication mode. After that, dominant lock can be detected again if any of the following conditions is met.

- a recessive bit is detected after the BLF bit has been cleared from 1 to 0.
- the CAN module transitions to channel reset mode and returns to channel communication mode after the BLF bit has been cleared from 1 to 0.

OVLF Flag

This flag is set to 1 when the overload frame transmit condition has been detected when performing reception or transmission.

BORF Flag

This flag is set to 1 when 11 consecutive recessive bits have been detected 128 times and the CAN module returns from the bus off state. However, this flag is not set to 1 if the CAN module returns from the bus off state in any of the following ways before 11 consecutive recessive bits are detected 128 times.

- The CHMDC[1:0] bits in the RSCAN0CmCTR register are set to 01_B (channel reset mode).
- The RTBO bit in the RSCAN0CmCTR register is set to 1 (forcible return from the bus off state is made).
- The BOM[1:0] bits in the RSCAN0CmCTR register are set to 01_B (transition to channel halt mode at bus off entry).
- The CHMDC[1:0] bits in the RSCAN0CmCTR register are set to 10_B (channel halt mode) before 11 consecutive recessive bits are detected 128 times with the BOM[1:0] bits set to 11_B (transition to channel halt mode upon a request from the program during bus off).

BOEF Flag

This flag is set to 1 when the bus off state is reached (TEC[7:0] value > 255). This flag is also set to 1 if the bus off state is reached when the BOM[1:0] bits in the RSCAN0CmCTR register (m = 0 or 1) set to 01_B (transition to channel halt mode at bus off entry).

chanhvinh.com

EPF Flag

This flag becomes 1 when the error passive state is reached (REC[7:0] or TEC[7:0] value > 127).

This flag becomes 1 only when the REC[7:0] or TEC[7:0] value first exceeds 127. Therefore, if the program writes 0 to this flag while the value of REC[7:0] or TEC[7:0] remains over 127, this bit is not set to 1 until both REC [7:0] and TEC[7:0] values become 127 or less and then the REC[7:0] or TEC[7:0] value exceeds 127 again.

EWF Flag

This flag is set to 1 only when the REC[7:0] or TEC[7:0] value first exceeds 95. Therefore, if the program writes 0 to this flag while the value of REC[7:0] or TEC[7:0] remains over 95, this bit is not set to 1 until both REC [7:0] and TEC[7:0] values become 95 or less and then the REC[7:0] or TEC[7:0] value exceeds 95 again.

BEF Flag

This flag is set to 1 when any one of the ADERR, B0ERR, B1ERR, CERR, AERR, FERR, and SERR flags in the RSCAN0CmERFL register is set to 1.

chanhvinh.com

21.3.5 RSCAN0GCFG — Global Configuration Register

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0084_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ITRCP															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSBTCS[2:0]			TSSS	TSP[3:0]				—	—	—	DCS	MME	DRE	DCE	TPRI
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 21.19 RSCAN0GCFG register contents (1/2)

Bit Position	Bit Name	Function
31 to 16	ITRCP[15:0]	Interval Timer Prescaler Set When these bits are set to M, the pclk is divided by M. Setting 0000_H is prohibited when the interval timer is in use.
15 to 13	TSBTCS[2:0]	Timestamp Clock Source Select b15 b14 b13 0 0 0: Channel 0 bit time clock 0 0 1: Channel 1 bit time clock 0 1 0: Setting prohibited 0 1 1: Setting prohibited 1 0 0: Setting prohibited 1 0 1: Setting prohibited 1 1 0: Setting prohibited 1 1 1: Setting prohibited
12	TSSS	Timestamp Source Select 0: pclk/2*1 1: Bit time clock
11 to 8	TSP[3:0]	Timestamp Clock Source Division b11 b10 b9 b8 0 0 0 0: Not divided 0 0 0 1: Divided by 2 0 0 1 0: Divided by 4 0 0 1 1: Divided by 8 0 1 0 0: Divided by 16 0 1 0 1: Divided by 32 0 1 1 0: Divided by 64 0 1 1 1: Divided by 128 1 0 0 0: Divided by 256 1 0 0 1: Divided by 512 1 0 1 0: Divided by 1024 1 0 1 1: Divided by 2048 1 1 0 0: Divided by 4096 1 1 0 1: Divided by 8192 1 1 1 0: Divided by 16384 1 1 1 1: Divided by 32768
7 to 5	Reserved	These bits are always read as 0. The write value should always be 0.
4	DCS	CAN Clock Source Select*2 0: clkc 1: clk_xincan

chanhvinh.com

Table 21.19 RSCAN0GCFG register contents (2/2)

Bit Position	Bit Name	Function
3	MME	Mirror Function Enable 0: Mirror function is disabled. 1: Mirror function is enabled.
2	DRE	DLC Replacement Enable 0: DLC replacement is disabled. 1: DLC replacement is enabled.
1	DCE	DLC Check Enable 0: DLC check is disabled. 1: DLC check is enabled.
0	TPRI	Transmit Priority Select 0: ID priority 1: Transmit buffer number priority

Note 1. When specifying CKSCLK_ICANOSC as the timestamp counter count source, set bits TSBTCS[2:0] to 000_B.

Note 2. For the setting of the CAN clock frequency, see Table 21.6, Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in this LSI.

Modify the RSCAN0GCFG register only in global reset mode.

ITRCP[15:0] Bits

These bits are used to set a clock source division value of the interval timer for FIFO buffers. Refer to Section 21.7.3.1, Interval Transmission Function

TSBTCS[2:0] Bits

When the TSSS bit is 1, these bits are used to select the channel of the bit time clock that will be the clock source of the timestamp counter.

TSSS Bit

This bit is used to select a clock source of the timestamp counter.

TSP[3:0] Bits

A clock obtained by dividing the clock source selected with the TSBTCS[2:0] bits and TSSS bit according to the TSP[3:0] bits is used as the timestamp counter count source.

DCS Bit

When this bit is set to 0, clkc is used as the clock source of the CAN clock (f_{CAN}).

When this bit is set to 1, clk_xincan is used as the clock source of the CAN clock (f_{CAN}).

For the setting of the CAN clock frequency, see Table 21.6, Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in this LSI.

MME Bit

Setting this bit to 1 makes the mirror function available.

chanhvinh.com

DRE Bit

When the DRE bit is set to 1, the DLC value of the receive rule is stored in the buffer instead of the DLC value of the received message after the DLC value has passed through the DLC filter. In this case, a value of 00_H is stored in each data byte beyond the first n bytes, where n is the DLC value of the receive rule.

The DLC replacement function is only available when the DCE bit is set to 1 (DLC check is enabled).

DCE Bit

Setting this bit to 1 makes the DLC check function available. When disabling the DLC check function, set the GAFLDLC[3:0] bits in the RSCAN0GAFLP0j register to 0000_B before clearing the DCE bit in the RSCAN0GCFG register to 0.

TPRI Bit

This bit is used to set the transmit priority.

When this bit is set to 0, ID priority is selected and the transmit priority complies with the CAN bus arbitration rule (ISO11898-1 specifications). When this bit is set to 1, transmit buffer number priority is selected and the lowest transmit buffer number of those with pending messages has the highest priority.

While the transmit queue is in use, this bit should be set to 0.

chanhvinh.com

21.3.6 RSCAN0GCTR — Global Control Register

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0088_H

Initial value: $0000\ 0005_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSRST
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	THLEIE	MEIE	DEIE	—	—	—	—	—	GSLPR	GMDC[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 21.20 RSCAN0GCTR register contents

Bit Position	Bit Name	Function
31 to 17	Reserved	These bits are always read as 0. The write value should always be 0.
16	TSRST	Timestamp Counter Reset Setting the TSRST bit to 1 resets the timestamp counter. This bit is always read as 0.
15 to 11	Reserved	These bits are always read as 0. The write value should always be 0.
10	THLEIE	Transmit History Buffer Overflow Interrupt Enable 0: Transmit history buffer overflow interrupt is disabled. 1: Transmit history buffer overflow interrupt is enabled.
9	MEIE	FIFO Message Lost Interrupt Enable 0: FIFO message lost interrupt is disabled. 1: FIFO message lost interrupt is enabled.
8	DEIE	DLC Error Interrupt Enable 0: DLC error interrupt is disabled. 1: DLC error interrupt is enabled.
7 to 3	Reserved	These bits are always read as 0. The write value should always be 0.
2	GSLPR	Global Stop Mode 0: Other than global stop mode 1: Global stop mode
1, 0	GMDC[1:0]	Global Mode Select b1 b0 0 0: Global operating mode 0 1: Global reset mode 1 0: Global test mode 1 1: Setting prohibited

TSRST Bit

This bit is used to reset the timestamp counter. When this bit is set to 1, the RSCAN0GTSC register is cleared to 0000_H.

THLEIE Bit

When the THLEIE bit is set to 1 and the THLES flag in the RSCAN0GERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

chanhvinh.com

MEIE Bit

When the MEIE bit is set to 1 and the MES flag in the RSCAN0GERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

DEIE Bit

When the DEIE bit is set to 1 and the DEF flag in the RSCAN0GERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

GSLPR Bit

When the RSCAN module is in global reset mode, setting this bit to 1 places the RSCAN module into global stop mode.

Clearing this bit to 0 makes the RSCAN module leave from global stop mode.

Modify this bit only in global reset mode.

GMDC[1:0] Bits

These bits are used to select the mode of entire RS-CAN module (global operating mode, global reset mode, or global test mode). For details, see Section 21.5.1, Global Modes. Setting the GSLPR bit to 1 when in global reset mode places the RS-CAN module into global stop mode.

chanhvinh.com

21.3.7 RSCAN0GSTS — Global Status Register

Access: Can be read in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + $008C_H$

Initial value: $0000\ 000D_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	GRAMINIT	GSLPSTS	GHLTSTS	GRSTSTS
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.21 RSCAN0GSTS register contents

Bit Position	Bit Name	Function
31 to 4	Reserved	These bits are always read as 0.
3	GRAMINIT	CAN RAM Initialization Status Flag 0: CAN RAM initialization is completed. 1: CAN RAM initialization is ongoing.
2	GSLPSTS	Global Stop Status Flag 0: Not in global stop mode 1: In global stop mode
1	GHLTSTS	Global Test Status Flag 0: Not in global test mode 1: In global test mode
0	GRSTSTS	Global Reset Status Flag 0: Not in global reset mode 1: In global reset mode

GRAMINIT Flag

This flag indicates the initialization status of the CAN RAM.

This flag is set to 1 after this LSI has been reset, and is cleared to 0 when CAN RAM initialization is completed.

GSLPSTS Flag

This flag is set to 1 when the CAN module has transitioned to global stop mode, and is cleared to 0 when the CAN module has returned from global stop mode.

GHLTSTS Flag

This flag is set to 1 when the CAN module has transitioned to global test mode, and is cleared to 0 when the CAN module has exited global test mode.

chanhvinh.com

GRSTSTS Flag

This flag is set to 1 when the CAN module has transitioned to global reset mode, and is cleared to 0 when the CAN module has exited global reset mode. This flag remains 1 even when the CAN module has transitioned from global reset mode to global stop mode.

chanhvinh.com

21.3.8 RSCAN0GERFL — Global Error Flag Register

Access: Can be read in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0090_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	THLES	MES	DEF
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 21.22 RSCAN0GERFL register contents

Bit Position	Bit Name	Function
31 to 29	Reserved	The read value is undefined. The write value should always be 0.
2	THLES	Transmit History Buffer Overflow Status Flag 0: No transmit history buffer overflow has occurred. 1: A transmit history buffer overflow has occurred.
1	MES	FIFO Message Lost Status Flag 0: No FIFO message lost error has occurred. 1: A FIFO message lost error has occurred.
0	DEF	DLC Error Flag 0: No DLC error has occurred. 1: A DLC error has occurred.

All flags in the RSCAN0GERFL register are cleared to 0 in global reset mode.

THLES Flag

The THLES flag is set to 1 when any one of the THLELT flags in the RSCAN0THLSTSm register (m = 0 or 1) is set to 1.

This flag is cleared to 0 when the THLELT flags of all channels are set to 0.

MES Flag

The MES flag is set to 1 when any one of the RFMLT flags in the RSCAN0RFSTSx register (x = 0 to 7) or the CFMLT flags in the RSCAN0CFSTSk register (k = 0 to 5) is set to 1.

This flag is cleared to 0 when all RFMLT flags and CFMLT flags are set to 0.

DEF Flag

The DEF flag is set to 1 when an error has been detected during the DLC check. The program can clear this flag by writing 0 to this bit.

chanhvinh.com

21.3.9 RSCAN0GTINTSTS0 — Global TX Interrupt Status Register 0

Access: Can be read in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0460_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	THIF1	CFTIF1	TQIF1	TAIF1	TSIF1	—	—	—	THIF0	CFTIF0	TQIF0	TAIF0	TSIF0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R*1	R*1	R*1	R*1	R*1	R	R	R	R*1	R*1	R*1	R*1	R*1

Note 1. This bit is automatically cleared in the global reset or channel reset mode.

Table 21.23 RSCAN0GTINTSTS0 register contents

Bit Position	Bit Name	Function
31 to 13	Reserved	These bits are always read as 0. The write value should always be 0.
12	THIF1	Channel 1 transmit history interrupt status flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
11	CFTIF1	Channel 1 transmit/receive FIFO transmit/gateway mode interrupt status flag 0: Transmit/receive FIFO transmit/gateway mode interrupt is not requested. 1: Transmit/receive FIFO transmit/gateway mode interrupt is requested.
10	TQIF1	Channel 1 transmit queue interrupt status flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
9	TAIF1	Channel 1 transmit buffer abort interrupt status flag 0: Transmit buffer transmit abort interrupt is not requested. 1: Transmit buffer transmit abort interrupt is requested.
8	TSIF1	Channel 1 transmit buffer interrupt status flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.
7 to 5	Reserved	These bits are always read as 0. The write value should always be 0.
4	THIF0	Channel 0 transmit history interrupt status flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
3	CFTIF0	Channel 0 transmit/receive FIFO transmit/gateway mode interrupt status flag 0: Transmit/receive FIFO transmit/gateway mode interrupt is not requested. 1: Transmit/receive FIFO transmit/gateway mode interrupt is requested.
2	TQIF0	Channel 0 transmit queue interrupt status flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
1	TAIF0	Channel 0 transmit buffer abort interrupt status flag 0: Transmit buffer transmit abort interrupt is not requested. 1: Transmit buffer transmit abort interrupt is requested.
0	TSIF0	Channel 0 transmit buffer interrupt status flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.

chanhvinh.com

TSIFm Bits

The TSIFm bit is set to 1 when the TMIE bit in the RSCAN0TMIECy register is set to 1 (transmit buffer interrupt enabled) and the TMTRF[1:0] flags in the RSCAN0TMSTSp register are set to 10_B (transmit completed without abort request) or 11_B (transmit completed with abort request).

When the TMTRF[1:0] flags are cleared to 00_B under the condition that the TSIFm bit can be set to 1, this flag is cleared to 0. In addition, clearing the TMIE bit to 0 also clears this flag to 0.

TAIFm Bits

The TAIFm bit is set to 1 when the TAIE bit in the RSCAN0CmCTR register is 1 (transmit abort interrupt enabled) and the TMTRF[1:0] flags in the RSCAN0TMSTSp register are set to 01_B (transmit abort completed).

This flag is cleared to 0 when the TMTRF[1:0] flags are cleared to 00_B after the transmit abort is completed.

TQIFm Bits

When the TXQIE bit in the RSCAN0TXQCCm register is set to 1 (transmit queue interrupt enabled) and the TXQIF bit in the RSCAN0TXQSTSm register is set to 1 (transmit queue interrupt request), the TQIFm bit is set to 1.

When the TXQIF bit (transmit queue interrupt request) in the RSCAN0TXQSTSm register is cleared to 0, this bit is cleared to 0. This flag is also cleared to 0 when the TXQIE bit is cleared to 0.

CFTIFm Bits

When the CFTXIE bit in the RSCAN0CFCCk register is set to 1 (transmit/receive FIFO buffer transmit interrupt enabled) and the CFTXIF bit in the RSCAN0CFSTSk register is set to 1 (transmit/receive FIFO transmit interrupt request), the CFTIFm bit is set to 1.

When the CFTXIF bit is cleared to 0 under the condition that the CFTIFm bit can be set to 1, this bit is cleared to 0. This flag is also cleared to 0 when the CFTXIE bit is cleared to 0.

THIFm Bits

When the THLIE bit in the RSCAN0THLCCm register is set to 1 (transmit history interrupt enabled) and the THLIF bit in the RSCAN0THLSTSm register is set to 1 (transmit history interrupt request), the THIFm bit is set to 1.

When the THLIF bit in the RSCAN0THLSTSm register is cleared to 0, this bit is cleared to 0. This flag is also cleared to 0 when the THLIE bit is cleared to 0.

chanhvinh.com

21.3.10 RSCAN0GTSC — Global Timestamp Counter Register

Access: Can be read in 8-, 16- and 32-bit units

Address: <RSCAN0_base> + 0094_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TS[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.24 RSCAN0GTSC register contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are always read as 0.
15 to 0	TS[15:0]	Timestamp Value The timestamp counter value can be read. Counter Value: 0000_H to $FFFF_H$

TS[15:0] Bits

When the TS[15:0] bits are read, the read value shows the timestamp counter (16-bit free-running counter) value at that time. When the SOF is detected, the TS[15:0] value is captured and later stored in the receive buffer or the FIFO buffer. The timestamp counter is initialized in global reset mode.

The timestamp counter starts and stops counting differently, depending on the count source.

- When the TSSS bit in the RSCAN0GCFG register is 0 (pclk):
  The timestamp counter starts counting when the RSCAN module has transitioned to global operating mode.
  This counter stops counting when the RSCAN module has transitioned to global stop mode or global test mode.
- When the TSSS bit is 1 (CANm bit time clock):
  The timestamp counter starts counting when the corresponding channel has transitioned to channel communication mode.
  This counter stops counting when the corresponding channel has transitioned to channel reset mode or channel halt mode.

chanhvinh.com

21.3.11 RSCAN0GAFLECTR — Receive Rule Entry Control Register

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0098_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	AFLDAE	—	—	—	AFLPN[4:0]				
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 21.25 RSCAN0GAFLECTR register contents

Bit Position	Bit Name	Function
31 to 9	Reserved	These bits are always read as 0. The write value should always be 0.
8	AFLDAE	Receive Rule Table Write Enable 0: Receive rule table write is disabled. 1: Receive rule table write is enabled.
7 to 5	Reserved	These bits are always read as 0. The write value should always be 0.
4 to 0	AFLPN[4:0]	Receive Rule Table Page Number Configuration A page number can be selected from a range of page 0 (00000_B) to page 7 (00111_B).

AFLDAE Bit

Setting this bit to 0 disables the write to the receive rule table. After writes to the receive rule table are completed, set this bit to 0 to disable the write to the table. The receive rule table can be read regardless of the value of this bit.

Set the AFLDAE bit to 1 only in global reset mode.

AFLPN[4:0] Bits

These bits are used to set the page number of the receive rule table. Sixteen receive rules can be set per page.

Set these bits to a value within the range of 00000_B to 00111_B.

chanhvinh.com

21.3.12 RSCAN0GAFLCFG0 — Receive Rule Configuration Register 0

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + $009C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RNC0[7:0]								RNC1[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.26 RSCAN0GAFLCFG0 register contents

Bit Position	Bit Name	Function
31 to 24	RNC0[7:0]	Number of Rules for Channel 0 Set the number of receive rules exclusively used for channel 0.
23 to 16	RNC1[7:0]	Number of Rules for Channel 1 Set the number of receive rules exclusively used for channel 1.
15 to 0	Reserved	These bits are always read as 0. The write value should always be 0.

Modify the RSCAN0GAFLCFG0 register only in global reset mode.

Up to 64 x (number of channels) rules can be registered in the receive rule table as the entire unit. The number of receive rules per channel should meet the following conditions.

- The maximum number of rules per channel is 128.
- The total of the number of rules allocated to each channel is not larger than the number of rules that can be registered in the entire unit.

RNC0[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 0 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

RNC1[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 1 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

chanhvinh.com

21.3.13 RSCAN0GAFLIDj — Receive Rule ID Register (j = 0 to 15)

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0500_H + (j * 0010_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLIDE	GAFLRTR	GAFLLB	GAFLID[28:16]												
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLID[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.27 RSCAN0GAFLIDj register contents

Bit Position	Bit Name	Function
b31	GAFLIDE	IDE Select 0: Standard ID 1: Extended ID
b30	GAFLRTR	RTR Select 0: Data frame 1: Remote frame
b29	GAFLLB	Receive Rule Target Message Select 0: When a message transmitted from another CAN node is received 1: When the own transmitted message is received
b28 to b0	GAFLID[28:0]	ID Set the ID of the receive rule. For the standard ID, set the ID in bits b10 to b0 and set bits b28 to b11 to 0.

Modify the RSCAN0GAFLIDj register when the AFLDAE bit in the RSCAN0GAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLIDE Bit

This bit is used to select the ID format (standard ID or extended ID) of the receive rule. This bit is compared with the IDE bit in the received message during the acceptance filter processing.

GAFLRTR Bit

This bit is used to select the frame format (data frame or remote frame) of the receive rule. This bit is compared with the RTR bit in the received message during the acceptance filter processing.

chanhvinh.com

GAFLLB Bit

When this bit is set to 0, data processing using the receive rule is performed when receiving messages transmitted from another CAN node.

When this bit is set to 1 when the mirror function is used, data processing using the receive rule is performed when the CAN node is receiving its own transmitted messages.

GAFLID[28:0] Bits

These bits are used to set the ID field of the receive rule. The ID value set by these bits is compared with the ID of the received message during the acceptance filter processing.

chanhvinh.com

21.3.14 RSCAN0GAFLMj — Receive Rule Mask Register (j = 0 to 15)

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0504_H + (j * 0010_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLIDEM	GAFLRTRM	—	GAFLIDM[28:16]												
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLIDM[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.28 RSCAN0GAFLMj register contents

Bit Position	Bit Name	Function
31	GAFLIDEM	IDE Mask 0: The IDE bit is not compared. 1: The IDE bit is compared.
30	GAFLRTRM	RTR Mask 0: The RTR bit is not compared. 1: The RTR bit is compared
29	Reserved	This bit is always read as 0. The write value should always be 0.
28 to 0	GAFLIDM[28:0]	ID Mask 0: The corresponding ID bit is not compared. 1: The corresponding ID bit is compared.

Modify the RSCAN0GAFLMj register when the AFLDAE bit in the RSCAN0GAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLIDEM Bit

When this bit is set to 1, filter processing is performed only for messages of the ID format specified by the GAFLIDE bit in the RSCAN0GAFLIDj register.

When this bit is cleared to 0, the IDs of all the receive messages and the specified IDs are regarded as matched. To set the GAFLIDEM bit to 0, set the GAFLIDM[28:0] bits to all 0 at the same time.

GAFLRTRM Bit

This bit is used to mask the RTR bit of the receive rule.

GAFLIDM[28:0] Bits

These bits are used to mask the corresponding ID bit of the receive rule.

chanhvinh.com

21.3.15 RSCAN0GAFLP0j — Receive Rule Pointer 0 Register (j = 0 to 15)

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0508_H + (j * 0010_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLDLC[3:0]				GAFLPTR[11:0]											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLRMV	GAFLRMDP[6:0]							—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 21.29 RSCAN0GAFLP0j register contents

Bit Position	Bit Name	Function
31 to 28	GAFLDLC[3:0]	Receive Rule DLC b31 b30 b29 b28 0 0 0 0: DLC check is disabled. 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 16	GAFLPTR[11:0]	Receive Rule Label Set the 12-bit label information.
15	GAFLRMV	Receive Buffer Enable 0: No receive buffer is used. 1: A receive buffer is used.
14 to 8	GAFLRMDP[6:0]	Receive Buffer Number Select Set the receive buffer number to store receive messages.
7 to 0	Reserved	These bits are always read as 0. The write value should always be 0.

Modify the RSCAN0GAFLP0j register when the AFLDAE bit in the RSCAN0GAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLDLC[3:0] Bits

These bits are used to set the minimum data length necessary for receiving messages. If the data length of a message that is being filtered is equal to or larger than the value set by the GAFLDLC[3:0] bits, the message passes the DLC check. Setting these bits to 0000_B disables the DLC check function allowing messages with any data length to pass the DLC check.

GAFLPTR[11:0] Bits

These bits are used to set a 12-bit label to be attached to messages that have passed through the filter. A label is attached when a message is stored in the receive buffer or the FIFO buffer.

chanhvinh.com

GAFLRMV Bit

When this bit is set to 1, receive messages that have passed through the filter are stored in the receive buffer selected by the GAFLRMDP[6:0] bits.

GAFLRMDP[6:0] Bits

These bits are used to select the number of the receive buffer that stores receive messages that have passed through the filter when the GAFLRMV bit is set to 1. Set these bits to a value smaller than the value set by the NRXMB[7:0] bits in the RSCAN0RMNB register.

chanhvinh.com

21.3.16 RSCAN0GAFLP1j — Receive Rule Pointer 1 Register (j = 0 to 15)

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + $050C_H$ + (j * 0010_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	GAFLFDP [13:0]													
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.30 RSCAN0GAFLP1j register contents

Bit Position	Bit Name	Function
31 to 14	Reserved	These bits are always read as 0. The write value should always be 0.
13 to 0	GAFLFDP [13:0]	FIFO Buffer z Select (z = 0 to 13) z = 0 to 7 0: Receive FIFO buffer z is not selected. 1: Receiver FIFO buffer z is selected. z = 8 to 13 0: Transmit/receive FIFO buffer z-8 is not selected. 1: Transmit/receive FIFO buffer z-8 is selected.

Modify the RSCAN0GAFLP1j register when the AFLDAE bit in the RSCAN0GAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLFDP [13:0] Bits

These bits are used to specify FIFO buffers that store receive messages that have passed through the filter. Up to eight FIFO buffers are selectable. However, when the GAFLRMV bit in the RSCAN0GAFLP0j register is set to 1 (a message is stored in the receive buffer), up to seven FIFO buffers can be selected.

Only receive FIFO buffers and the transmit/receive FIFO buffer for which the CFM[1:0] bits in the RSCAN0CFCCk register are set to 00_B (receive mode) or 10_B (gateway mode) are selectable.

chanhvinh.com

21.3.17 RSCAN0RMNB — Receive Buffer Number Register

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + $00A4_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	NRXMB[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.31 RSCAN0RMNB register contents

Bit Position	Bit Name	Function
31 to 8	Reserved	These bits are always read as 0. The write value should always be 0.
7 to 0	NRXMB[7:0]	Receive Buffer Number Configuration Set the number of receive buffers. Set a value of 0 to 31.

Modify the RSCAN0RMNB register only in global reset mode.

NRXMB[7:0] Bits

These bits are used to set the total number of receive buffers of the RS-CAN module. The maximum value is 16 × (number of channels).

Setting these bits all to 0 makes receive buffers unavailable.

chanhvinh.com

21.3.18 RSCAN0RMNDy — Receive Buffer New Data Register y (y = 0)

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + $00A8_H$ + (y * 0004_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMNSq (q = y × 32 + 31 to y × 32 + 16 (y = 0))															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMNSq (q = y × 32 + 15 to y × 32 + 0 (y = 0))															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.32 RSCAN0RMNDy register contents

Bit Position	Bit Name	Function
31 to 16	RMNSq	Receive Buffer Receive Complete Flag q (q = y × 32 + 31 to y × 32 + 16) 0: There is no new message in receive buffer q. 1: There is a new message in receive buffer q.
15 to 0	RMNSq	Receive Buffer Receive Complete Flag q (q = y × 32 + 15 to y × 32 + 0) 0: There is no new message in receive buffer q. 1: There is a new message in receive buffer q.

Write 0 to the RSCAN0RMNDy register in global operating mode or global test mode.

RMNSq Flags (q = 0 to 31)

Each RMNS flag is set to 1 when the processing for storing a message in the corresponding receive buffer starts.

To clear a flag to 0, the program must write 0 to the flag. Use a store instruction to write “0” to the flag and “1” to other flags. These bits cannot be set to 0 while a message is being stored. It takes ten clock cycles of pclk to store a message.

These flags are cleared to 0 in global reset mode.

chanhvinh.com

21.3.19 RSCAN0RMIDq — Receive Buffer ID Register (q = 0 to 31)

Access: Can be read in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0600_H + (q * 0010_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMIDE	RMRTR	—	RMID[28:16]												
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMID[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.33 RSCAN0RMIDq register contents

Bit Position	Bit Name	Function
31	RMIDE	Receive Buffer IDE 0: Standard ID 1: Extended ID
30	RMRTR	Receive Buffer RTR 0: Data frame 1: Remote frame
29	Reserved	This bit is always read as 0.
28 to 0	RMID[28:0]	Receive Buffer ID Data These bits contain the standard ID or extended ID of the received message. Read bits b10 to b0 for standard ID. Bits b28 to b11 are read as 0.

RMIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive buffer.

RMRTR Bit

This bit indicates the frame format (data frame or remote frame) of the message stored in the receive buffer.

RMID[28:0] Bits

These bits contain the ID of the message stored in the receive buffer.

chanhvinh.com

21.3.20 RSCAN0RMPTRq — Receive Buffer Pointer Register (q = 0 to 31)

Access: Can be read in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0604_H + (q * 0010_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDLC[3:0]				RMPTR[11:0]											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMTS[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.34 RSCAN0RMPTRq

Bit Position	Bit Name	Function
31 to 28	RMDLC[3:0]	Receive Buffer DLC Data b31 b30 b29 b28 0 0 0 0: 0 data bytes 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 16	RMPTR[11:0]	Receive Buffer Label Data Label information of the received message.
15 to 0	RMTS[15:0]	Receive Buffer Timestamp Data Timestamp value of the received message.

RMDLC[3:0] Bits

These bits indicate the data length of the message stored in the receive buffer.

RMPTR[11:0] Bits

These bits indicate the label information of the message stored in the receive buffer.

RMTS[15:0] Bits

These bits indicate the timestamp value of the message stored in the receive buffer.

chanhvinh.com

21.3.21 RSCAN0RMDF0q — Receive Buffer Data Field 0 Register (q = 0 to 31)

Access: Can be read in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0608_H + (q * 0010_H)

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDB3[7:0]								RMDB2[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMDB1[7:0]								RMDB0[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.35 RSCAN0RMDF0q register contents

Bit Position	Bit Name	Function
31 to 24	RMDB3[7:0]	Receive Buffer Data Byte 3 Receive Buffer Data Byte 2 Receive Buffer Data Byte 1 Receive Buffer Data Byte 0 Data for a message stored in the receive buffer can be read.
23 to 16	RMDB2[7:0]	
15 to 8	RMDB1[7:0]	
7 to 0	RMDB0[7:0]	

When the RMDLC[3:0] value in the RSCAN0RMPTRq register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

chanhvinh.com

21.3.22 RSCAN0RMDF1q — Receive Buffer Data Field 1 Register (q = 0 to 31)

Access: Can be read in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + $060C_H$ + (q * 0010_H)

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDB7[7:0]								RMDB6[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMDB5[7:0]								RMDB4[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.36 RSCAN0RMDF1q register contents

Bit Position	Bit Name	Function
31 to 24	RMDB7[7:0]	Receive Buffer Data Byte 7 Receive Buffer Data Byte 6 Receive Buffer Data Byte 5 Receive Buffer Data Byte 4 Data for a message stored in the receive buffer can be read.
23 to 16	RMDB6[7:0]	
15 to 8	RMDB5[7:0]	
7 to 0	RMDB4[7:0]	

When the RMDLC[3:0] value in the RSCAN0RMPTRq register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

chanhvinh.com

21.3.23 RSCAN0RFCCx — Receive FIFO Buffer Configuration and Control Register (x = 0 to 7)

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + $00B8_H$ + (x * 0004_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFIGCV[2:0]			RFIM	—	RFDC[2:0]			—	—	—	—	—	—	RFIE	RFE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 21.37 RSCAN0RFCCx register contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are always read as 0. The write value should always be 0.
15 to 13	RFIGCV[2:0]	Receive FIFO Interrupt Request Timing Select b15 b14 b13 0 0 0: When FIFO is 1/8 full. 0 0 1: When FIFO is 2/8 full. 0 1 0: When FIFO is 3/8 full. 0 1 1: When FIFO is 4/8 full. 1 0 0: When FIFO is 5/8 full. 1 0 1: When FIFO is 6/8 full. 1 1 0: When FIFO is 7/8 full. 1 1 1: When FIFO is full.
12	RFIM	Receive FIFO Interrupt Source Select 0: An interrupt occurs when the condition set by the RFIGCV[2:0] bits is met. 1: An interrupt occurs each time a message has been received.
11	Reserved	This bit is always read as 0. The write value should always be 0.
10 to 8	RFDC[2:0]	Receive FIFO Buffer Depth Configuration b10 b9 b8 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages 1 1 0: 64 messages 1 1 1: 128 messages
7 to 2	Reserved	These bits are always read as 0. The write value should always be 0.
1	RFIE	Receive FIFO Interrupt Enable 0: Receive FIFO interrupt is disabled. 1: Receive FIFO interrupt is enabled.
0	RFE	Receive FIFO Buffer Enable 0: No receive FIFO buffer is used. 1: Receive FIFO buffers are used.

chanhvinh.com

RFIGCV[2:0] Bits

These bits are used to specify the number of received messages for generating a receive FIFO interrupt request when the RFIM bit is set to 0 with a fraction for the total number of buffers (the setting of RFDC[2:0]).

When the RFDC[2:0] bits are set to 001_B (4 messages), set the RFIGCV[2:0] bits to 001_B, 011_B, 101_B, or 111_B (fractions which are even multiples of 1/4). Modify these bits only in global reset mode.

RFIM Bit

This bit is used to select a FIFO interrupt source. Modify this bit only in global reset mode.

RFDC[2:0] Bits

These bits are used to select the number of messages that can be stored in a single receive FIFO buffer. When these bits are set to 000_B, no receive FIFO buffer should be used. Modify these bits only in global reset mode.

RFIE Bit

Setting the RFIE bit to 1 enables receive FIFO interrupts. Modify this bit when the RFE bit set to 0 (no receive FIFO buffer is used).

RFE Bit

Setting the RFE bit to 1 makes receive FIFO buffers available. Clearing this bit to 0 sets the RFEMP flag in the RSCAN0RFSTSx register to 1 (the receive FIFO buffer contains no unread message (buffer empty)). Modify this bit in global operating mode or global test mode.

chanhvinh.com

21.3.24 RSCAN0RFSTSx — Receive FIFO Buffer Status Register (x = 0 to 7)

Access: Can be read in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + $00D8_H$ + (x * 0004_H)

Initial value: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFMC[7:0]								—	—	—	—	RFIF	RFMLT	RFFLL	RFEMP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 21.38 RSCAN0RFSTSx register contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are always read as 0. The write value should always be 0.
15 to 8	RFMC[7:0]	Receive FIFO Unread Message Counter The number of unread messages stored in the receive FIFO buffer is displayed.
7 to 4	Reserved	These bits are always read as 0. The write value should always be 0.
3	RFIF	Receive FIFO Interrupt Request Flag 0: No receive FIFO interrupt request is present. 1: A receive FIFO interrupt request is present.
2	RFMLT	Receive FIFO Message Lost Flag 0: No receive FIFO message is lost. 1: A receive FIFO message is lost.
1	RFFLL	Receive FIFO Buffer Full Status Flag 0: The receive FIFO buffer is not full. 1: The receive FIFO buffer is full.
0	RFEMP	Receive FIFO Buffer Empty Status Flag 0: The receive FIFO buffer contains unread message. 1: The receive FIFO buffer contains no unread message (buffer empty).

RFMC[7:0] Flag

This flag indicates the number of unread messages in the receive FIFO buffer. This flag becomes 00_H when the RFE bit in the RSCAN0RFCCx register is set to 0.

chanhvinh.com

RFIF Flag

This flag is set to 1 when the receive FIFO interrupt request generation conditions set by the RFIGCV[2:0] bits and the RFIM bit in the RSCAN0RFCCx register are met. This flag is cleared to 0 in global reset mode or by writing 0 to this flag. Modify this bit in global operating mode or global test mode.

RFMLT Flag

This flag is set to 1 when an attempt is made to store a new message while the receive FIFO buffer is full. In this case, the new message is discarded.

This flag is cleared to 0 in global reset mode or by writing 0 to this flag.

Modify this bit in global operating mode or global test mode.

RFFLL Flag

This flag is set to 1 when the number of messages stored in the receive FIFO buffer matches the FIFO buffer depth set by the RFDC[2:0] bits in the RSCAN0RFCCx register.

If the number of messages stored in the receive FIFO buffer becomes smaller than the FIFO buffer depth set by the RFDC[2:0] bits, this flag is cleared to 0. This flag is also cleared to 0 when the RFE bit in the RSCAN0RFCCx register is set to 0 (no receive FIFO buffer is used) or in global reset mode.

RFEMP Flag

This flag is set to 1 when all messages in the receive FIFO buffer have been read. This flag is also set to 1 when the RFE bit in the RSCAN0RFCCx register is 0 or in global reset mode.

This flag is cleared to 0 when even a single received message has been stored in the receive FIFO buffer.

chanhvinh.com

21.3.25 RSCAN0RFPCTRx — Receive FIFO Buffer Pointer Control Register (x = 0 to 7)

Access: Can be written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + $00F8_H$ + (x * 0004_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RFPC[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 21.39 RSCAN0RFPCTRx register contents

Bit Position	Bit Name	Function
31 to 8	Reserved	The write value should always be 0.
7 to 0	RFPC[7:0]	Receive FIFO Pointer Control When these bits are set to FF_H, the read pointer moves to the next unread message in the receive FIFO buffer.

RFPC[7:0] Bits

When the RFPC[7:0] bits are set to FF_H, the read pointer moves to the next unread message in the receive FIFO buffer. At this time, the RFMC[7:0] (receive FIFO unread message counter) value in the RSCAN0RFSTSx register is decremented. Read the RSCAN0RFID, RSCAN0RFPTR, RSCAN0RFDF0, and RSCAN0RFDF1 registers to read messages in the receive FIFO buffer, and then write FF_H to the RFPC[7:0] bits.

Write FF_H to these bits when the RFE bit in the RSCAN0RFCCx register is set to 1 (receive FIFO buffers are used) and the RFEMP flag in the RSCAN0RFSTSx register is 0 (the receive FIFO buffer contains unread messages).

chanhvinh.com

21.3.26 RSCAN0RFIDx — Receive FIFO Buffer Access ID Register (x = 0 to 7)

Access: Can be read in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + $0E00_H$ + (x * 0010_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFIDE	RFRTR	—	RFID[28:16]												
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFID[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.40 RSCAN0RFIDx register contents

Bit Position	Bit Name	Function
31	RFIDE	Receive FIFO Buffer IDE 0: Standard ID 1: Extended ID
30	RFRTR	Receive FIFO Buffer RTR 0: Data frame 1: Remote frame
29	Reserved	This bit is always read as 0.
28 to 0	RFID[28:0]	Receive FIFO Buffer ID Data The standard ID or extended ID of received message can be read. Read bits b10 to b0 for standard ID. Bits b28 to b11 are read as 0.

RFIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive FIFO buffer.

RFRTR Bit

This bit indicates the frame format (data frame or remote frame) of the message stored in the receive FIFO buffer.

RFID[28:0] Bits

These bits indicate the ID of the message stored in the receive FIFO buffer.

chanhvinh.com

21.3.27 RSCAN0RFPTRx — Receive FIFO Buffer Access Pointer Register (x = 0 to 7)

Access: Can be read in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + $0E04_H$ + (x * 0010_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDLC[3:0]				RFPTR[11:0]											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFTS[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.41 RSCAN0RFPTRx register contents

Bit Position	Bit Name	Function
31 to 28	RFDLC[3:0]	Receive FIFO Buffer DLC Data b31 b30 b29 b28 0 0 0 0: 0 data bytes 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 16	RFPTR[11:0]	Receive FIFO Buffer Label Data Label information of the received message can be read.
15 to 0	RFTS[15:0]	Receive FIFO Buffer Timestamp Data Timestamp value of the received message can be read.

RFDLC[3:0] Bits

These bits contain the data length of the message stored in the receive FIFO buffer.

RFPTR[11:0] Bits

These bits contain the label information of the message stored in the receive FIFO buffer.

RFTS[15:0] Bits

These bits contain the timestamp value of the message stored in the receive FIFO buffer.

chanhvinh.com

21.3.28 RSCAN0RFDF0x — Receive FIFO Buffer Access Data Field 0 Register (x = 0 to 7)

Access: Can be read in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + $0E08_H$ + (x * 0010_H)

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDB3[7:0]								RFDB2[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFDB1[7:0]								RFDB0[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.42 RSCAN0RFDF0x register contents

Bit Position	Bit Name	Function
31 to 24	RFDB3[7:0]	Receive FIFO Buffer Data Byte 3
23 to 16	RFDB2[7:0]	Receive FIFO Buffer Data Byte 2
15 to 8	RFDB1[7:0]	Receive FIFO Buffer Data Byte 1
7 to 0	RFDB0[7:0]	Receive FIFO Buffer Data Byte 0 Data for a message stored in the receive FIFO buffer can be read.

When the RFDLC[3:0] value in the RSCAN0RFPTRx register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

chanhvinh.com

21.3.29 RSCAN0RFDF1x — Receive FIFO Buffer Access Data Field 1 Register (x = 0 to 7)

Access: Can be read in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + $0E0C_H$ + (x * 0010_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDB7[7:0]								RFDB6[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFDB5[7:0]								RFDB4[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.43 RSCAN0RFDF1x register contents

Bit Position	Bit Name	Function
31 to 24	RFDB7[7:0]	Receive FIFO Buffer Data Byte 7 Receive FIFO Buffer Data Byte 6 Receive FIFO Buffer Data Byte 5 Receive FIFO Buffer Data Byte 4 Data for a message stored in the receive FIFO buffer can be read.
23 to 16	RFDB6[7:0]	
15 to 8	RFDB5[7:0]	
7 to 0	RFDB4[7:0]	

When the RFDLC[3:0] value in the RSCAN0RFPTRx register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

chanhvinh.com

21.3.30 RSCAN0CFCCk — Transmit/receive FIFO buffer Configuration and Control Register k (k = 0 to 5)

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0118_H + (k * 0004_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFITT[7:0]								CFTML[3:0]				CFITR	CFITSS	CFM[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFIGCV[2:0]			CFIM	—	CFDC[2:0]			—	—	—	—	—	CFTXIE	CFRXIE	CFE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 21.44 RSCAN0CFCCk register contents (1/2)

Bit Position	Bit Name	Function
31 to 24	CFITT[7:0]	Set a message transmission interval. Set Value: 00_H to FF_H
23 to 20	CFTML[3:0]	Transmit Buffer Link Configuration Set the transmit buffer number to be linked to the transmit/receive FIFO buffer.
19	CFITR	Transmit/Receive FIFO Interval Timer Resolution 0: Clock dividing pclk by ((ITRCP [15:0] bits) 1: Clock dividing pclk by ((ITRCP [15:0] bits × 10)
18	CFITSS	Transmit/Receive FIFO Interval Timer Clock Source Select 0: Interval timer clock source selected by the CFITR bit 1: Interval timer clock source is the bit time clock for the channel to which the FIFO is linked.
17, 16	CFM[1:0]	Transmit/Receive FIFO Mode Select b17 b16 0 0: Receive mode 0 1: Transmit mode 1 0: Gateway mode 1 1: Setting prohibited
15 to 13	CFIGCV[2:0]	Transmit/Receive FIFO Receive Interrupt Request Timing Select b15 b14 b13 0 0 0: When FIFO is 1/8 full. 0 0 1: When FIFO is 2/8 full. 0 1 0: When FIFO is 3/8 full. 0 1 1: When FIFO is 4/8 full. 1 0 0: When FIFO is 5/8 full. 1 0 1: When FIFO is 6/8 full. 1 1 0: When FIFO is 7/8 full. 1 1 1: When FIFO is full.

chanhvinh.com

Table 21.44 RSCAN0CFCCk register contents (2/2)

Bit Position	Bit Name	Function
b12	CFIM	Transmit/Receive FIFO Interrupt Source Select 0: • Receive mode/gateway mode When the number of received messages has met the condition set by the CFIGCV[2:0] bits, a FIFO receive interrupt request is generated. • Transmit mode/gateway mode When the buffer becomes empty upon completion of message transmission, a FIFO transmit interrupt request is generated. 1: • Receive mode/gateway mode A FIFO receive interrupt request is generated each time a message has been received. • Transmit mode/gateway mode A FIFO transmit interrupt request is generated each time a message has been transmitted.
b11	Reserved	This bit is always read as 0. The write value should always be 0.
b10 to b8	CFDC[2:0]	Transmit/Receive FIFO Buffer Depth Configuration b10 b9 b8 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages 1 1 0: 64 messages 1 1 1: 128 messages
b7 to b3	Reserved	These bits are always read as 0. The write value should always be 0.
b2	CFTXIE	Transmit/Receive FIFO Transmit Interrupt Enable 0: Transmit/receive FIFO transmit interrupt is disabled. 1: Transmit/receive FIFO transmit interrupt is enabled.
b1	CFRXIE	Transmit/Receive FIFO Receive Interrupt Enable 0: Transmit/receive FIFO receive interrupt is disabled. 1: Transmit/receive FIFO receive interrupt is enabled.
b0	CFE	Transmit/Receive FIFO Buffer Enable 0: No transmit/receive FIFO buffer is used. 1: Transmit/receive FIFO buffers are used.

CFITT[7:0] Bits

These bits are used to set a message transmission interval when transmitting messages continuously from a transmit/receive FIFO buffer whose CFM[1:0] bits are set to 01_B (transmit mode) or 10_B (gateway mode).

Clear the CFE bit to 0 (no transmit/receive FIFO buffer is used) before modifying the CFITT[7:0] bits.

CFTML[3:0] Bits

These bits are used to set the number of transmit buffer on the channel which will be linked to transmit/receive FIFO buffer k when the CFM[1:0] bits are set to 01_B (transmit mode) or 10_B (gateway mode). There are three transmit/receive FIFO buffers per channel, so channel number m of FIFO buffer k is calculated as m = k/3 (integer division). The actual assigned transmit buffer number p linked to FIFO buffer k will be ((16 × m) + CFTML[3:0]).

Refer to Table 21.11 and Table 21.12, as for the relationship between transmit/receive FIFO buffer k and transmit buffer p.

chanhvinh.com

Setting the CFDC[2:0] bits to 001_B or more enables the setting of the CFTML[3:0] bits.

Do not link to any transmit buffer which is already allocated to a transmit queue on the identical channel or to another transmit/receive FIFO buffer. Modify these bits only in global reset mode.

CFITR Bit

This bit is enabled when the CFITSS bit is 1.

When this bit is 0, the interval timer clock source is the pclk/2 clock divided by the value of the ITRCP[15:0] bits in the RSCAN0GCFG register.

When this bit is 1, the interval timer clock source is the pclk/2 clock divided by (the value of the ITRCP[15:0] bits in the RSCAN0GCFG register × 10).

CFITSS Bit

When this bit is 0, the clock selected by the CFITR bit is the count source of the interval timer.

When this bit is 1, the bit time clock of the channel to which the FIFO is linked is the count source of the interval timer.

CFM[1:0] Bits

These bits are used to select transmit/receive FIFO mode. Modify these bits only in global reset mode.

CFIGCV[2:0] Bits

These bits are used to specify the number of received messages for generating a transmit/receive FIFO receive interrupt request when the CFM[1:0] bits are set to 00_B (receive mode) or 10_B (gateway mode) and the CFIM bit is set to 0 with a fraction for the total number of buffers (the setting of CFDC[2:0]).

An interrupt request is generated when the number of stored messages reaches the specified ratio (in fraction) of the storable messages set with the CFDC[2:0] bits.

When the CFDC[2:0] bits are set to 001_B (4 messages), set the CFIGCV[2:0] bits to 001_B, 011_B, 101_B, or B'111.

Modify these bits only in global reset mode.

CFIM Bit

This bit is used to select a transmit/receive FIFO interrupt source. Modify this bit only in global reset mode.

CFDC[2:0] Bits

These bits are used to set the number of messages that can be stored in a single transmit/receive FIFO buffer. When these bits are set to 000_B, do not use a transmit/receive FIFO buffer. Modify these bits only in global reset mode.

CFTXIE Bit

When this bit is set to 1 and the CFTXIF flag in the RSCAN0CFSTSk register is set to 1, a transmit/receive FIFO transmit interrupt request is generated.

Modify this bit with the CFE bit set to 0 (no transmit/receive FIFO buffer is used).

chanhvinh.com

CFRXIE Bit

When this bit is set to 1 and the CFRXIF flag in the RSCAN0CFSTSk register is set to 1, a transmit/receive FIFO receive interrupt request is generated.

Modify this bit with the CFE bit set to 0.

CFE Bit

Setting this bit to 1 makes transmit/receive FIFO buffers available.

When this bit is set to 0 in transmit mode or gateway mode, if a message in the transmit/receive FIFO buffer is being transmitted or will be transmitted next, the transmit/receive FIFO buffer becomes empty after completion of transmission of that message, or upon detection of a CAN bus error, or arbitration-lost. In other cases or in receive mode, the transmit/receive FIFO buffer becomes empty immediately.

This bit is cleared to 0 when the following conditions are met.

- Receive mode: Global reset mode
- Transmit mode or gateway mode: Channel reset mode
  Modify this bit in the following mode.
- Receive mode: Global operating mode or global test mode
- Transmit mode or gateway mode: Channel communication mode or channel halt mode

chanhvinh.com

21.3.31 RSCAN0CFSTSk — Transmit/receive FIFO buffer Status Register (k = 0 to 5)

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0178_H + (k * 0004_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFMC[7:0]								—	—	—	CFTXIF	CFRXIF	CFMLT	CFFLL	CFEMP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 21.45 RSCAN0CFSTSk register contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are always read as 0. The write value should always be 0.
15 to 8	CFMC[7:0]	Transmit/Receive FIFO Message Counter The number of messages stored in the transmit/receive FIFO buffer.
7 to 5	Reserved	These bits are always read as 0. The write value should always be 0.
4	CFTXIF	Transmit/Receive FIFO Transmit Interrupt Request Flag 0: No transmit/receive FIFO transmit interrupt request is present. 1: A transmit/receive FIFO transmit interrupt request is present.
3	CFRXIF	Transmit/Receive FIFO Receive Interrupt Request Flag 0: No transmit/receive FIFO receive interrupt request is present. 1: A transmit/receive FIFO receive interrupt request is present.
2	CFMLT	Transmit/Receive FIFO Message Lost Flag 0: No transmit/receive FIFO message is lost. 1: A transmit/receive FIFO message is lost.
1	CFFLL	Transmit/Receive FIFO Buffer Full Status Flag 0: The transmit/receive FIFO buffer is not full. 1: The transmit/receive FIFO buffer is full.
0	CFEMP	Transmit/Receive FIFO Buffer Empty Status Flag 0: The transmit/receive FIFO buffer contains messages. 1: The transmit/receive FIFO buffer contains no message (buffer empty).

chanhvinh.com

CFMC[7:0] Bits

The CFMC[7:0] bits indicate the following values that depend on the setting of the CFM[1:0] bits in the RSCAN0CFCCk register.

- When CFM[1:0] value is 01_B (transmit mode): Number of untransmitted messages in the buffer
- When CFM[1:0] value is 00_B (receive mode): Number of unread received messages in the buffer
- When CFM[1:0] value is 10_B (gateway mode): Number of untransmitted received messages in the buffer

These bits are cleared to 0 when any of the following conditions is met.

- When CFM[1:0] value is 00_B: In global reset mode
- When CFM[1:0] value is 01_B or 10_B: In channel reset mode

CFTXIF Flag

The CFTXIF flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 01_B or 10_B, and the factor selected by the CFIM bit in the RSCAN0CFCCk register occurs

The CFTXIF flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFTXIF flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode.

CFRXIF Flag

The CFRXIF flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B or 10_B, and the factor selected by the CFIM bit in the RSCAN0CFCCk register occurs

The CFRXIF flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFRXIF flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode.

CFMLT Flag

The CFMLT flag is set to 1 when any of the following conditions is met.

- When an attempt is made to store a new message while the transmit/receive FIFO buffer is full. In this case, the new message is discarded.

The CFMLT flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFMLT flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode

chanhvinh.com

- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode

CFFLL Flag

The CFFLL flag is set to 1 when any of the following conditions is met.

- When the number of messages stored in the transmit/receive FIFO buffer matches the FIFO buffer depth set by the CFDC[2:0] bits in the RSCAN0CFCCk register.

The CFFLL flag is cleared to 0 when any of the following conditions is met.

- When the number of messages stored in the transmit/receive FIFO buffer becomes smaller than the FIFO buffer depth set by the CFDC[2:0] bits.
- When the CFE bit in the RSCAN0CFCCk register is 0 (no transmit/receive FIFO buffer is used): When not in the transmit abort
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

CFEMP Flag

The CFEMP flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B or 10_B: All messages have been read, or in global reset mode
- When the CFM[1:0] bits are set to 01_B: All messages have been transmitted, or in channel reset mode
- When the CFE bit is 0 (no transmit/receive FIFO buffer is used): Not in the transmit abort

The CFEMP flag is cleared to 0 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B: At least one received message has been stored in the transmit/receive FIFO buffer.
- When the CFM[1:0] bits are set to 01_B or 10_B: A value of FF_H has been written to the RSCAN0CFPCTRk register after data was written to the RSCAN0CFIDk, RSCAN0CFPTRk, RSCAN0CFDF0k, and RSCAN0CFDF1k registers.

chanhvinh.com

21.3.32 RSCAN0CFPCTRk — Transmit/receive FIFO buffer Pointer Control Register (k = 0 to 5)

Access: Can be written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + $01D8_H$ + (k * 0004_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CFPC[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 21.46 RSCAN0CFPCTRk register contents

Bit Position	Bit Name	Function
31 to 8	Reserved	The write value should always be 0.
7 to 0	CFPC[7:0]	Transmit/Receive FIFO Pointer Control • Receive mode: Writing FF_H to these bits moves the read pointer to the next unread message in the transmit/receive FIFO buffer. • Transmit mode: Writing FF_H to these bits moves the write pointer to the next stage of the transmit/receive FIFO buffer. • Gateway mode: Setting prohibited

CFPC[7:0] Bits

- Receive mode (CFM[1:0] value in the RSCAN0CFCCk register is 00_B):
  Writing FF_H to the CFPC[7:0] bits moves the read pointer to the next unread message in the transmit/receive FIFO buffer. At this time, the CFMC[7:0] value (transmit/receive FIFO message counter) in the RSCAN0CFSTSk register is decremented. Read the RSCAN0CFIDk, RSCAN0CFPTRk, RSCAN0CFDF0k, and RSCAN0CFDF1k registers to read messages from the transmit/receive FIFO buffer, and then write FF_H to the CFPC[7:0] bits.
  Write FF_H to these bits when the CFE bit in the RSCAN0CFCCk register is set to 1 (transmit/receive FIFO buffers are used) and the CFEMP flag in the RSCAN0CFSTSk register is cleared to 0 (the transmit/receive FIFO buffer contains messages).

chanhvinh.com

- Transmit mode (CFM[1:0] value in the RSCAN0CFCCk register is 01_B):
  Writing FF_H to the CFPC[7:0] bits stores the data written to the RSCAN0CFIDk, RSCAN0CFPTRk, RSCAN0CFDF0k, and RSCAN0CFDF1k registers in the transmit/receive FIFO buffer and moves the write pointer to the next stage of the transmit/receive FIFO buffer. At this time, the CFMC[7:0] value is incremented. Write transmit messages to the RSCAN0CFIDk, RSCAN0CFPTRk, RSCAN0CFDF0k, and RSCAN0CFDF1k registers before writing FF_H to the CFPC[7:0] bits.
  Write FF_H to these bits when the CFE bit in the RSCAN0CFCCk register is set to 1 and the CFFLL flag in the RSCAN0CFSTSk register is cleared to 0 (the transmit/receive FIFO buffer is not full).
- Gateway mode (CFM[1:0] value in the RSCAN0CFCCk register is 10_B):
  Setting prohibited

chanhvinh.com

21.3.33 RSCAN0CFIDk — Transmit/receive FIFO buffer Access ID Register (k = 0 to 5)

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + $0E80_H$ + (k * 0010_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFIDE	CFRTR	THLEN	CFID[28:16]												
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFID[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.47 RSCAN0CFIDk register contents

Bit Position	Bit Name	Function
b31	CFIDE	Transmit/Receive FIFO Buffer IDE 0: Standard ID 1: Extended ID
b30	CFRTR	Transmit/Receive FIFO Buffer RTR 0: Data frame 1: Remote frame
b29	THLEN	Transmit History Data Store Enable This bit is valid only when the CFM[1:0] value is 01_B (transmit mode). 0: Transmit history data is not stored in the buffer. 1: Transmit history data is stored in the buffer.
b28 to b0	CFID[28:0]	Transmit/Receive FIFO Buffer ID Data • When CFM[1:0] value is B'01 (transmit mode): Set standard ID or extended ID. For standard ID, write an ID to bits b10 to b0 and write 0 to bits b28 to b11. • When CFM[1:0] value is B'00 (receive mode): Standard ID or extended ID in the received message can be read. For standard ID, read bits b10 to b0. Bits b28 to b11 are read as 0.

This register is writable only when the CFM[1:0] value in the RSCAN0CFCCk register is 01_B (transmit mode). This register is readable only when the CFM[1:0] value is 00_B (receive mode). This RSCAN0CFIDk register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

CFIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, these bits are used to set the ID format of the message to be transmitted from the transmit/receive FIFO buffer.

chanhvinh.com

CFRTR Bit

This bit indicates the data format (data frame or remote frame) of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, this bit is used to set the data format of the message to be transmitted from the transmit/receive FIFO buffer.

THLEN Bit

When this bit is set to 1, the transmit history data (label information, buffer number, and buffer type) of transmit messages is stored in the transmit history buffer after transmission is completed.

This bit is enabled when the CFM[1:0] value is 01_B (transmit mode).

CFID[28:0] Bits

These bits contain the ID of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B.

When the CFM[1:0] value is B'01, this bit is used to set the ID of the message to be transmitted from the transmit/receive FIFO buffer.

chanhvinh.com

21.3.34 RSCAN0CFPTRk — Transmit/receive FIFO buffer Access Pointer Register (k = 0 to 5)

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + $0E84_H$ + (k * 0010_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDLC[3:0]				CFPTR[11:0]											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFTS[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.48 RSCAN0CFPTRk register contents

Bit Position	Bit Name	Function
b31 to b28	CFDLC[3:0]	Transmit/Receive FIFO Buffer DLC Data b31 b30 b29 b28 0 0 0 0: 0 data bytes 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
b27 to b16	CFPTR[11:0]	Transmit/Receive FIFO Buffer Label Data • When CFM[1:0] value is 01_B (transmit mode): Set the label information to be stored in the transmit history buffer. Only bits CFPTR[7:0] are valid. • When CFM[1:0] value is 00_B (receive mode): The label information of the received message can be read.
b15 to b0	CFTS[15:0]	Transmit/Receive FIFO Buffer Timestamp Data These bits are valid only when the CFM[1:0] value is B'00 (receive mode). The timestamp value of the received message can be read.

This register is writable only when the CFM[1:0] value in the RSCAN0CFCCk register is 01_B (transmit mode). This register is readable only when the CFM[1:0] value is 00_B (receive mode). This register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

chanhvinh.com

CFDLC[3:0] Bits

These bits indicate the data length of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, these bits are used to set the data length of the message to be transmitted from the transmit/receive FIFO buffer. If the data length is set to 9 bytes or more, the actual transmit data defaults to 8 bytes.

CFPTR[11:0] Bits

These bits indicate the label information attached to the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, the CFPTR[7:0] value is stored in the transmit history buffer when message transmission has been completed.

CFTS[15:0] Bits

These bits indicate the timestamp value of the message stored in the transmit/receive FIFO buffer.

These bits are valid when the CFM[1:0] value is 00_B.

chanhvinh.com

21.3.35 RSCAN0CFDF0k — Transmit/receive FIFO buffer Access Data Field 0 Register (k = 0 to 5)

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + $0E88_H$ + (k * 0010_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDB3[7:0]								CFDB2[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFDB1[7:0]								CFDB0[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.49 RSCAN0CFDF0k register contents

Bit Position	Bit Name	Function
b31 to b24	CFDB3[7:0]	Transmit/Receive FIFO Buffer Data Byte 3
b23 to b16	CFDB2[7:0]	Transmit/Receive FIFO Buffer Data Byte 2
b15 to b8	CFDB1[7:0]	Transmit/Receive FIFO Buffer Data Byte 1
b7 to b0	CFDB0[7:0]	Transmit/Receive FIFO Buffer Data Byte 0 • When CFM[1:0] value is 01_B (transmit mode): Set the transmit/receive FIFO buffer data. • When CFM[1:0] value is 00_B (receive mode): The message data stored in the transmit/receive FIFO buffer can be read.

This register is writable only when the CFM[1:0] value in the RSCAN0CFCCk register is 01_B (transmit mode). This register is readable only when the CFM[1:0] value is 00_B (receive mode). When the CFDLC[3:0] value in the RSCAN0CFPTRk register is smaller than 1000_B, data bytes for which no data is set are read as 00_H. This register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

chanhvinh.com

21.3.36 RSCAN0CFDF1k — Transmit/receive FIFO buffer Access Data Field 1 Register (k = 0 to 5)

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + $0E8C_H$ + (k * 0010_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDB7[7:0]								CFDB6[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFDB5[7:0]								CFDB4[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.50 RSCAN0CFDF1k register contents

Bit Position	Bit Name	Function
b31 to b24	CFDB7[7:0]	Transmit/Receive FIFO Buffer Data Byte 7
b23 to b16	CFDB6[7:0]	Transmit/Receive FIFO Buffer Data Byte 6
b15 to b8	CFDB5[7:0]	Transmit/Receive FIFO Buffer Data Byte 5
b7 to b0	CFDB4[7:0]	Transmit/Receive FIFO Buffer Data Byte 4 • When CFM[1:0] value is 01_B (transmit mode): Set the transmit/receive FIFO buffer data. • When CFM[1:0] value is 00_B (receive mode): The message data stored in the transmit/receive FIFO buffer can be read.

This register is writable only when the CFM[1:0] value in the RSCAN0CFCCk register is 01_B (transmit mode). This register is readable only when the CFM[1:0] value is 00_B (receive mode). When the CFDLC[3:0] value in the RSCAN0CFPTRk register is smaller than 1000_B, data bytes for which no data is set are read as 00_H. This register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

chanhvinh.com

21.3.37 RSCAN0FESTS — FIFO Empty Status Register

Access: Can be read in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0238_H

Initial value: $007F\ FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	CF5EMP	CF4EMP	CF3EMP	CF2EMP	CF1EMP	CF0EMP	RF7EMP	RF6EMP	RF5EMP	RF4EMP	RF3EMP	RF2EMP	RF1EMP	RF0EMP
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.51 RSCAN0FESTS register contents

Bit Position	Bit Name	Function
31 to 23	Reserved	These bits are always read as 0.
22 to 14	Reserved	These bits are always read as 1.
13	CF5EMP	Transmit/Receive FIFO Buffer Empty Status Flag 0: Transmit/receive FIFO buffer k contains a message. 1: Transmit/receive FIFO buffer k contains no message. (k = 0 to 5)
12	CF4EMP	
11	CF3EMP	
10	CF2EMP	
9	CF1EMP	
8	CF0EMP	
7	RF7EMP	Receive FIFO Buffer Empty Status Flag 0: Receive FIFO buffer x contains an unread message. 1: Receive FIFO buffer x contains no unread message (buffer empty). (x = 0 to 7)
6	RF6EMP	
5	RF5EMP	
4	RF4EMP	
3	RF3EMP	
2	RF2EMP	
1	RF1EMP	
0	RF0EMP	

The RSCAN0FESTS register is set to $007F\ FFFF_H$ in global reset mode.

CFkEMP Flag (k = 0 to 5)

The CFkEMP flag is set to 1 when the CFEMP flag in the RSCAN0CFSTSk register is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)). When the CFEMP flag is cleared to 0 (the transmit/receive FIFO buffer contains messages), the CFkEMP flag is cleared to 0.

RFxEMP Flag (x = 0 to 7)

The RFxEMP flag is set to 1 when the RFEMP flag in the RSCAN0RFSTSx register is set to 1 (the receive FIFO buffer contains no unread message (buffer empty)). When the RFEMP flag is cleared to 0 (the receive FIFO buffer contains unread messages), the RFxEMP flag is cleared to 0.

chanhvinh.com

21.3.38 RSCAN0FFSTS — FIFO Full Status Register

Access: Can be read in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + $023C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	CF5FLL	CF4FLL	CF3FLL	CF2FLL	CF1FLL	CF0FLL	RF7FLL	RF6FLL	RF5FLL	RF4FLL	RF3FLL	RF2FLL	RF1FLL	RF0FLL
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.52 RSCAN0FFSTS register contents

Bit Position	Bit Name	Function
31 to 14	Reserved	These bits are always read as 0.
13	CF5FLL	Transmit/Receive FIFO Buffer Full Status Flag 0: Transmit/receive buffer k is not full. 1: Transmit/receive buffer k is full. (k = 0 to 5)
12	CF4FLL	
11	CF3FLL	
10	CF2FLL	
9	CF1FLL	
8	CF0FLL	
7	RF7FLL	Receive FIFO Buffer Full Status Flag 0: Receive FIFO buffer x is not full. 1: Receive FIFO buffer x is full. (x = 0 to 7)
6	RF6FLL	
5	RF5FLL	
4	RF4FLL	
3	RF3FLL	
2	RF2FLL	
1	RF1FLL	
0	RF0FLL	

The RSCAN0FFSTS register is cleared to $0000\ 0000_H$ in global reset mode.

CFkFLL Flag (k = 0 to 5)

The CFkFLL flag is set to 1 when the CFFLL flag in the RSCAN0CFSTSk register is set to 1 (the transmit/receive FIFO buffer is full). When the CFFLL flag is cleared to 0 (the transmit/receive FIFO buffer is not full), the CFkFLL flag is cleared to 0.

RFxFLL Flag (x = 0 to 7)

The RFxFLL flag is set to 1 when the RFFLL flag in the RSCAN0RFSTSx register is set to 1 (the receive FIFO buffer is full). When the RFFLL flag is cleared to 0 (the receive FIFO buffer is not full), the RFxFLL flag is cleared to 0.

chanhvinh.com

21.3.39 RSCAN0FMSTS — FIFO Message Lost Status Register

Access: Can be read in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0240_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	CF5MLT	CF4MLT	CF3MLT	CF2MLT	CF1MLT	CF0MLT	RF7MLT	RF6MLT	RF5MLT	RF4MLT	RF3MLT	RF2MLT	RF1MLT	RF0MLT
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.53 RSCAN0FMSTS register contents

Bit Position	Bit Name	Function
31 to 14	Reserved	These bits are always read as 0.
13	CF5MLT	Transmit/Receive FIFO Buffer Message Lost Status Flag 0: No transmit/receive FIFO buffer k message is lost. 1: A transmit/receive FIFO buffer k message is lost. (k = 0 to 5)
12	CF4MLT	
11	CF3MLT	
10	CF2MLT	
9	CF1MLT	
8	CF0MLT	
7	RF7MLT	Receive FIFO Buffer Message Lost Status Flag 0: No receive FIFO buffer x message is lost. 1: A receive FIFO buffer x message is lost. (x = 0 to 7)
6	RF6MLT	
5	RF5MLT	
4	RF4MLT	
3	RF3MLT	
2	RF2MLT	
1	RF1MLT	
0	RF0MLT	

The RSCAN0FMSTS register is cleared to $0000\ 0000_H$ in global reset mode.

CFkMLT Flag (k = 0 to 5)

The CFkMLT flag is set to 1 when the CFMLT flag in the RSCAN0CFSTSk register is set to 1 (a transmit/receive FIFO message is lost). When the CFMLT flag is cleared to 0, the CFkMLT flag is cleared to 0.

RFxMLT Flag (x = 0 to 7)

The RFxMLT flag is set to 1 when the RFMLT flag in the RSCAN0RFSTSx register is set to 1 (a receive FIFO message is lost). When the RFMLT flag is cleared to 0, the RFxMLT flag is cleared to 0.

chanhvinh.com

21.3.40 RSCAN0RFISTS — Receive FIFO Buffer Interrupt Flag Status Register

Access: Can be read in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0244_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RF7IF	RF6IF	RF5IF	RF4IF	RF3IF	RF2IF	RF1IF	RF0IF
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.54 RSCAN0RFISTS register contents

Bit Position	Bit Name	Function
31 to 8	Reserved	These bits are always read as 0.
7	RF7IF	Receive FIFO Buffer Interrupt Request Status Flag 0: No receive FIFO buffer x interrupt request is present. 1: A receive FIFO buffer x interrupt request is present. (x = 0 to 7)
6	RF6IF	
5	RF5IF	
4	RF4IF	
3	RF3IF	
2	RF2IF	
1	RF1IF	
0	RF0IF	

The RSCAN0RFISTS register is cleared to $0000\ 0000_H$ in global reset mode.

RFxIF Flag (x = 0 to 7)

The RFxIF flag is set to 1 when the RFIF flag in the RSCAN0RFSTSx register is set to 1 (a receive FIFO interrupt request is present). When the RFIF flag is cleared to 0, the RFxIF flag is cleared to 0.

chanhvinh.com

21.3.41 RSCAN0CFRISTS — Transmit/receive FIFO buffer Receive Interrupt Flag Status Register

Access: Can be read in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0248_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	CF5RXIF	CF4RXIF	CF3RXIF	CF2RXIF	CF1RXIF	CF0RXIF
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.55 RSCAN0CFRISTS register contents

Bit Position	Bit Name	Function
31 to 6	Reserved	These bits are always read as 0.
5	CF5RXIF	Transmit/Receive FIFO Buffer Receive Interrupt Request Status Flag 0: No transmit/receive FIFO buffer k receive interrupt request is present. 1: A transmit/receive FIFO buffer k receive interrupt request is present. (k = 0 to 5)
4	CF4RXIF	
3	CF3RXIF	
2	CF2RXIF	
1	CF1RXIF	
0	CF0RXIF	

The RSCAN0CFRISTS register is cleared to $0000\ 0000_H$ in global reset mode.

CFkRXIF Flag (k = 0 to 5)

The CFkRXIF flag is set to 1 when the CFRXIF flag in the RSCAN0CFSTSk register is set to 1 (a transmit/receive FIFO receive interrupt request is present). When the CFRXIF flag is cleared to 0, the CFkRXIF flag is cleared to 0.

chanhvinh.com

21.3.42 RSCAN0CFTISTS — Transmit/receive FIFO buffer Transmit Interrupt Flag Status Register

Access: Can be read in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + $024C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	CF5TXIF	CF4TXIF	CF3TXIF	CF2TXIF	CF1TXIF	CF0TXIF
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.56 RSCAN0CFTISTS register contents

Bit Position	Bit Name	Function
31 to 6	Reserved	These bits are always read as 0.
5	CF5TXIF	Transmit/Receive FIFO Buffer Transmit Interrupt Request Status Flag 0: No transmit/receive FIFO buffer k transmit interrupt request is present. 1: A transmit/receive FIFO buffer k transmit interrupt request is present. (k = 0 to 5)
4	CF4TXIF	
3	CF3TXIF	
2	CF2TXIF	
1	CF1TXIF	
0	CF0TXIF	

The RSCAN0CFTISTS register is cleared to $0000\ 0000_H$ in global reset mode.

CFkTXIF Flag (k = 0 to 5)

The CFkTXIF flag is set to 1 when the CFTXIF flag in the RSCAN0CFSTSk register is set to 1 (a transmit/receive FIFO transmit interrupt request is present). When the CFTXIF flag is cleared to 0, the CFkTXIF flag is cleared to 0.

chanhvinh.com

21.3.43 RSCAN0TMCp — Transmit Buffer Control Register (p = 0 to 31)

Access: Can be read/written in 8-bit units

Address: <RSCAN0_base> + 0250_H + (01_H × p)

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	TMOM	TMTAR	TMTR
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W*1	R/W*1

Note 1. The only effective value for writing to this bit is 1, which sets the bit. Otherwise writing to the bit results in retention of its state.

Table 21.57 RSCAN0TMCp register contents

Bit Position	Bit Name	Function
7 to 3	Reserved	These bits are always read as 0. The write value should always be 0.
2	TMOM	One-Shot Transmission Enable 0: One-shot transmission is disabled. 1: One-shot transmission is enabled.
1	TMTAR	Transmit Abort Request 0: Transmit abort is not requested. 1: Transmit abort is requested.
0	TMTR	Transmit Request 0: Transmission is not requested. 1: Transmission is requested.

When the RSCAN0TMCp register meets any of the following conditions, set it to 00_H.

- The RSCAN0TMCp register corresponds to the transmit buffer number selected by the CFTML[3:0] bits in the RSCAN0CFCCk register (p = m × 16 + the value of CFTML[3:0] bits).
- The RSCAN0TMCp register corresponds to the transmit buffer allocated to the transmit queue by the TXQDC[3:0] bits in the RSCAN0TXQCCm register (m = 0 or 1) (p = (m × 16 + 15) to (m × 16 + 15 - the value of TXQDC[3:0] bits)).

Bits in the RSCAN0TMCp register are all cleared to 0 in channel reset mode. Modify the RSCAN0TMCp register in channel communication mode or channel halt mode.

TMOM Bit

Setting this bit to 1 enables one-shot transmission. When transmission fails, retransmission defined in the CAN protocol is not performed.

Modify the TMOM bit when the TMTRM flag in the RSCAN0TMSTSp register is set to 0. Set the TMOM bit to 1 together with the TMTR bit.

chanhvinh.com

TMTAR Bit

Setting this bit to 1 generates a transmit abort request for the message stored in the transmit buffer. However, a message that is being transmitted or one that will be transmitted next cannot be aborted.

When the TMTAR bit can be set to 1 when TMTR bit is 1.

The TMTAR bit is cleared to 0 when any of the following conditions is met, but cannot be cleared by the program writing 0 to the bit.

- Transmission has been completed.
- Transmit abort has been completed.
- An error or arbitration loss has been detected.

If this bit becomes 0 at the same time as the program writes 1 to this bit, this bit becomes 0.

TMTR Bit

Setting this bit to 1 transmits the message stored in the transmit buffer.

The TMTR bit is cleared to 0 when any of the following conditions is met, but cannot be cleared by the program writing 0 to the bit.

- Transmission has been completed.
- Transmit abort has been completed after the TMTAR bit was set to 1.
- An error or arbitration-lost has been detected with the TMOM bit set to 1.

Set the TMTR bit to 1 when the value of TMTRF[1:0] in the RSCAN0TMSTSp register is 00_B.

chanhvinh.com

21.3.44 RSCAN0TMSTSp — Transmit Buffer Status Register (p = 0 to 31)

Access: Can be read/written in 8-bit units

Address: <RSCAN0_base> + $02D0_H$ + ($01_H \times p$)

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	TMTARM	TMTRM	TMTRF[1:0]		TMTSTS
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R

Table 21.58 RSCAN0TMSTSp register contents

Bit Position	Bit Name	Function
7 to 5	Reserved	These bits are always read as 0. The write value should always be 0.
4	TMTARM	Transmit Buffer Transmit Abort Request Status Flag 0: No transmit abort request is present. 1: A transmit abort request is present.
3	TMTRM	Transmit Buffer Transmit Request Status Flag 0: No transmit request is present. 1: A transmit request is present.
2, 1	TMTRF[1:0]	Transmit Buffer Transmit Result Status Flag b2 b1 0 0: Transmission is in progress or no transmit request is present. 0 1: Transmit abort has been completed. 1 0: Transmission has been completed (without transmit abort request). 1 1: Transmission has been completed (with transmit abort request).
0	TMTSTS	Transmit Buffer Transmit Status Flag 0: Transmission is not in progress. 1: Transmission is in progress.

The RSCAN0TMSTSp register is cleared to all 0 in channel reset mode.

TMTARM Flag

The TMTARM flag is set to 1 when the TMTAR bit in the RSCAN0TMCp register is set to 1.

The TMTARM flag is set to 0 when the TMTAR bit in the RSCAN0TMCp register is set to 0.

TMTRM Flag

The TMTRM flag is set to 1 when the TMTR bit in the RSCAN0TMCp register is set to 1.

The TMTRM flag is set to 0 when the TMTR bit in the RSCAN0TMCp register is set to 0

chanhvinh.com

TMTRF[1:0] Flag

This flag indicates the result of transmission from the transmit buffer.

00_B: Transmission is in progress or no transmit request is present.

01_B: Transmission from the transmit buffer was aborted.

10_B: Transmission has been completed with the TMTAR bit in the RSCAN0TMCp register set to 0 (transmit abort is not requested).

11_B: Transmission has been completed with the TMTAR bit in the RSCAN0TMCp register set to 1 (transmit abort is requested).

Write 00_B to the TMTRF[1:0] flag in channel communication mode or channel halt mode. Do not write any value other than 00_B to this flag.

TMTSTS Flag

This flag is set to 1 when transmission from the transmit buffer starts, and is cleared to 0 when transmission from the transmit buffer has been completed or terminated due to a bus error or arbitration lost.

chanhvinh.com

21.3.45 RSCAN0TMTRSTSy — Transmit Buffer Transmit Request Status Register y (y = 0)

Access: Can be read in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0350_H + (y * 0004_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTRSTSp (p = y × 32 + 31 to y × 32 + 16 (y = 0))															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTRSTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0))															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.59 RSCAN0TMTRSTSy register contents

Bit Position	Bit Name	Function
31 to 16	TMTRSTSp	Transmit Buffer Transmit Request Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: No transmission is requested. 1: Transmission is requested.
15 to 0	TMTRSTSp	Transmit Buffer Transmit Request Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: No transmit request is present. 1: A transmit request is present.

TMTRSTSp Flags (p = 0 to 31)

These flags indicate the status of the TMTR bit in the RSCAN0TMCp register.

When the TMTR bit is set to 1 (transmission is requested), the corresponding TMTRSTSp flag is set to 1.

The corresponding TMTRSTSp flag is cleared to 0 when the TMTR bit is set to 0 (transmission is not requested) or in channel reset mode.

Table 21.60 shows the bit assignment.

Table 21.60 TMTRSTSp Bit Assignment

Bit	Channel	Transmit Buffer Number
0	0	0
1	0	1
.	.	.
.	.	.
15	0	15
16	1	0
.	.	.
.	.	.
30	1	14
31	1	15

chanhvinh.com

21.3.46 RSCAN0TMTARSTSy — Transmit Buffer Transmit Abort Request Status Register y (y = 0)

Access: Can be read in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0360_H + (y * 0004_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTARSTSp (p = y × 32 + 31 to y × 32 + 16 (y = 0))															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTARSTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0))															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.61 RSCAN0TMTARSTSy register contents

Bit Position	Bit Name	Function
31 to 16	TMTARSTSp	Transmit Buffer Transmit Abort Request Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: No transmission abort is requested. 1: Transmission abort is requested.
15 to 0	TMTARSTSp	Transmit Buffer Transmit Abort Request Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: No transmit abort request is present. 1: A transmit abort request is present.

TMTARSTSp Flags (p = 0 to 31)

These flags indicate the status of the TMTAR bit in the RSCAN0TMCp register.

When the TMTAR bit is set to 1 (transmit abort is requested), the corresponding TMTARSTSp flag is set to 1.

The corresponding TMTARSTSp flag is cleared to 0 when the TMTAR bit is set to 0 (transmit abort is not requested) or in channel reset mode.

Table 21.62 shows the bit assignment.

Table 21.62 TMTARSTSp Bit Assignment

Bit	Channel	Transmit Buffer Number
0	0	0
1	0	1
.	.	.
.	.	.
15	0	15
16	1	0
.	.	.
.	.	.
30	1	14
31	1	15

chanhvinh.com

21.3.47 RSCAN0TMTCSTSy — Transmit Buffer Transmit Complete Status Register y (y = 0)

Access: Can be read in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0370_H + (y * 0004_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTCSTSp (p = y × 32 + 31 to y × 32 + 16 (y = 0))															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTCSTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0))															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.63 RSCAN0TMTCSTSy register contents

Bit Position	Bit Name	Function
31 to 16	TMTCSTSp	Transmit Buffer Transmit Complete Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: Transmission is not completed 1: Transmission is completed
15 to 0	TMTCSTSp	Transmit Buffer Transmit Complete Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: Transmission has not been completed. 1: Transmission has been completed.

TMTCSTSp Flags (p = 0 to 31)

When the TMTRF[1:0] flag in the RSCAN0TMSTSp register is set to 10_B (transmission has been completed (without transmit abort request)) or 11_B (transmission has been completed (with transmit abort request)), the corresponding TMTCSTSp flag is set to 1.

A TMTCSTSp flag is cleared to 0 when the corresponding TMTRF[1:0] flag is set to 00_B or in channel reset mode.

Table 21.64 shows the bit assignment.

Table 21.64 TMTCSTSp Bit Assignment

Bit	Channel	Transmit Buffer Number
0	0	0
1	0	1
. .	. .	. .
15	0	15
16	1	0
. .	. .	. .
30	1	14
31	1	15

chanhvinh.com

21.3.48 RSCAN0TMTASTSy — Transmit Buffer Transmit Abort Status Register y (y = 0)

Access: Can be read in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0380_H + (y * 0004_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTASTSp (p = y × 32 + 31 to y × 32 + 16 (y = 0))															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTASTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0))															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.65 RSCAN0TMTASTSy register contents

Bit Position	Bit Name	Function
31 to 16	TMTASTSp	Transmit Buffer Transmit Abort Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: Transmission is not aborted 1: Transmission is aborted
15 to 0	TMTASTSp	Transmit Buffer Transmit Abort Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: Transmission is not aborted. 1: Transmission is aborted.

TMTASTSp Flags (p = 0 to 31)

When the TMTRF[1:0] flag in the RSCAN0TMSTSp register is set to 01_B (transmit abort has been completed), the corresponding TMTASTSp flag is set to 1.

A TMTASTSp flag is cleared to 0 when the corresponding TMTRF[1:0] flag is set to 00_B or in channel reset mode.

Table 21.66 shows the bit assignment.

Table 21.66 TMTASTSp Bit Assignment

Bit	Channel	Transmit Buffer Number
0	0	0
1	0	1
. .	. .	. .
15	0	15
16	1	0
. .	. .	. .
30	1	14
31	1	15

chanhvinh.com

21.3.49 RSCAN0TMIECy — Transmit Buffer Interrupt Enable Configuration Register y (y = 0)

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0390_H + (y * 0004_H)

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMIEp (p = y × 32 + 31 to y × 32 + 16 (y = 0))															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMIEp (p = y × 32 + 15 to y × 32 + 0 (y = 0))															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.67 RSCAN0TMIECy register contents

Bit Position	Bit Name	Function
31 to 16	TMIEp	Transmit Buffer Interrupt Enable Bit p (p = y × 32 + 31 to y × 32 + 16) 0: Transmit buffer interrupt is disabled 1: Transmit buffer interrupt is enabled
15 to 0	TMIEp	Transmit Buffer Interrupt Enable Bit p (p = y × 32 + 15 to y × 32 + 0) 0: Transmit buffer interrupt is disabled. 1: Transmit buffer interrupt is enabled.

TMIEp Bits (p = 0 to 31)

When any of these bits is set to 1 and the corresponding transmission has been completed, a transmit buffer interrupt request is generated.

Modify these bits when the TMTRM flag in the corresponding RSCAN0TMSTSp register is 0 (no transmit request is present).

Write 0 to bits corresponding to transmit buffers linked to transmit/receive FIFO buffers or transmit buffers allocated to the transmit queue.

Table 21.68 shows the bit assignment.

Table 21.68 TMIEp Bit Assignment

Bit	Channel	Transmit Buffer Number
0	0	0
1	0	1
.	.	.
.	.	.
15	0	15
16	1	0
.	.	.
.	.	.
30	1	14
31	1	15

chanhvinh.com

21.3.50 RSCAN0TMIDp — Transmit Buffer ID Register (p = 0 to 31)

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 1000_H + (p * 0010_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMIDE	TMRTR	THLEN	TMID[28:16]												
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMID[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.69 RSCAN0TMIDp register contents

Bit Position	Bit Name	Function
31	TMIDE	Transmit Buffer IDE 0: Standard ID 1: Extended ID
30	TMRTR	Transmit Buffer RTR 0: Data frame 1: Remote frame
29	THLEN	Transmit History Data Store Enable 0: Transmit history data is not stored in the buffer. 1: Transmit history data is stored in the buffer.
28 to 0	TMID[28:0]	Transmit Buffer ID Data Set standard ID or extended ID. For standard ID, write an ID to bits 10 to 0 and write 0 to bits 28 to 11.

Modify this register when the TMTRM bit in the corresponding RSCAN0TMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write data to this register. If this register is allocated to the transmit queue, only write data to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

TMIDE Bit

This bit is used to set the ID format of the message to be transmitted from the transmit buffer.

TMRTR Bit

This bit is used to set the data format of the message to be transmitted from the transmit buffer.

THLEN Bit

When this bit is set to 1, the transmit history data of the transmit message (the label information and the number and type of the transmit buffer) are stored in the transmit history buffer after transmission is completed.

TMID[28:0] Bits

These bits are used to set the ID of the message to be transmitted from the transmit buffer.

chanhvinh.com

21.3.51 RSCAN0TMPTRp — Transmit Buffer Pointer Register (p= 0 to 31)

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 1004_H + (p * 0010_H)

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMDLC[3:0]				—	—	—	—	TMPTR[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.70 RSCAN0TMPTRp register contents

Bit Position	Bit Name	Function
31 to 28	TMDLC[3:0]	Transmit Buffer DLC Data b31 b30 b29 b28 0 0 0 0: 0 data bytes 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 24	Reserved	These bits are always read as 0. The write value should always be 0.
23 to 16	TMPTR[7:0]	Transmit Buffer Label Data Set the label information to be stored in the transmit history buffer.
15 to 0	Reserved	These bits are always read as 0. The write value should always be 0.

Modify this register when the TMTRM bit in the corresponding RSCAN0TMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write to this register. If this register is allocated to the transmit queue, only write to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

TMDLC[3:0] Bits

These bits are used to set the data length of the message to be transmitted from the transmit buffer when the TMRTR bit in the RSCAN0TMIDp register is set to 0 (data frame). If the data length is set to 9 bytes or more, the transmit data is 8 bytes long.

When the TMRTR bit is set to 1 (remote frame), set the data length of messages to be requested.

TMPTR[7:0] Bits

When message transmission has been completed, the TMPTR[7:0] value is stored in the transmit history buffer.

chanhvinh.com

21.3.52 RSCAN0TMDF0p — Transmit Buffer Data Field 0 Register (p = 0 to 31)

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 1008_H + (p * 0010_H)

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMDB3[7:0]								TMDB2[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMDB1[7:0]								TMDB0[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.71 RSCAN0TMDF0p register contents

Bit Position	Bit Name	Function
31 to 24	TMDB3[7:0]	Transmit Buffer Data Byte 3 Transmit Buffer Data Byte 2 Transmit Buffer Data Byte 1 Transmit Buffer Data Byte 0 Set the transmit buffer data.
23 to 16	TMDB2[7:0]	
15 to 8	TMDB1[7:0]	
7 to 0	TMDB0[7:0]	

Modify this register when the TMTRM bit in the corresponding RSCAN0TMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write to this register. If this register is allocated to the transmit queue, only write to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

chanhvinh.com

21.3.53 RSCAN0TMDF1p — Transmit Buffer Data Field 1 Register (p = 0 to 31)

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + $100C_H$ + (p * 0010_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMDB7[7:0]								TMDB6[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMDB5[7:0]								TMDB4[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.72 RSCAN0TMDF1p register contents

Bit Position	Bit Name	Function
31 to 24	TMDB7[7:0]	Transmit Buffer Data Byte 7 Transmit Buffer Data Byte 6 Transmit Buffer Data Byte 5 Transmit Buffer Data Byte 4 Set the transmit buffer data.
23 to 16	TMDB6[7:0]	
15 to 8	TMDB5[7:0]	
7 to 0	TMDB4[7:0]	

Modify this register when the TMTRM bit in the corresponding RSCAN0TMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write to this register. If this register is allocated to the transmit queue, only write to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

chanhvinh.com

21.3.54 RSCAN0TXQCCm — Transmit Queue Configuration and Control Register (m = 0 or 1)

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + $03A0_H$ + (m * 0010_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	TXQIM	TXQIE	TXQDC[3:0]				—	—	—	—	—	—	—	TXQE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 21.73 RSCAN0TXQCCm register contents

Bit Position	Bit Name	Function
31 to 14	Reserved	These bits are always read as 0. The write value should always be 0.
13	TXQIM	Transmit Queue Interrupt Source Select 0: When the buffer becomes empty upon completion of message transmission, a transmit queue interrupt request is generated. 1: A transmit queue interrupt request is generated each time a message has been transmitted.
12	TXQIE	Transmit Queue Interrupt Enable 0: Transmit queue interrupt is disabled. 1: Transmit queue interrupt is enabled.
11 to 8	TXQDC[3:0]	Transmit Queue Depth Configuration Setting these bits to g (g = 2 to 15) makes the (g + 1)-buffer transmit queue available. Setting these bits to 0 disables the transmit queue. Setting these bits to 1 is prohibited.
7 to 1	Reserved	These bits are always read as 0. The write value should always be 0.
0	TXQE	Transmit Queue Enable 0: The transmit queue is not used. 1: The transmit queue is used.

TXQIM Bit

This bit is used to select a transmit queue interrupt source. Modify this bit in channel reset mode.

TXQIE Bit

When the TXQIE bit is set to 1 and the source selected by the TXQIM bit occurs, an interrupt request is generated.

Set the TXQE bit to 0 before modifying the TXQIE bit.

chanhvinh.com

TXQDC[3:0] Bits

These bits are used to specify the number of transmit buffers to be allocated to the transmit queues. Transmit buffers are allocated to transmit queues in descending order of buffer number, that is, from (m × 16 + 15) to (m × 16 + 0). For examples of how buffer allocation is done, see Figure 21.9. Modify these bits only in channel reset mode.

TXQE Bit

Setting this bit to 1 makes the transmit queue available. Modify this bit in channel communication mode or channel halt mode. This bit is cleared to 0 in channel reset mode.

Before setting the TXQE bit to 1, set the TXQDC[3:0] bits to 0010_B or more.

chanhvinh.com

21.3.55 RSCAN0TXQSTSm — Transmit Queue Status Register (m = 0 or 1)

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + $03C0_H$ + (m * 0004_H)

Initial value: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQIF	TXQFLL	TXQEMP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 21.74 RSCAN0TXQSTSm register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	The read value is undefined. The write value should always be 0.
2	TXQIF	Transmit Queue Interrupt Request Flag 0: No transmit queue interrupt request is present. 1: A transmit queue interrupt request is present.
1	TXQFLL	Transmit Queue Full Status Flag 0: The transmit queue is not full. 1: The transmit queue is full.
0	TXQEMP	Transmit Queue Empty Status Flag 0: The transmit queue contains messages. 1: The transmit queue contains no message (transmit queue empty).

TXQIF Flag

The TXQIF flag is set to 1 when the event specified by the TXQIM bit in the RSCAN0TXQCCm register has occurred.

The TXQIF flag is cleared to 0 in channel reset mode or by writing 0 to this flag. This flag is not cleared to 0 by setting the TXQE bit in the RSCAN0TXQCCm register to 0 (the transmit queue is not used).

TXQFLL Flag

The TXQFLL flag is set to 1 when the number of messages set for the transmit queue matches the transmit queue depth set by the TXQDC[3:0] bits in the RSCAN0TXQCCm register.

This flag is cleared to 0 in any of the following cases.

- The number of messages set for the transmit queue is smaller than the transmit queue depth set by the TXQDC[3:0] bits.
- In channel reset mode

chanhvinh.com

TXQEMP Flag

The TXQEMP flag is cleared to 0 when even a single message is pending in the transmit queue.

This flag is set to 1 in any of the following cases.

- The TXQE bit is set to 0 (the transmit queue is not used).
- The transmit queue becomes empty.
- In channel reset mode

chanhvinh.com

21.3.56 RSCAN0TXQPCTRm — Transmit Queue Pointer Control Register (m = 0 or 1)

Access: Can be written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + $03E0_H$ + (m * 0004_H)

Initial value: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TXQPC[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 21.75 RSCAN0TXQPCTRm register contents

Bit Position	Bit Name	Function
31 to 8	Reserved	The write value should always be 0.
7 to 0	TXQPC[7:0]	Transmit Queue Pointer Control Writing FF_H to these bits moves the write pointer of the transmit queue to the next queue buffer. Set Value: FF_H

TXQPC[7:0] Bits

Writing FF_H to the TXQPC[7:0] bits moves the write pointer to the next transmit queue buffer and generates a transmit request of the message. Write transmit messages to the RSCAN0TMIDp, RSCAN0TMPTRp, RSCAN0TMDF0p, and RSCAN0TMDF1p registers (p = 15 and 31) before writing FF_H to the TXQPC[7:0] bits.

Write FF_H only when the TXQE bit in the RSCAN0TXQCCm register is set to 1 (the transmit queue is used) and the TXQFLL flag in the RSCAN0TXQSTSm register is set to 0 (not full).

chanhvinh.com

21.3.57 RSCAN0THLCCm — Transmit History Configuration and Control Register (m = 0 or 1)

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0400_H + (m * 0004_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	THLDTE	THLIM	THLIE	—	—	—	—	—	—	—	THLE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 21.76 RSCAN0THLCCm register contents

Bit Position	Bit Name	Function
31 to 11	Reserved	These bits are always read as 0. The write value should always be 0.
10	THLDTE	Transmit History Target Buffer Select 0: Entry from transmit/receive FIFO buffers and transmit queue 1: Entry from transmit buffers, transmit/receive FIFO buffers, and transmit queue
9	THLIM	Transmit History Interrupt Source Select 0: When 12 sets of data have been stored in the transmit history buffer 1: When a single set of transmit history data has been stored
8	THLIE	Transmit History Interrupt Enable 0: Transmit history interrupt is disabled. 1: Transmit history interrupt is enabled.
7 to 1	Reserved	These bits are always read as 0. The write value should always be 0.
0	THLE	Transmit History Buffer Enable 0: Transmit history buffer is not used. 1: Transmit history buffer is used.

THLDTE Bit

When this bit is set to 0, the transmit history data of messages transmitted from transmit/receive FIFO buffers and the transmit queue is stored in the transmit history buffer. When this bit is set to 1, the transmit history data of messages transmitted from transmit buffers, transmit/receive FIFO buffers, and the transmit queue is stored in the transmit history buffer.

Modify this bit only in channel reset mode.

THLIM Bit

This bit is used to select a transmit history interrupt source.

Modify this bit only in channel reset mode.

THLIE Bit

When the THLIE bit is set to 1 and the source selected by the THLIM bit has occurred, a transmit history interrupt request is generated. Modify the THLIE bit only when the THLE bit set to 0.

chanhvinh.com

THLE Bit

Setting this bit to 1 makes the transmit history buffer available. When data transmission from the buffer selected by the THLDTE bit has been completed, the transmit history data of transmit messages is stored in the transmit history buffer.

Modify this bit in channel communication mode or channel halt mode.

chanhvinh.com

21.3.58 RSCAN0THLSTSm — Transmit History Status Register (m = 0 or 1)

Access: Can be read in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0420_H + (m * 0004_H)

Initial value: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	THLMC[4:0]					—	—	—	—	THLIF	THLELT	THLFLL	THLEMP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 21.77 RSCAN0THLSTSm register contents

Bit Position	Bit Name	Function
31 to 13	Reserved	These bits are always read as 0. The write value should always be 0.
12 to 8	THLMC[4:0]	Transmit History Buffer Unread Data Counter These bits indicate the number of unread data sets stored in the transmit history buffer.
7 to 4	Reserved	These bits are always read as 0. The write value should always be 0.
3	THLIF	Transmit History Interrupt Request Flag 0: No transmit history interrupt request is present. 1: A transmit history interrupt request is present.
2	THLELT	Transmit History Buffer Overflow Flag 0: Transmit history buffer overflow has not occurred. 1: Transmit history buffer overflow has occurred.
1	THLFLL	Transmit history Buffer Full Status Flag 0: Transmit history buffer is not full. 1: Transmit history buffer is full.
0	THLEMP	Transmit History Buffer Empty Status Flag 0: Transmit history buffer contains unread data. 1: Transmit history buffer contains no unread data (buffer empty).

THLMC[4:0] Bits

These bits indicate the number of unread data sets stored in the transmit history buffer.

THLIF Flag

The THLIF flag is set to 1 when the interrupt source specified with the THLIM bit in the RSCAN0THLCCm register occurs.

This flag is cleared to 0 in channel reset mode or by the program writing 0 to this flag.

chanhvinh.com

THLELT Flag

The THLELT flag is set to 1 when an attempt is made to store new transmit history data while the transmit history buffer is full. In this case, the new data is discarded. This flag becomes 0 in channel reset mode or by the program writing 0 to this flag.

THLFLL Flag

The THLFLL flag is set to 1 when 16 data sets have been stored in the transmit history buffer, and is cleared to 0 when the number of data sets stored in the transmit history buffer has decreased to less than 16. This bit is also cleared to 0 in channel reset mode or when the THLE bit in the RSCAN0THLCCm register is set to 0 (transmit history buffer is not used).

THLEMP Flag

The THLEMP flag is cleared to 0 when even a single set of transmit history data has been stored in the transmit history buffer.

This flag is set to 1 when all the data in the transmit history buffer has been read. This flag is also set to 1 in channel reset mode or when the THLE bit in the RSCAN0THLCCm register is set to 0 (transmit history buffer is not used).

chanhvinh.com

21.3.59 RSCAN0THLACCm — Transmit History Access Register (m = 0 or 1)

Access: Can be read in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 1800_H + (m * 0004_H)

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TID[7:0]								—	BN[3:0]				BT[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.78 RSCAN0THLACCm register contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are always read as 0.
15 to 8	TID[7:0]	Label Data The label information of stored data can be read.
7	Reserved	This bit is always read as 0.
6 to 3	BN[3:0]	Buffer Number Data The buffer number of transmit source (transmit buffer, transmit/receive FIFO or transmit queue) can be read.
2 to 0	BT[2:0]	Buffer Type Data b2 b1 b0 0 0 1: Transmit buffer 0 1 0: Transmit FIFO buffer 1 0 0: Transmit queue

TID[7:0] Bits

These bits indicate the label information of transmit history data stored in the transmit history buffer.

BN[3:0] Bits

These bits indicate the transmit history data stored in the transmit history buffer.

BT[2:0] Bits

These bits indicate the transmit history data stored in the transmit history buffer.

chanhvinh.com

21.3.60 RSCAN0THLPCTRm — Transmit History Pointer Control Register (m = 0 or 1)

Access: Can be written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0440_H + (m * 0004_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	THLPC[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 21.79 RSCAN0THLPCTRm register contents

Bit Position	Bit Name	Function
31 to 8	Reserved	The write value should always be 0.
7 to 0	THLPC[7:0]	Transmit History List Pointer Control Writing FF_H to these bits moves the read pointer to the next unread data in the transmit history buffer. Set Value: FF_H

THLPC[7:0] Bits

When the THLPC[7:0] bits are set to FF_H, the read pointer moves to the next data in the transmit history buffer. At this time, the THLMC[4:0] (transmit history buffer unread data counter) value in the RSCAN0THLSTSm register is decremented. Write FF_H to the THLPC[7:0] bits after reading from the RSCAN0THLACCm register.

Write FF_H only when the THLE bit in the RSCAN0THLCCm register is set to 1 (transmit history buffer is used) and the THLEMP flag in the RSCAN0THLSTSm register is 0.

chanhvinh.com

21.3.61 RSCAN0GTSTCFG — Global Test Configuration Register

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + 0468_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	C1ICBCE	C0ICBCE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 21.80 RSCAN0GTSTCFG register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	These bits are always read as 0. The write value should always be 0.
1	C1ICBCE	CAN1 Inter-Channel Communication Test Enable 0: CAN1 inter-channel communication test is disabled. 1: CAN1 inter-channel communication test is enabled.
0	C0ICBCE	CAN0 Inter-Channel Communication Test Enable 0: CAN0 inter-channel communication test is disabled. 1: CAN0 inter-channel communication test is enabled.

Modify the RSCAN0GTSTCFG register only in global test mode.

C1ICBCE Bit

Setting this bit to 1 enables the channel 1 inter-channel communication test.

C0ICBCE Bit

Setting this bit to 1 enables the channel 0 inter-channel communication test.

chanhvinh.com

21.3.62 RSCAN0GTSTCTR — Global Test Control Register

Access: Can be read/written in 8-, 16-, and 32-bit units

Address: <RSCAN0_base> + $046C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICBCTME
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 21.81 RSCAN0GTSTCTR register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	These bits are always read as 0. The write value should always be 0.
0	ICBCTME	Communication Test between Channels Enable 0: Communication test between channels disabled 1: Communication test between channels enabled

ICBCTME Bit

When this bit is set to 1, a communication test is enabled between the channels for which the CmICBCE bit (m = 0 or 1) in the RSCAN0GTSTCFG register has been set to 1. Modify the ICBCTME bit only in global test mode.

chanhvinh.com

21.3.63 RSCAN0GLOCKK — Global Lock Key Register

Access: Can be written in 16- and 32-bit units

Address: <RSCAN0_base> + $047C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LOCK[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1

Note 1. Writing to these bits is effective only when the RS-CAN module is in global test mode.

Table 21.82 RSCAN0GLOCKK register contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, an undefined value is returned. The write value should be 0.
15 to 0	LOCK[15:0]	Lock Key These bits are key bits to release protection of test mode.

The RSCAN0GLOCKK register releases protection of special test bits and is write-only.

LOCK[15:0] Bits

Writing the protection release data to the LOCK[15:0] bits in succession enables writing 1 to the RTME bit in the RSCAN0GTSTCTR register.

After the protection has been released, writing to the I/O register area (<RSCAN0_base> + 0000_H to <RSCAN0_base> + $04FF_H$) of the CAN (except the RAM) enables the protection again.

Reading from the I/O register area of the CAN or reading from/wring to other areas does not enable the protection.

chanhvinh.com

21.4 Interrupt Sources

The RS-CAN module has 8 interrupts that are grouped into global interrupts and channel interrupts.

Global interrupts [2 sources: common among channels]:

- Receive FIFO interrupt [1 source: common among channels]
- Global error interrupt [1 source: common among channels]

Channel interrupts [6 sources: 3 sources × number of channels]:

- CANm transmit interrupt [1 source for each channel]
  - CANm transmit complete interrupt
  - CANm transmit abort interrupt
  - CANm transmit/receive FIFO transmit complete interrupt (in transmit mode, gateway mode)
  - CANm transmit history interrupt
  - CANm transmit queue Interrupt
- CANm transmit/receive FIFO receive complete interrupt (in transmit mode, gateway mode) [1 source for each channel]
- CANm error interrupt [1 source for each channel]

  (m = 0 or 1)

When an interrupt request is generated, the corresponding interrupt request flag is set to 1 (interrupt request present). In that case, when the interrupt enable bit is set to 1 (enabling interrupts), an interrupt request is output from the RS-CAN module. (Generation of interrupts also depends on the interrupt control register settings of the interrupt controller.)

Setting the interrupt request flag to 0 (no interrupt request present) or setting the interrupt enable bit to 0 (disabling interrupts) clears the current interrupt request. The current interrupt request is still output until the interrupt request flag is cleared.

Table 21.83 lists the CAN interrupt sources. Figure 21.2 shows the CAN global interrupt block diagram. Figure 21.3 shows the CAN channel interrupt block diagram.

chanhvinh.com

Table 21.83 List of CAN Interrupt Sources

Interrupt Source			Corresponding Interrupt Request Flag	Corresponding Interrupt Enable Bit
Global interrupts	Receive FIFO	Receive FIFO 0	RFIF in the RSCAN0RFSTS0 register	RFIE in the RSCAN0RFCC0 register
		Receive FIFO 1	RFIF in the RSCAN0RFSTS1 register	RFIE in the RSCAN0RFCC1 register
		Receive FIFO 2	RFIF in the RSCAN0RFSTS2 register	RFIE in the RSCAN0RFCC2 register
		Receive FIFO 3	RFIF in the RSCAN0RFSTS3 register	RFIE in the RSCAN0RFCC3 register
		Receive FIFO 4	RFIF in the RSCAN0RFSTS4 register	RFIE in the RSCAN0RFCC4 register
		Receive FIFO 5	RFIF in the RSCAN0RFSTS5 register	RFIE in the RSCAN0RFCC5 register
		Receive FIFO 6	RFIF in the RSCAN0RFSTS6 register	RFIE in the RSCAN0RFCC6 register
		Receive FIFO 7	RFIF in the RSCAN0RFSTS7 register	RFIE in the RSCAN0RFCC7 register
	Global error		• DEF in the RSCAN0GERFL register • MES in the RSCAN0GERFL register • THLES in the RSCAN0GERFL register	• DEIE in the RSCAN0GCTR register • MEIE in the RSCAN0GCTR register • THLEIE in the RSCAN0GCTR register
Channel interrupts (m = 0 or 1)	CANm transmit	CANm transmit complete	TMTRF[1:0] in the RSCAN0TMSTSp register	TMIE in the RSCAN0TMIECy register
		CANm transmit abort	TMTRF[1:0] in the RSCAN0TMSTSp register	TAIE in the RSCAN0CmCTR register
		CANm transmit/receive FIFO transmit	CFTXIF in the RSCAN0CFSTSk register	CFTXIE in the RSCAN0CFCCk register
		CANm transmit queue	TXQIF in the RSCAN0TXQSTSm register	TXQIE in the RSCAN0TXQCCm register
		CANm transmit history	THLIF in the RSCAN0THLSTSm register	THLIE in the RSCAN0THLCCm register
	CANm transmit/receive FIFO receive		CFRXIF in the RSCAN0CFSTSk register	CFRXIE in the RSCAN0CFCCk register
	CANm error		• BEF in the RSCAN0CmERFL register • ALF in the RSCAN0CmERFL register • BLF in the RSCAN0CmERFL register • OVLF in the RSCAN0CmERFL register • BORF in the RSCAN0CmERFL register • BOEF in the RSCAN0CmERFL register • EPF in the RSCAN0CmERFL register • EWF in the RSCAN0CmERFL register	• BEIE in the RSCAN0CmCTR register • ALIE in the RSCAN0CmCTR register • BLIE in the RSCAN0CmCTR register • OLIE in the RSCAN0CmCTR register • BORIE in the RSCAN0CmCTR register • BOEIE in the RSCAN0CmCTR register • EPIE in the RSCAN0CmCTR register • EWIE in the RSCAN0CmCTR register

chanhvinh.com

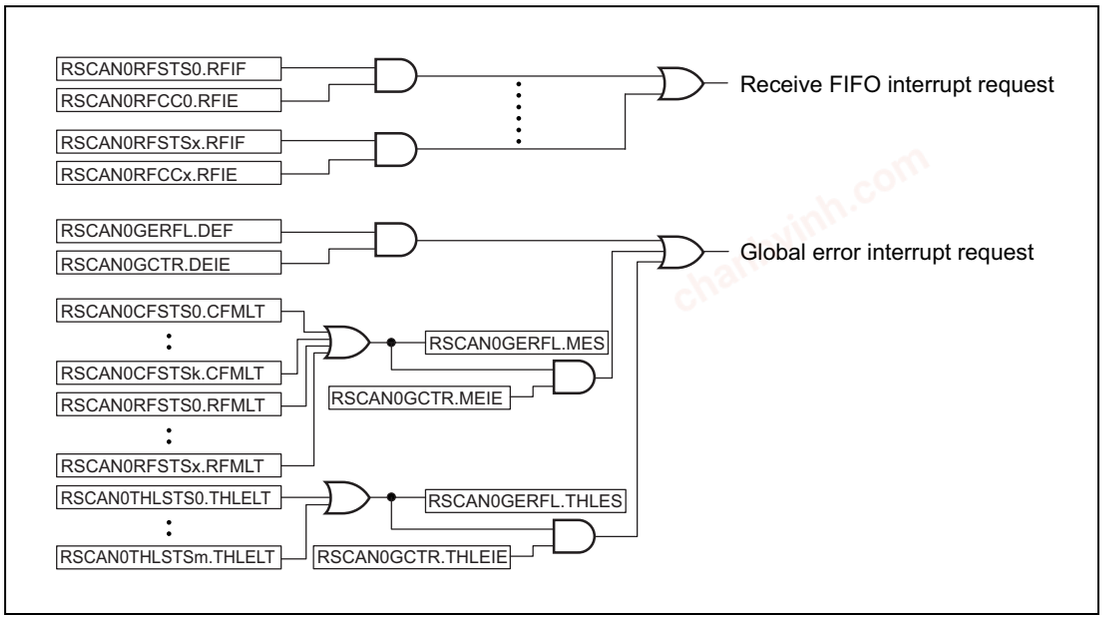


Figure 21.2 CAN Global Interrupt Block Diagram

chanhvinh.com

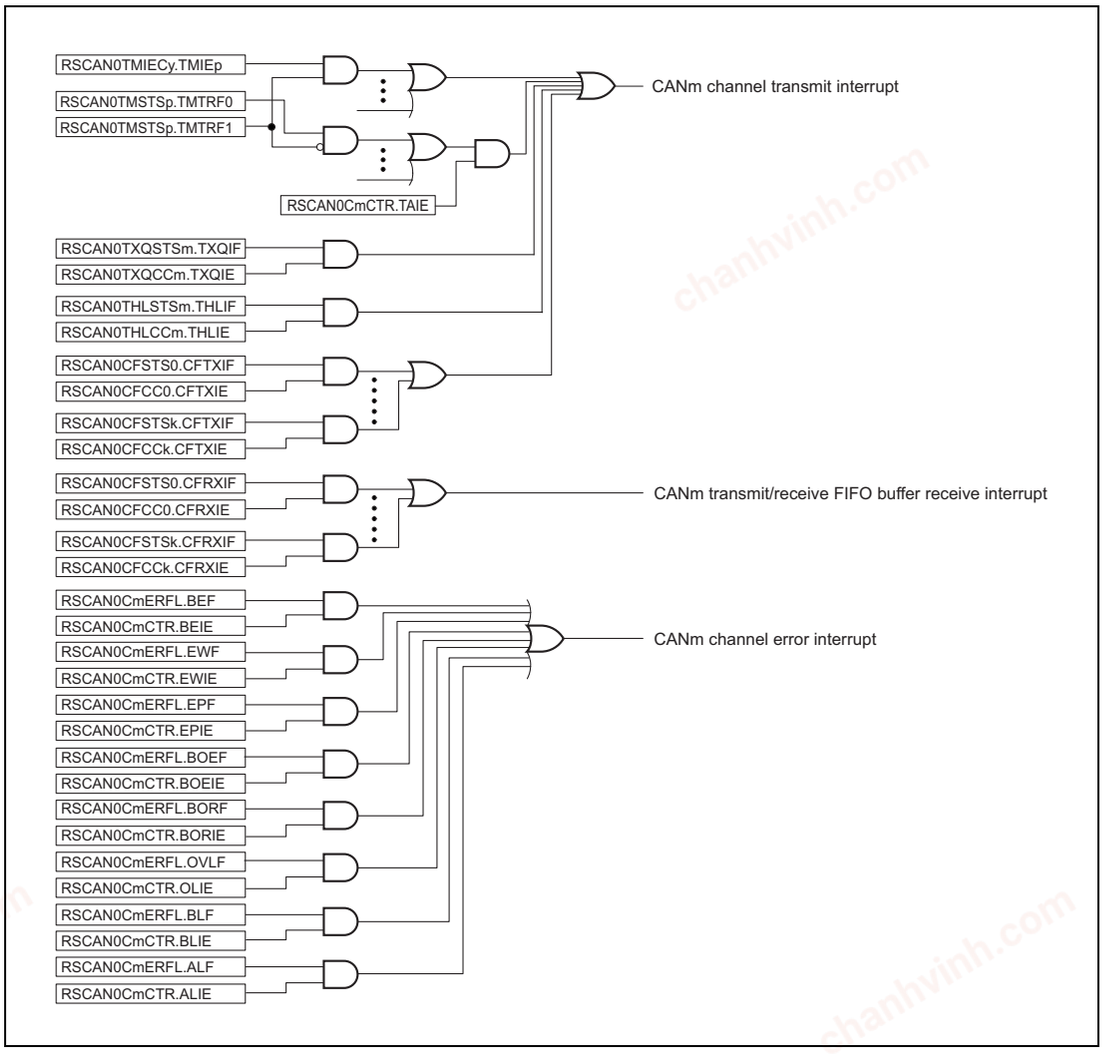


Figure 21.3 CAN Channel Interrupt Block Diagram

chanhvinh.com

21.5 RSCAN Modes

The RS-CAN module has four global modes to control the entire RS-CAN module status and four channel modes to control individual channel status. Details of global modes are described in Section 21.5.1, Global Modes, and details of channel modes are described in Section 21.5.2, Channel Modes.

- Global stop mode: Stops the clocks of the entire module to achieve low power consumption.
- Global reset mode: Performs initial settings for the entire module.
- Global test mode: Performs test settings.
- Global operating mode: Makes the entire module operable.
- Channel stop mode: Stops the channel clock.
- Channel reset mode: Performs initial settings for the channels.
- Channel halt mode: Stops CAN communication and allows channel testing.
- Channel communication mode: Performs CAN communication.

21.5.1 Global Modes

Figure 21.4 shows the transitions of global modes.

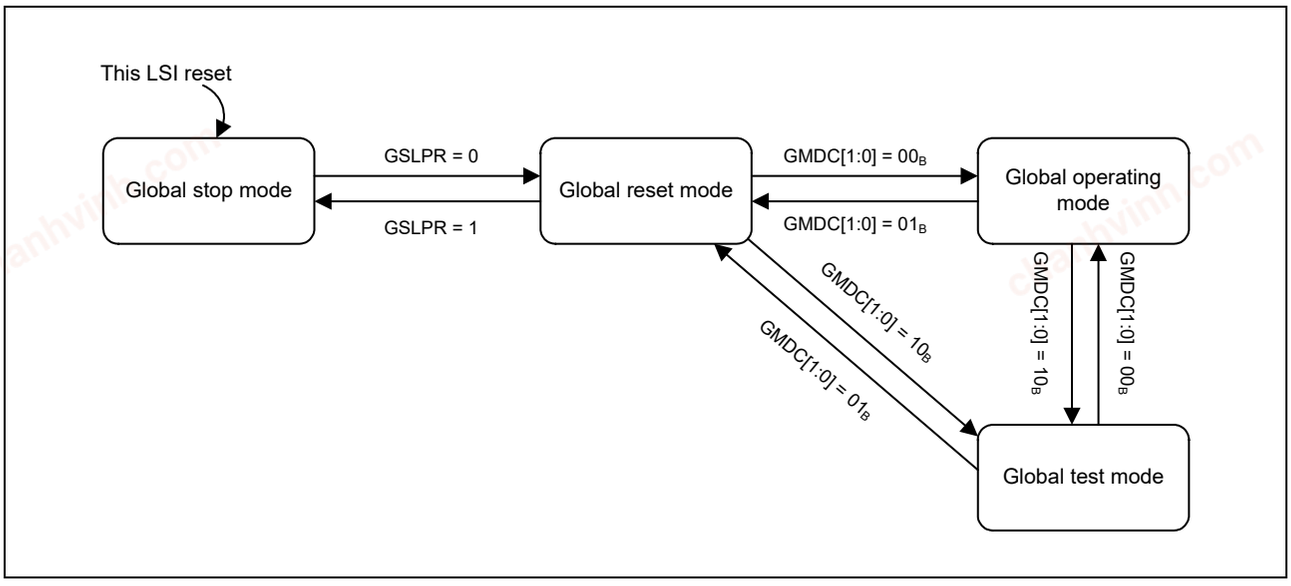


Figure 21.4 Transitions of Global Modes

chanhvinh.com

In some cases, global mode transitions also force channel mode transitions. Table 21.84 shows the channel mode transitions depending on the global mode setting dictated by the GMDC[1:0] bits and the GSLPR bit.

Table 21.84 Transitions of Channel Modes Depending on Global Mode Setting (GMDC[1:0] and GSLPR Bits)

Channel Mode before Setting	Channel Mode after Setting			
	GMDC[1:0] = 00_B GSLPR = 0 (Global Operation)	GMDC[1:0] = 10_B GSLPR = 0 (Global Test)	GMDC[1:0] = 01_B GSLPR = 0 (Global Reset)	GMDC[1:0] = 01_B GSLPR = 1 (Global Stop)
Channel communication	Channel communication	Channel halt	Channel reset	Transition prohibited
Channel halt	Channel halt	Channel halt	Channel reset	Transition prohibited
Channel reset	Channel reset	Channel reset	Channel reset	Channel stop
Channel stop	Channel stop	Channel stop	Channel stop	Channel stop

Note: GMDC[1:0], GSLPR: Bits in the RSCAN0GCTR register

Table 21.85 shows the global mode transition time.

Table 21.85 Global Mode Transition Time

Mode before Transition	Mode after Transition	Maximum Transition Time
Global stop	Global reset	Three pclk cycles
Global reset	Global stop	Three pclk cycles
Global reset	Global test	Ten pclk cycles
Global reset	Global operating	Ten pclk cycles
Global test	Global reset	Three pclk cycles
Global test	Global operating	Three pclk cycles
Global operating	Global reset	Three pclk cycles
Global operating	Global test	Two CAN frames*1

Note 1. CAN frame time of the lowest communication speed of the channels in use

chanhvinh.com

21.5.1.1 Global Stop Mode

In global stop mode, clocks of the CAN do not run and therefore power consumption is reduced. CAN registers can be read, but writing data to them is prohibited. Register values are retained. Only the clock used by the CPU for writing to the GSLPR bit runs in this mode.

After this LSI is reset, the CAN module transitions to global stop mode. Setting the GSLPR bit in the RSCAN0GCTR register to 1 (in global stop mode) in global reset mode sets the CSLPR bit in each of the RSCAN0CmCTR register to 1 (channel stop mode). If all channels are forced to transition to channel stop mode, the CAN module transitions to global stop mode. The GSLPR bit should not be modified in global operating mode or global test mode.

21.5.1.2 Global Reset Mode

In global reset mode, RS-CAN module settings are performed. When the RS-CAN module transitions to global reset mode, some registers are initialized. Table 21.88 and Table 21.89 list the registers to be initialized.

Setting the GMDC[1:0] bits in the RSCAN0GCTR register to 01_B sets the CHMDC[1:0] bits in each of the RSCAN0CmCTR registers (m = 0 or 1) to 01_B (channel reset mode). If all channels are forced to transition to channel reset mode, the CAN module transitions to global reset mode. Channels that are already in channel reset mode or channel stop mode do not transition (because the CHMDC[1:0] bits have already been set to 01_B).

21.5.1.3 Global Test Mode

In global test mode, settings for test-related registers are performed. When the CAN module transitions to global test mode, all CAN communications are disabled.

Setting the GMDC[1:0] bits in the RSCAN0GCTR register to 10_B sets the CHMDC[1:0] bits in each of the RSCAN0CmCTR register to 10_B (channel halt mode). If all channels are forced to transition to channel halt mode, the CAN module transitions to global test mode. Channels that are in channel stop mode, channel reset mode, or channel halt mode do not transition.

21.5.1.4 Global Operating Mode

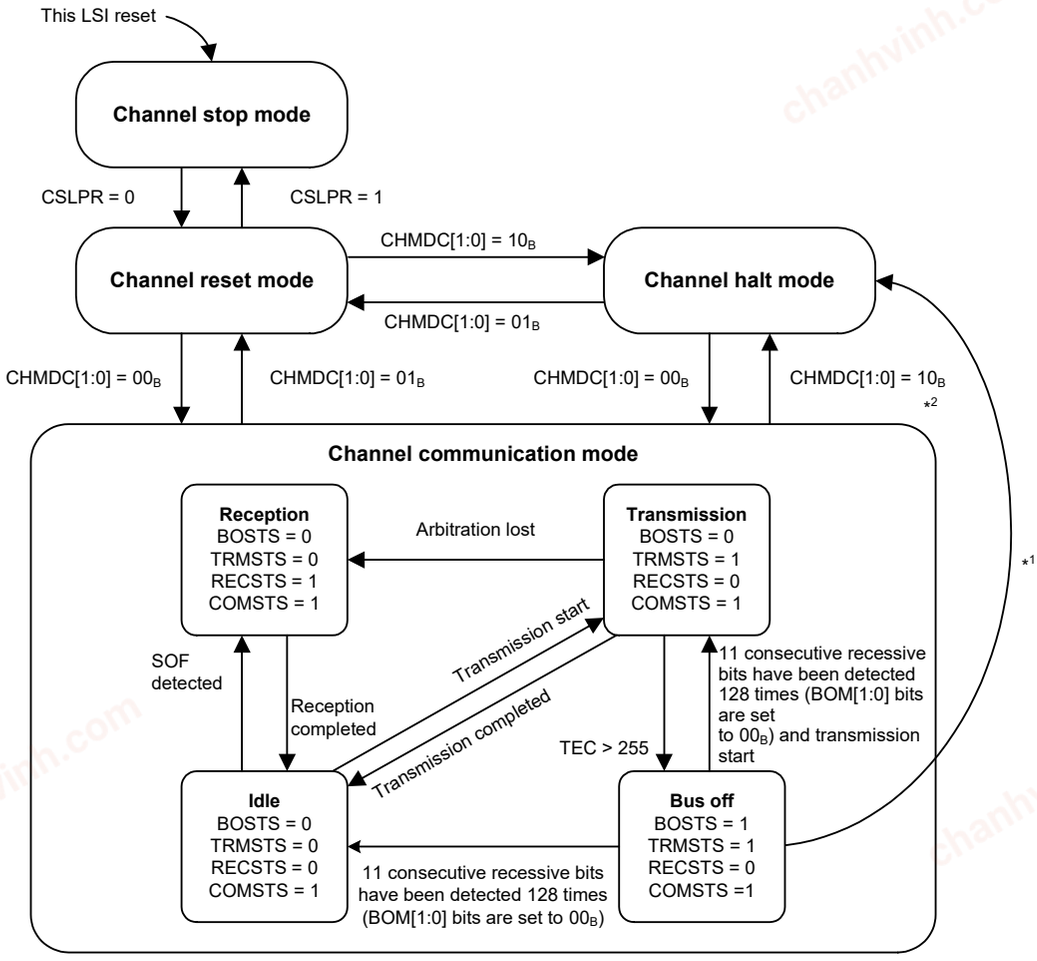
The RS-CAN module operates in global operating mode.

When the GMDC[1:0] bits in the RSCAN0GCTR register are set to 00_B, the RS-CAN module transitions to global operating mode.

chanhvinh.com

21.5.2 Channel Modes

Figure 21.5 shows a channel mode state transition chart. Table 21.86 shows the channel mode transition time.



CHMDC[1:0], CSLPR, BOM[1:0]: Bits in the RSCAN0CmCTR register (m = 0 or 1)
BOSTS, TRMSTS, RECSTS, COMSTS: Bits in the RSCAN0CmSTS register

Note 1. Timing of transition from bus off state to channel halt mode
- When BOM[1:0] = 01_B: Transition to channel halt mode when TEC exceeds 255
- When BOM[1:0] = 10_B: Transition to channel halt mode when 11 consecutive recessive bits have been detected 128 times
- When BOM[1:0] = 11_B: Transition to channel halt mode when the CHMDC[1:0] bits are set to 10_B

Note 2. While the CAN bus is locked at the dominant level, transition to channel halt mode is not made. In that case, enter channel reset mode.

Figure 21.5 Channel Mode State Transition Chart

chanhvinh.com

Table 21.86 Channel Mode Transition Time

Mode before Transition	Mode after Transition	Maximum Transition Time
Channel stop	Channel reset	Three pclk cycles
Channel reset	Channel stop	Three pclk cycles
Channel reset	Channel halt	Three CANm bit times
Channel reset	Channel communication	Two CANm bit times
Channel halt	Channel reset	Three pclk cycles
Channel halt	Channel communication	Three CANm bit times
Channel communication	Channel reset	Three pclk cycles
Channel communication	Channel halt	Two CANm frames

21.5.2.1 Channel Stop Mode

In channel stop mode, clocks are not supplied to channels and therefore power consumption is reduced. CAN registers can be read, but writing data to them is prohibited. Register values are retained.

Each channel enters channel stop mode after this LSI is reset. Channels also transition to channel stop mode when the CSLPR bit in the RSCAN0CmCTR register (m = 0 or 1) is set to 1 (channel stop mode) in channel reset mode. The CSLPR bit should not be modified in channel communication mode and channel halt mode.

21.5.2.2 Channel Reset Mode

In channel reset mode, channel settings are performed. When a channel transitions to channel reset mode, some channel-related registers are initialized. Table 21.88 lists the registers to be initialized.

When the CHMDC[1:0] bits in the RSCAN0CmCTR register are set to 01_B (channel reset mode) during CAN communication, communication is terminated before it is completed and the channel transitions to channel reset mode. Table 21.87 shows the operation when the CHMDC[1:0] bits are set to 01_B (channel reset mode) during CAN communication.

chanhvinh.com

21.5.2.3 Channel Halt Mode

In channel halt mode, settings for test-related registers of channels are performed. When a channel transitions to channel halt mode, CAN communication of the channel stops.

Table 21.87 shows operation when the CHMDC[1:0] bits are set to 10_B (channel halt mode) during CAN communication.

Table 21.87 Operation a Channel Transitions to Channel Reset Mode/Channel Halt Mode

Mode	During Reception	During Transmission	Bus Off State
Channel reset (CHMDC[1:0] = 01_B)	Transitions to channel reset mode before reception is completed.<asterisk>*<Super_A>1	Transitions to channel reset mode before transmission is completed.<asterisk>*<Super_A>1	Transitions to channel reset mode before bus off recovery.
Channel halt<asterisk>*<Super_A>3 (CHMDC[1:0] = 10_B)	Transitions to channel halt mode after reception is completed.<asterisk>*<Super_A>2	Transitions to channel halt mode after transmission is completed.	[When BOM[1:0] = 00_B] Transitions to channel halt mode (CHMDC[1:0] = 10_B) only after bus off recovery. [When BOM[1:0] = 01_B] Transitions to channel halt mode automatically when the condition for transition to bus off state is met. [When BOM[1:0] = 10_B] Transitions to channel halt mode automatically after bus off recovery. [When BOM[1:0] = 11_B] Transitions to channel halt mode immediately after the CHMDC[1:0] bits are set to 10_B before bus off recovery.

Note 1. To allow transition to channel reset mode after communication is completed, set the CHMDC[1:0] bits to 10_B and confirm that communication has been completed and transition to channel halt mode has been made, and then set the CHMDC[1:0] bits to 01_B.

Note 2. While the CAN bus is locked at the dominant level, transition to channel halt mode is not made. In that case, enter channel reset mode. The CAN bus status can be confirmed with the BLF flag of the RSCAN0CmERFL register that becomes 1 when dominant lock is detected.

Note 3. When the transition from channel reset mode to channel wait mode is to be made, set the RSCAN0CmCFG register in channel reset mode and then shift to channel wait mode.

21.5.2.4 Channel Communication Mode

In channel communication mode, CAN communication is performed. Each channel has the following communication states during CAN communication.

- Idle : Neither reception nor transmission is in progress.
- Reception : Receiving a message sent from another node.
- Transmission : Transmitting a message.
- Bus off : Isolated from CAN communication.

When the CHMDC[1:0] bits in the RSCAN0CmCTR register are set to 00_B, the channel transitions to channel communication mode. After that, once 11 consecutive recessive bits have been detected, the COMSTS flag in the RSCAN0CmSTS register (m = 0 or 1) is set to 1 (communication is ready) and transmission and reception are enabled on the CAN network as an active node. At this time, transmission and reception of messages can be started.

chanhvinh.com

21.5.2.5 Bus Off State

A channel transitions to the bus off state according to the transmit/receive error counter increment/decrement rules of the CAN specifications.

The conditions for returning from the bus off state are determined by the BOM[1:0] bits in the RSCAN0CmCTR register.

- When BOM[1:0] = 00_B:
  Bus off recovery is compliant with the CAN specifications. After 11 consecutive recessive bits have been detected 128 times, a channel returns from the bus off state to the CAN communication ready state (error active state). At that time, the TEC[7:0] and REC[7:0] bits in the RSCAN0CmSTS register are initialized to 00_H and the BORF flag in the RSCAN0CmERFL register is set to 1 (bus off recovery is detected). When the CHMDC[1:0] bits in the RSCAN0CmCTR register are set to B'10 (channel halt mode) in the bus off state, the channel transitions to channel halt mode after bus off recovery has been completed (11 consecutive recessive bits have been detected 128 times).
- When BOM[1:0] = 01_B:
  When a channel transitions to the bus off state, the CHMDC[1:0] bits are set to 10_B and the channel transitions to channel halt mode. At that time, the TEC[7:0] and REC[7:0] bits are initialized to 00_H but the BORF flag is not set to 1.
- When BOM[1:0] = 10_B:
  When a channel has transitioned to the bus off state, the CHMDC[1:0] bits are set to 10_B. After bus off recovery has been completed (11 consecutive recessive bits have been detected 128 times), the channel transitions to channel halt mode. At that time, the TEC[7:0] and REC[7:0] bits are initialized to 00_H and the BORF flag is set to 1.
- When BOM[1:0] = 11_B:
  When the CHMDC[1:0] bits are set to 10_B in the bus off state, the channel transitions to channel halt mode before bus off recovery is completed. At that time, the TEC[7:0] and REC[7:0] bits are initialized to 00_H but the BORF flag is not set to 1.
  However, the BORF flag becomes 1 if a CAN module transitions to error active state (by detecting 128 times of 11 consecutive recessive bits) before CHMDC[1:0] bits are set to 10_B.

If the RS-CAN module causes the channel to transition to channel halt mode simultaneously with a program write to the CHMDC[1:0] bits, the program write takes precedence. An automatic transition to channel halt mode when the BOM[1:0] bits are set to 01_B or 10_B is made only when the CHMDC[1:0] bits are 00_B (channel communication mode). Furthermore, setting the RTBO bit in the RSCAN0CmCTR register to 1 allows a forced return from the bus off state. As soon as the RTBO bit is set to 1, the state changes to the error active state. After 11 consecutive recessive bits have been detected, the CAN module becomes ready for communication. In this case, the BORF flag is not set to 1 and the TEC[7:0] and REC[7:0] bits are initialized to 00_H. Write 1 to the RTBO bit only when the BOM[1:0] value is 00_B.

chanhvinh.com

Table 21.88 Registers Initialized in Global Reset Mode or Channel Reset Mode

Register	Bit / Flag
RSCAN0CmCTR register	CTMS[1:0], CTME, CHMDC[1:0]
RSCAN0CmSTS register	CHLTSTS, EPSTS, BOSTS, TRMSTS, RECSTS, COMSTS, REC[7:0], TEC[7:0]
RSCAN0CmERFL register	CRCREG[14:0], ADERR, B0ERR, B1ERR, CERR, AERR, FERR, SERR, ALF, BLF, OVLF, BORF, BOEF, EPF, EWF, BEF
RSCAN0CFCCk register	When transmit/receive FIFO buffer is in transmit mode or gateway mode: CFE
RSCAN0CFSTSk register	When transmit/receive FIFO buffer is in transmit mode or gateway mode: CFMC[7:0], CFFLL, CFEMP, CFMLT, CFRXIF, CFTXIF
RSCAN0CFTISTS register	CFkTXIF
RSCAN0TMCp register	TMOM, TMTAR, TMTR
RSCAN0TMSTSp register	TMTARM, TMTRM, TMTRF[1:0], TMTSTS
RSCAN0TMTRSTSy register	TMTRSTSp (Bits of corresponding channel are initialized in channel reset mode.)
RSCAN0TMTARSTSy register	TMTARSTSp (Bits of corresponding channel are initialized in channel reset mode.)
RSCAN0TMTCSTSy register	TMTCSTSp (Bits of corresponding channel are initialized in channel reset mode.)
RSCAN0TMTASTSy register	TMTASTSp (Bits of corresponding channel are initialized in channel reset mode.)
RSCAN0TXQCCm register	TXQE
RSCAN0TXQSTSm register	TXQIF, TXQFLL, TXQEMP
RSCAN0THLCCm register	THLE
RSCAN0THLSTSm register	THLMC[4:0], THLIF, THLELT, THLFLL, THLEMP
RSCAN0GTINTSTS0 register	TSIFm, TAIFm, TQIFm, CFTIFm, THIFm (m = 0 or 1)

Table 21.89 Registers Initialized Only in Global Reset Mode

Register	Bit / Flag
RSCAN0GSTS register	GHLTSTS
RSCAN0GERFL register	THLES, MES, DEF
RSCAN0GTSC register	TS[15:0]
RSCAN0RMNDy register	RMNSq
RSCAN0RFCCx register	RFE
RSCAN0RFSTSx register	RFMC[7:0], RFIF, RFMLT, RFFLL, RFEMP
RSCAN0CFCCk register	When transmit/receive FIFO buffer is in receive mode: CFE
RSCAN0CFSTSk register	When transmit/receive FIFO buffer is in receive mode: CFMC[7:0], CFFLL, CFEMP, CFTXIF, CFRXIF, CFMLT
RSCAN0FESTS register	CFkEMP, RFxEMP
RSCAN0FFSTS register	CFkFLL, RFxFLL
RSCAN0FMSTS register	CFkMLT, RFxMLT
RSCAN0RFISTS register	RFxIF
RSCAN0CFRISTS register	CFkRXIF
RSCAN0GTSTCFG register	C0ICBCE, C1ICBCE
RSCAN0GTSTCTR register	ICBCTME

chanhvinh.com

21.6 Reception Function

There are two reception types.

- Reception by receive buffers:
  Zero to 31 receive buffers can be shared by all channels. Since messages stored in receive buffers are overwritten at each reception, the latest receive data can always be read.
- Reception by receive FIFO buffers and transmit/receive FIFO buffers (receive mode):
  Eight receive FIFO buffers can be shared by all channels and three dedicated transmit/receive FIFO buffers are provided for each channel. Messages of up to the number of buffer stages specified with the RFDC[2:0] and CFDC[2:0] bits can be stored in FIFO buffers and can be read sequentially from the oldest.

21.6.1 Data Processing Using the Receive Rule Table

Data processing using the receive rule table allows dispatching of selected messages to the specified buffer. Data processing includes acceptance filter processing, DLC filter processing, routing processing, label addition processing, and mirror function processing.

Up to 128 receive rules can be registered per channel and up to (64 × number of channels) total receive rules can be registered in the entire module. (Up to 128 receive rules can be registered in this module that has two channels.) Set receive rules for each channel. Receive rules cannot be shared with other channels. If receive rules are not set, no messages can be received. Figure 21.6 illustrates how receive rules are registered.

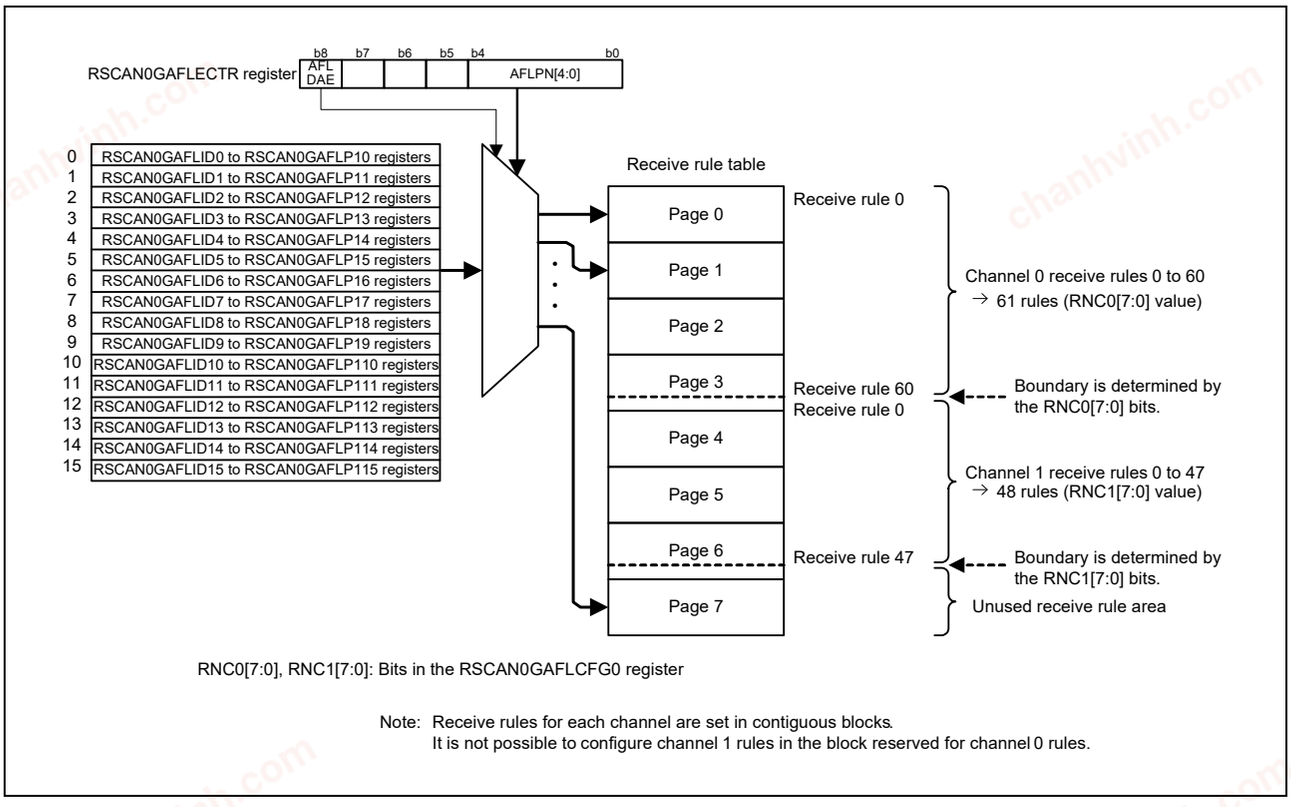


Figure 21.6 Entry of Receive Rules (for Setting Channel 0 and 1)

chanhvinh.com

Each receive rule consists of 16 bytes in the RSCAN0GAFLIDj, RSCAN0GAFLMj, RSCAN0GAFLP0j, and RSCAN0GAFLP1j registers (j = 0 to 15). The RSCAN0GAFLIDj register (j = 0 to 15) is used to set GAFLID, GAFLIDE bit, GAFLRTR bit, and the mirror function, the RSCAN0GAFLMj register is used to set mask, the RSCAN0GAFLP0j register is used to set label information to be added, DLC value, and storage receive buffer, and the RSCAN0GAFLP1j register is used to set storage FIFO buffer. Up to 16 receive rules can be set per page.

21.6.1.1 Acceptance Filter Processing

In the acceptance filter processing, the ID data, IDE bit, and RTR bit in a received message are compared with the ID data, IDE bit, and RTR bit set in the receive rule of the corresponding channel. When all these bits match, the message passes through the acceptance filter processing. The ID data, IDE bit, and RTR bit in the received message which correspond to the bits set to 0 (bits are not compared) in the RSCAN0GAFLMj register are not compared and are regarded as matched.

Check begins with the receive rule of the minimum number for the corresponding channel. When all the bits to be compared in a received message match the bits set in the receive rule or when all the receive rules are compared without any match, filter processing stops. If there is no matching receive rule, the received message is not stored in the receive buffer or FIFO buffer.

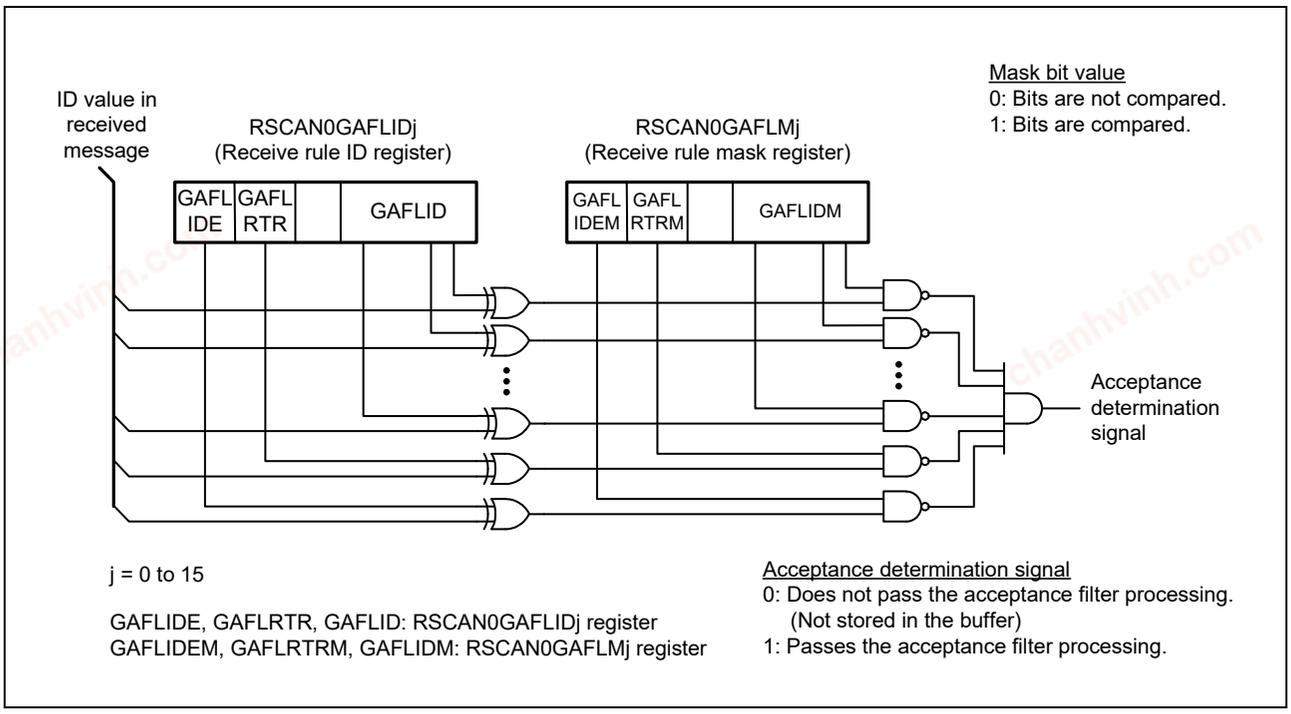


Figure 21.7 Acceptance Filter Function

chanhvinh.com

21.6.1.2 DLC Filter Processing

When the DCE bit in the RSCAN0GCFG register is set to 1 (DLC check is enabled), DLC filter processing is added to messages that passed through the acceptance filter processing. When the DLC value in a message is equal to or larger than the DLC value set in the receive rule, the message passes through the DLC filter processing.

When a message has passed through the DLC filter processing with the DRE bit in the RSCAN0GCFG register set to 0 (DLC replacement is disabled), the DLC value in the received message is stored in the buffer. In this case, all the data bytes in the received message are stored in the buffer.

When a message has passed through the DLC filter processing with the DRE bit in the RSCAN0GCFG register set to 1 (DLC replacement is enabled), the DLC value in the receive rule is stored in the buffer instead of the DLC value in the received message. In this case, a value of 00_H is stored in each data byte beyond the number of bytes which is indicated by the DLC value in the receive rule.

When the DLC value in the received message is smaller than that in the receive rule, the message does not pass through the DLC filter processing. In this case, the message is not stored in the receive buffer or the FIFO buffer and the DEF flag in the RSCAN0GERFL register is set to 1 (a DLC error is present).

21.6.1.3 Routing Processing

Messages that passed through the acceptance filter processing and the DLC filter processing are stored in receive buffers, receive FIFO buffers, or transmit/receive FIFO buffers (set to receive mode or gateway mode). Message storage destination is set by the GAFLRMV and GAFLRMDP[6:0] bits in the RSCAN0GAFLP0j register (j = 0 to 15) and by the RSCAN0GAFLP1j register. Messages that passed through the acceptance filter processing and the DLC filter processing can be stored in up to eight buffers.

21.6.1.4 Label Addition Processing

It is possible to add 12-bit label information to messages that passed through the filter processing and store them in buffers. This label information is set in the GAFLPTR[11:0] bits in the RSCAN0GAFLP0j register.

21.6.1.5 Mirror Function Processing

The mirror function allows the CAN node to receive its own transmitted messages. The mirror function is made available by setting the MME bit in the RSCAN0GCFG register to 1 (mirror function is enabled).

When the mirror function is in use, receive rules for which the GAFLLB bit in the RSCAN0GAFLIDj register is set to 0 are used for data processing when receiving messages transmitted from other CAN nodes. When the CAN node is receiving its own transmitted messages, receive rules for which the GAFLLB bit is set to 1 are used for data processing.

chanhvinh.com

21.6.1.6 Timestamp

The timestamp counter is a 16-bit free-running counter used for recording message receive time. The timestamp counter value is fetched at the start-of-frame (SOF) timing of a message and is then stored in a receive buffer or a FIFO buffer together with the message ID and data. Ether pclk or the CANm bit time clock (m = 0 or 1) may be selected as a timestamp counter clock source using the TSBTCS[2:0] and TSSS bits in the RSCAN0GCFG register. The timestamp counter count source is obtained by dividing the selected clock source by the TSP[3:0] value in the RSCAN0GCFG register.

When the CANm bit time clock is used as a clock source, the timestamp counter stops when the corresponding channel transitions to channel reset mode or channel halt mode. When the pclk is used as a clock source, the timestamp function is not affected by channel mode.

The timestamp counter value is reset to 0000_H by setting the TSRST bit in the RSCAN0GCTR register to 1.

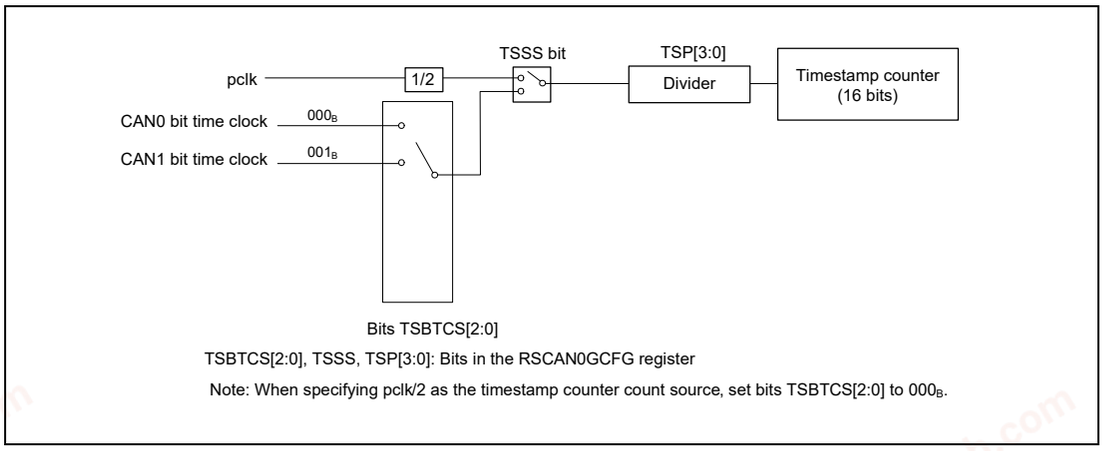


Figure 21.8 Timestamp Function Block Diagram

chanhvinh.com

21.7 Transmission Functions

There are three types of transmission.

- Transmission using transmit buffers:
  Each channel has 16 buffers.
- Transmission using transmit/receive FIFO buffers (transmit mode):
  Each channel has three FIFO buffers. Up to 96 messages can be contained in a single FIFO buffer. Each FIFO buffer is used with a link to a transmit buffer. Only the message to be transmitted next in a FIFO buffer becomes the target of transmit priority determination. Messages are transmitted sequentially on a first-in, first-out basis.
- Transmission using transmit queues:
  Up to 16 transmit buffers per channel can be allocated to the transmit queues. Transmit buffer ((16 × m) + 15) is used as an access window of a corresponding channel. Transmit buffers are allocated to transmit queues in descending order of buffer number. All messages in transmit queues, which are targets of priority determination, are transmitted in the order of ID number.

Figure 21.9 shows the allocation of transmit queues and transmit/receive FIFO buffer link.

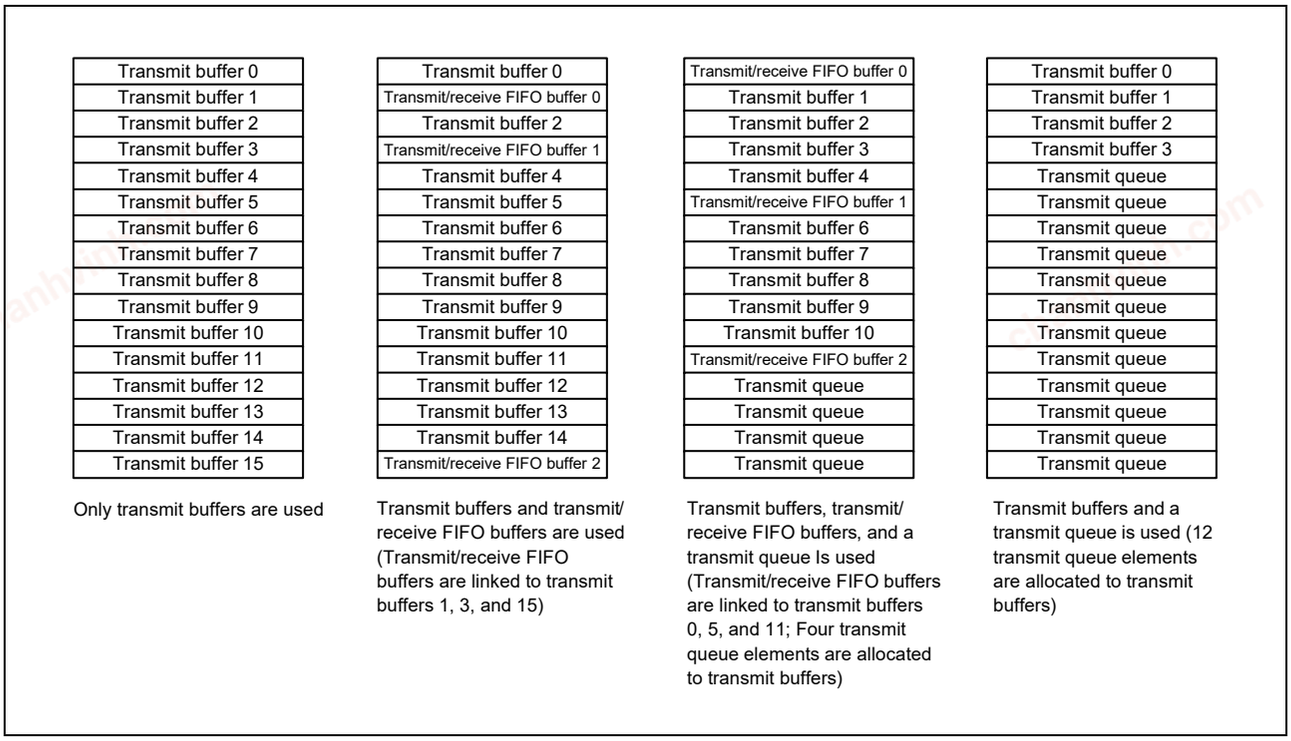


Figure 21.9 Allocation of Transmit Queues and Transmit/Receive FIFO Buffer Links

chanhvinh.com

21.7.1 Transmit Priority Determination

If transmit requests are issued from multiple buffers or from the queue on the same channel, transmit priority is determined using one of the following methods.

The priority is determined by using one of the following methods.

- ID priority (TPRI bit = 0)
- Transmit buffer number priority (TPRI bit = 1)

All CAN channels use the setting of the TPRI bit in the RSCAN0GCFG register.

When the TPRI bit is set to 0, messages are transmitted according to the priority of stored message IDs. ID priority conforms to the CAN bus arbitration specification defined in the CAN specifications. All IDs of pending transmit messages are targets of priority determination, regardless of whether they are stored in transmit buffers, transmit/receive FIFO buffers (set to transmit mode or gateway mode), or the transmit queue. If even a single transmit queue is used, select ID priority. When transmit/receive FIFO buffers are used, the oldest message in a FIFO buffer becomes the target of priority determination. When a message is being transmitted from a transmit/receive FIFO buffer, the next message in the FIFO buffer becomes the target of priority determination. When a transmit queue is used, all messages in the transmit queue are targets of priority determination. If the same ID is set for two or more buffers, the buffer with the smaller buffer number takes precedence.

When the TPRI bit is set to 1, the message in the transmit buffer with the minimum buffer number among all buffers with a transmit request is transmitted first. When transmit/receive FIFO buffers are linked to transmit buffers, transmit priority is determined according to linked transmit buffer numbers.

When messages are retransmitted due to an arbitration-lost or an error, transmit priority determination is made again regardless of the TPRI bit.

21.7.2 Transmission Using Transmit Buffers

Setting the transmit request bit (TMTR bit in the RSCAN0TMCp register) in a transmit buffer to 1 (transmission is requested) allows transmission of data frames or remote frames.

The transmit result is shown by the TMTRF[1:0] flag in the corresponding RSCAN0TMSTSp register (p = 0 to 31). When transmit completes successfully, the TMTRF[1:0] flag is set to 10_B (transmission has been completed (without transmit abort request)) or 11_B (transmission has been completed (with transmit abort request)).

21.7.2.1 Transmit Abort Function

With respect to transmit buffers for which the TMTRM bit in the RSCAN0TMSTSp register is set to 1 (a transmit request is present), when the TMTAR bit in the RSCAN0TMCp register is set to 1 (transmit abort is requested), the transmit request is canceled. When transmit abort is completed, the TMTRF[1:0] flag in the RSCAN0TMSTSp register is set to 01_B (transmit abort has been completed) and the transmit request is canceled (clearing the TMTRM bit to 0).

A message that is being transmitted or a message to be transmitted next according to the transmit priority determination cannot be aborted. However, when an arbitration-lost or an error occurs during transmission of a message for which the TMTAR bit is set to 1, retransmission is not performed.

chanhvinh.com

21.7.2.2 One-Shot Transmission Function (Retransmission Disabling Function)

When the TMOM bit in the RSCAN0TMCp register is set to 1 (one-shot transmission is enabled), transmission is performed only once. Even if an arbitration-lost or an error occurs, retransmission is not performed.

The one-shot transmit result is shown by the TMTRF[1:0] flag in the corresponding RSCAN0TMSTSp register. When one-shot transmission completes successfully, the TMTRF[1:0] flag is set to 10_B or 11_B. When an arbitration-lost or an error occurs, the TMTRF[1:0] flag is set to 01_B (transmit abort has been completed).

21.7.3 Transmission Using FIFO Buffers

Multiple messages can be stored in a single transmit/receive FIFO buffers, up to the number specified by the FIFO buffer depth, which is set by the CFDC[2:0] bits in the RSCAN0CFCCk register (k = 0 to 5). Messages are transmitted sequentially on a first-in, first-out basis.

Each transmit/receive FIFO buffer is linked to a transmit buffer selected by the CFTML[3:0] bits in the RSCAN0CFCCk register. When the CFE bit in the RSCAN0CFCCk register is set to 1 (transmit/receive FIFO buffers are used), transmit/receive FIFO buffers become targets of transmit priority determination. Priority of only the next transmit message is determined in the FIFO buffer.

When the CFE bit is set to 0 (no transmit/receive FIFO buffer is used), the CFEMP flag is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)) at the timing below.

- The transmit/receive FIFO buffer becomes empty immediately if the message in it is not being transmitted or is not to be transmitted next.
- The transmit/receive FIFO buffer becomes empty after transmission completion, CAN bus error detection, or arbitration-lost in the case that a message in it is being transmitted or to be transmitted next.

When the CFE bit is cleared to 0, all messages in transmit/receive FIFO buffers are lost and messages cannot be stored in FIFO buffers. Confirm that the CFEMP flag is set to 1 before setting the CFE bit to 1 again.

chanhvinh.com

21.7.3.1 Interval Transmission Function

A message transmission interval time can be set to space the transmission of messages from the same FIFO buffer when using a transmit/receive FIFO buffer set to transmit mode or gateway mode.

Immediately after the first message has been transmitted successfully from the FIFO buffer with the CFE bit in the RSCAN0CFCCk register set to 1, the interval timer starts counting (after EOF7 of the CAN protocol). After that, when the interval time has passed, the next message is transmitted. The interval timer stops in channel reset mode or by clearing the CFE bit to 0.

The interval time is set by the CFITT[7:0] bits in the RSCAN0CFCCk register. When the interval timer is not used, set the CFITT[7:0] bits to 00_H.

Select an interval timer count source using the CFITR and CFITSS bits in the RSCAN0CFCCk register. When the CFITR and CFITSS bits are set to 00_B, the count source is obtained by dividing pclk/2 by the value of the ITRCP[15:0] bits. When the CFITR and CFITSS bits are set to 10_B, the count source is obtained by dividing pclk/2 by (the value of the ITRCP[15:0] bits in the RSCAN0GCFG register × 10). When the CFITR and CFITSS bits are set to $x1_B$, the CANm bit time clock is used as a count source.

The interval time is calculated by the following equations where M is the value set to ITRCP[15:0] and N is the set CFITT[7:0] value.

- When CFITR and CFITSS = 00_B (fPBA is the frequency of pclk):

$$\frac{1}{fPBA} \times 2 \times M \times N$$

- When CFITR and CFITSS = 10_B:

$$\frac{1}{fPBA} \times 2 \times M \times 10 \times N$$

- When CFITR and CFITSS = $x1_B$ (fCANBIT is the frequency of CANm bit time clock):

$$\frac{1}{fCANBIT} \times N$$

Figure 21.10 shows the interval timer block diagram.

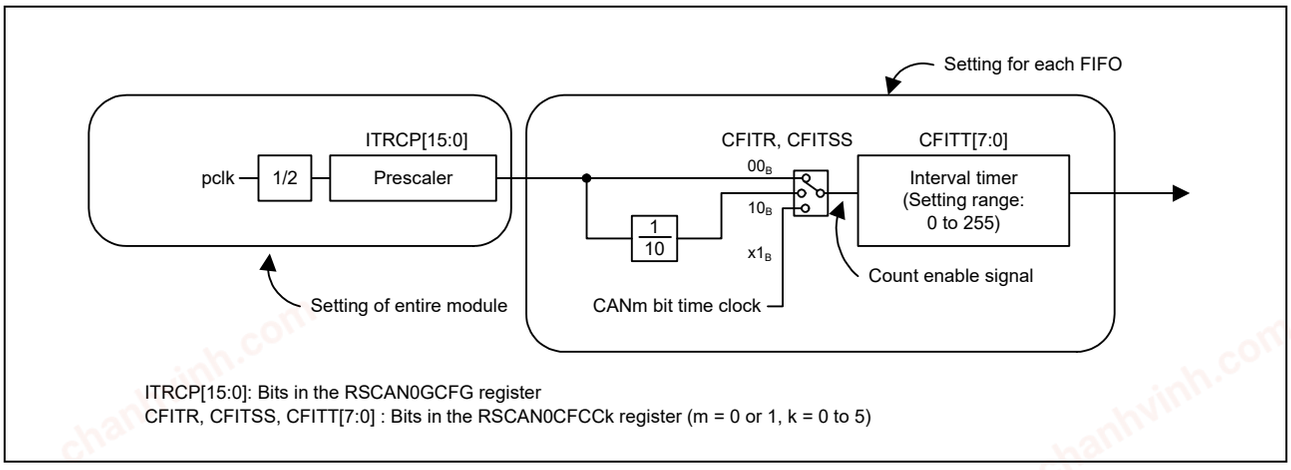


Figure 21.10 Interval Timer Block Diagram

chanhvinh.com

Figure 21.11 shows the interval timer timing diagram.

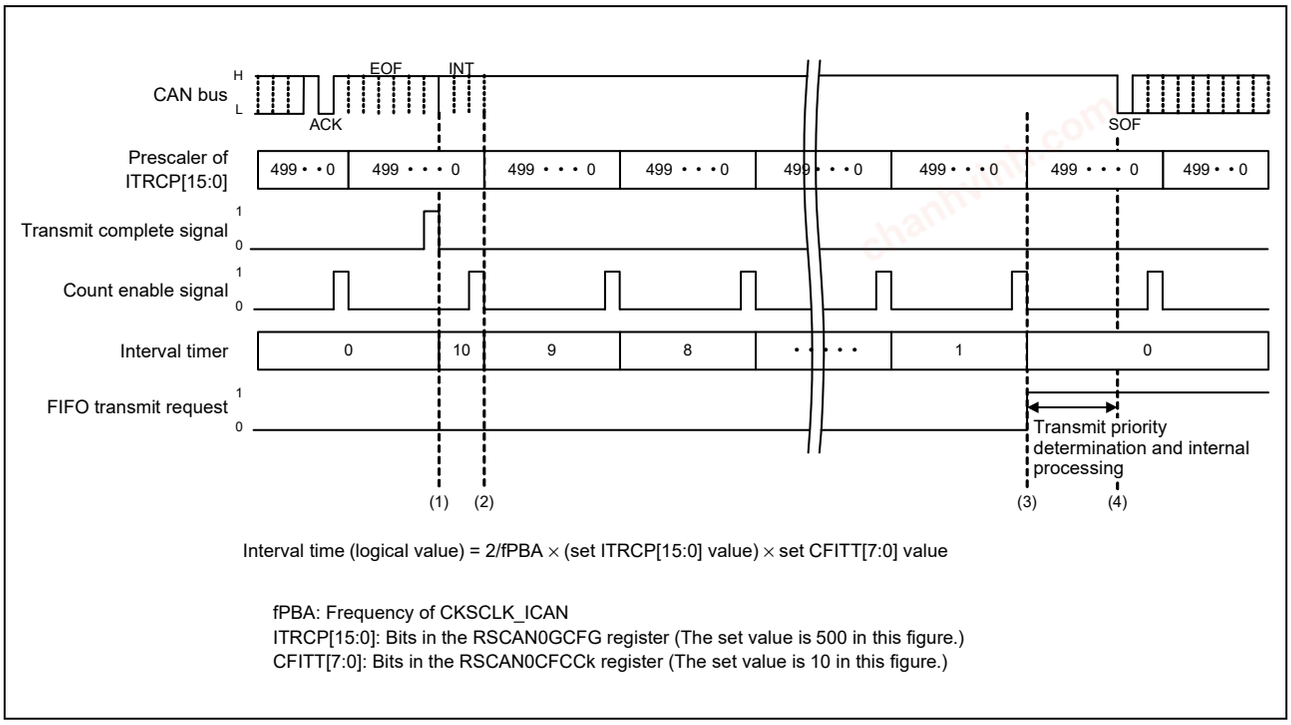


Figure 21.11 Interval Timer Timing Chart

(1) The interval timer starts counting upon completion of transmission. Since the prescaler is not initialized at the time of transmission completion, the first interval time contains an error of up to one count of the interval timer.

(2) The interval timer is decremented by the next count enable signal.

(3) When the interval timer has decreased to 0, the transmit/receive FIFO buffer issues a transmit request.

(4) The transmit/receive FIFO buffer is determined for the next transmission by the priority determination, it starts transmitting data. Transmission starts usually with a delay of three CANm bit time clock cycles or less from the issue of transmit request. If multiple internal processes (such as receive filter processing, message routing, and transmit priority determination) take place in all channels, a delay of up to 504 cycles of pclk may be generated.

chanhvinh.com

21.7.4 Transmission Using Transmit Queues

Three to sixteen buffers are allocated to a transmit queue for each channel, and transmit buffer ((16 × m) + 15) is used as an access window of a corresponding channel.

All messages in a transmit queue are targets of transmit priority determination and are transmitted in the ID priority order regardless of storage sequence. If two messages having the same ID are stored in a transmit queue, these messages are not always transmitted in the order of their storage in the transmit queue.

Setting the TXQE bit in the RSCAN0TXQCCm register to 0 disables transmit queues. When the TXQE bit is set to 0, the TXQEMP flag in the RSCAN0TXQSTSm register is set to 1 (the transmit queue contains no messages (transmit queue empty)) at the timing below.

- The transmit queue becomes empty immediately when no message in it is being transmitted or will be transmitted next.
- The transmit queue becomes empty after transmission completion, CAN bus error detection, or arbitration-lost when a message in it is being transmitted or will be transmitted next.

When the TXQE bit is cleared to 0, all messages in transmit queues are lost and messages cannot be stored in transmit queues. Confirm that the TXQEMP flag is set to 1 before setting the TXQE bit to 1 again.

21.7.5 Transmit History Function

Information about transmitted messages can be stored in the transmit history buffer. Each channel has a single transmit history buffer that can contain 16 sets of transmit history data.

A message transmit source buffer type can be selected by the THLDTE bit in the RSCAN0THLCCm register. The THLEN bit in the RSCAN0CFIDk register (k = 0 to 5) determines whether transmit history data is stored for each message.

The following information on a transmitted message will be stored in the transmission history buffer after the successful completion of transmission.

Storage of the transmission history data after the successful completion of transmission may take up to 144 cycles of pclk.

- Buffer type — 001_B: Transmit buffer
010_B: Transmit/receive FIFO buffer
100_B: Transmit queue
- Buffer number — Number of source transmit buffer, transmit queue, or transmit/receive FIFO buffer.
This number depends on buffer types. See Table 21.90.
- Label data — Label information of the transmit message

chanhvinh.com

Table 21.90 Transmit History Data Buffer Numbers

Buffer type / Buffer No.	001_B	010_B	100_B
0000_B	Transmit buffer 16 × m + 0	Buffer numbers of the transmit buffer linked to the transmit/ receive FIFO buffer by the CFTML[3:0] bits in the RSCAN0CFCCk register (k = 0 to 5)	Buffer numbers of the transmit buffer allocated to the transmit queue that performed transmission
0001_B	Transmit buffer 16 × m + 1		
0010_B	Transmit buffer 16 × m + 2		
0011_B	Transmit buffer 16 × m + 3		
0100_B	Transmit buffer 16 × m + 4		
0101_B	Transmit buffer 16 × m + 5		
0110_B	Transmit buffer 16 × m + 6		
0111_B	Transmit buffer 16 × m + 7		
1000_B	Transmit buffer 16 × m + 8		
1001_B	Transmit buffer 16 × m + 9		
1010_B	Transmit buffer 16 × m + 10		
1011_B	Transmit buffer 16 × m + 11		
1100_B	Transmit buffer 16 × m + 12		
1101_B	Transmit buffer 16 × m + 13		
1110_B	Transmit buffer 16 × m + 14		
1111_B	Transmit buffer 16 × m + 15		

Label data is used to identify each message. Unique label data can be added to each message transmitted from a transmit buffer, transmit queue, or transmit/receive FIFO buffer.

Transmit history data can be read from the RSCAN0THLACCm register. If an attempt is made to store new transmit history data while the buffer is full, the buffer overflows and the new data is discarded.

chanhvinh.com

21.8 Gateway Function

When a transmit/receive FIFO buffer is set to gateway mode, receive massages can be transmitted from an arbitrary channel without CPU intervention.

When a transmit/receive FIFO buffer for which the CFM[1:0] bits in the RSCAN0CFCCk register are set to 10_B (gateway mode) is selected by the RSCAN0GAFLP1j register, messages that passed through the filter processing of the receive rule are stored in the specified transmit/receive FIFO buffer and are automatically transmitted from the buffer.

Messages stored in a transmit/receive FIFO buffer are transmitted sequentially on a first-in, first-out basis. Only the message to be transmitted next becomes the target of transmit priority determination.

Transmit/receive FIFO buffers in the gateway mode are disabled by setting the CFE bit in the RSCAN0CFCCk register to 0 and the CFEMP flag becomes 1 according to the timing below.

- The transmit/receive FIFO buffer becomes empty immediately when the oldest message in it is not being transmitted and will not to be transmitted next.
- The transmit/receive FIFO buffer becomes empty after transmission completion, CAN bus error detection, or arbitration-lost when the message in it is being transmitted or will be transmitted next.

When the CFE bit is cleared to 0, all messages in transmit/receive FIFO buffers are lost and messages can no longer be stored in transmit/receive FIFO buffers. Confirm that the CFEMP flag is set to 1 before setting the CFE bit to 1 again.

chanhvinh.com

21.9 Test Function

The test function is classified into communication tests and global tests.

- Communication tests: Performed for each channel.
  - Standard test mode
  - Listen-only mode
  - Self-test mode 0 (external loopback mode)
  - Self-test mode 1 (internal loopback mode)
- Global tests: Performed for the entire module
  - Inter-channel communication test

21.9.1 Standard Test Mode

Standard test mode allows CRC test.

21.9.2 Listen-Only Mode

Listen-only mode allows reception of data frames and remote frames. Only recessive bits are transmitted on the CAN bus, and the ACK bit, overload flag, and active error flag are not transmitted.

Listen-only mode is available for detecting the communication speed.

Do not make a transmit request from any buffer or queue in listen-only mode.

Figure 21.12 shows the connection when listen-only mode is selected.

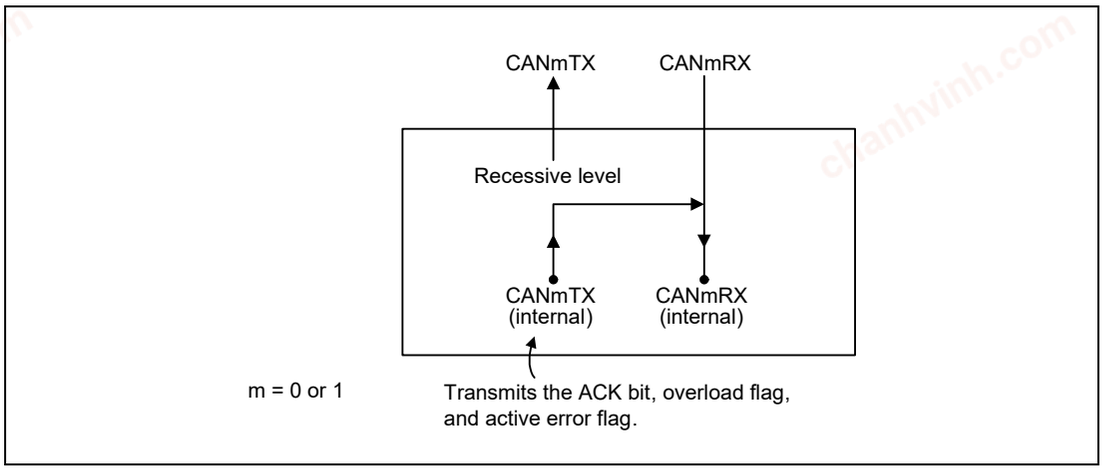


Figure 21.12 Connection when Listen-Only Mode is Selected

chanhvinh.com

21.9.3 Self-Test Mode (Loopback Mode)

In self-test mode, transmitted messages are compared with the receive rule of the own channel and the messages are stored in a buffer if they have passed through the filter processing. Messages transmitted from other CAN nodes are compared only with the receive rule for which the GAFLLB bit in the RSCAN0GAFLIDj register (j = 0 to 15) is set to 0 (when a message transmitted from another CAN node is received).

If the mirror function and self-test mode are both enabled, the self-test mode setting takes precedence.

21.9.3.1 Self-Test Mode 0 (External Loopback Mode)

Self-test mode 0 is used to perform a loopback test within a channel including the CAN transceiver.

In self-test mode 0, transmitted messages are handled as messages received through the CAN transceiver and are stored in a buffer. An ACK bit is generated to receive messages transmitted from the own CAN node.

Figure 21.13 shows the connection when self-test mode 0 is selected.

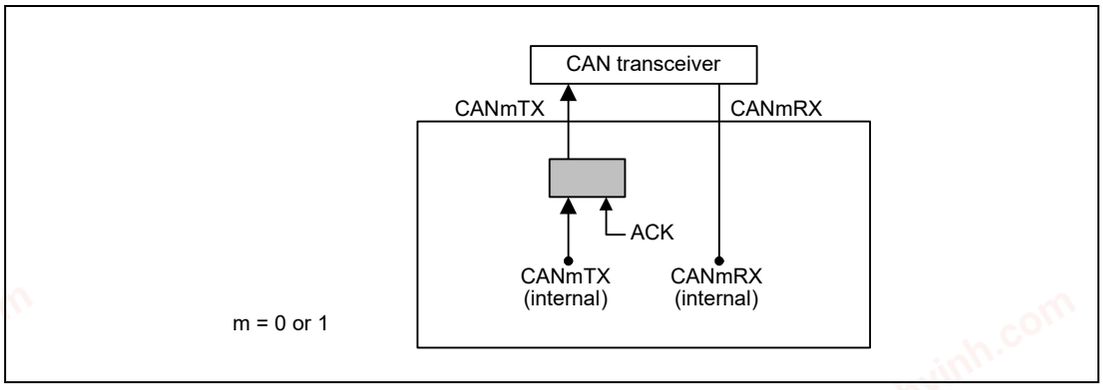


Figure 21.13 Connection when Self-Test Mode 0 is Selected

chanhvinh.com

21.9.3.2 Self-Test Mode 1 (Internal Loopback Mode)

In self-test mode 1, transmitted messages are handled as received messages and are stored in a buffer. An ACK bit is generated to receive messages transmitted from the own CAN node.

In self-test mode 1, internal feedback from the internal CANmTX pin (m = 0 or 1) to the internal CANmRX pin is performed. The external CANmRX pin input is isolated. The external CANmTX pin outputs only recessive bits.

Figure 21.14 shows the connection when self-test mode 1 is selected.

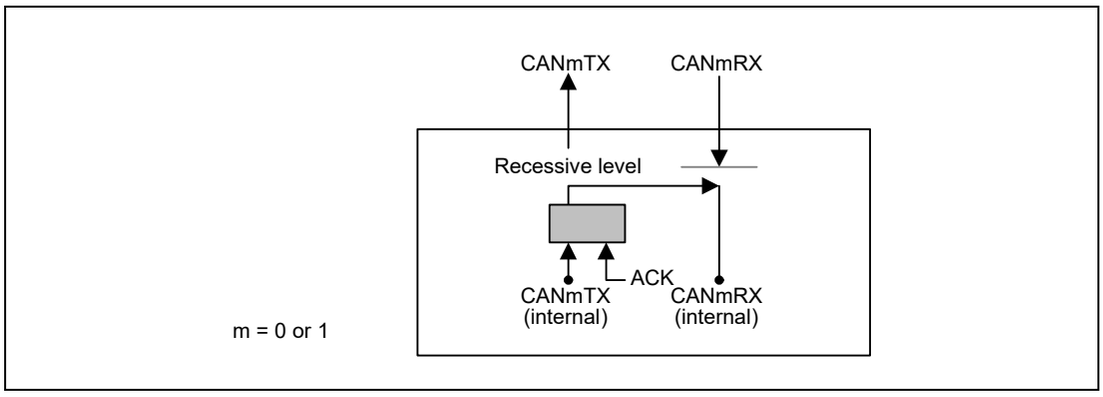


Figure 21.14 Connection when Self-Test Mode 1 is Selected

chanhvinh.com

21.9.4 Inter-Channel Communication Test

The inter-channel communication test function allows communication test by internally connecting CAN channels to each other. During this test, channels are isolated from the external CAN bus.

Before starting data transmission/reception in channel communication mode, make transmission/reception settings for each channel.

Figure 21.15 shows the connection for inter-channel communication test.

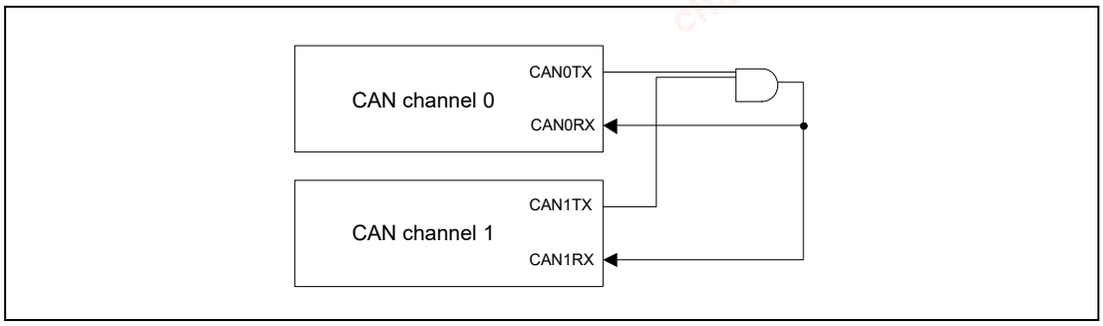


Figure 21.15 Connection for Inter-Channel Communication Test

chanhvinh.com

21.10 RS-CAN Setting Procedure

21.10.1 Initial Settings

The RS-CAN module initializes the CAN RAM after this LSI is reset. The RAM initialization time is 6082 cycles of pclk. The GRAMINIT flag in the RSCAN0GSTS register is set to 1 (CAN RAM initialization is ongoing) during the RAM initialization and is cleared to 0 (CAN RAM initialization is finished) when the initialization is completed. Make CAN settings after the GRAMINIT flag is cleared to 0. Figure 21.16 shows the CAN setting procedure after this LSI is reset.

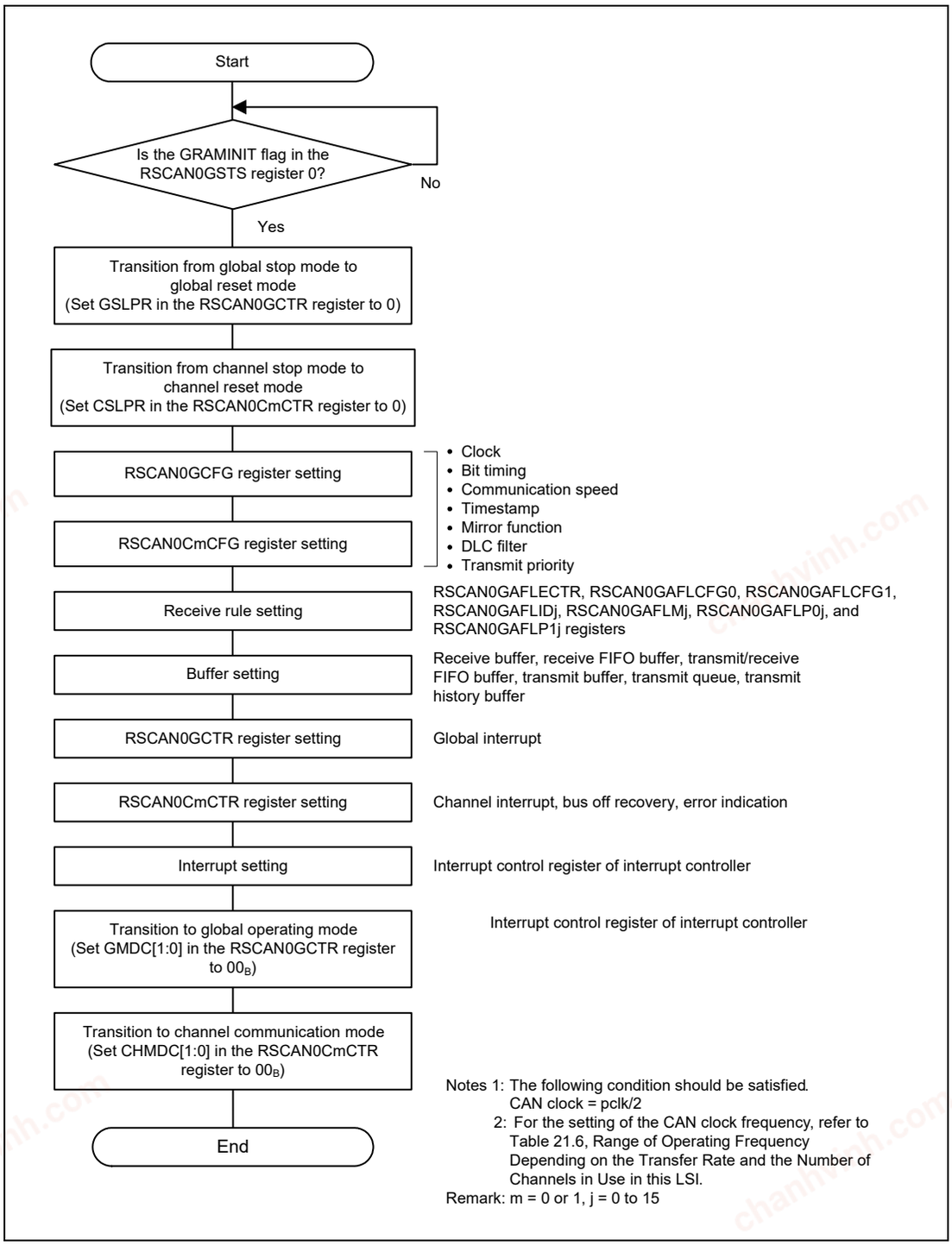


Figure 21.16 CAN Setting Procedure after the this LSI is Reset

chanhvinh.com

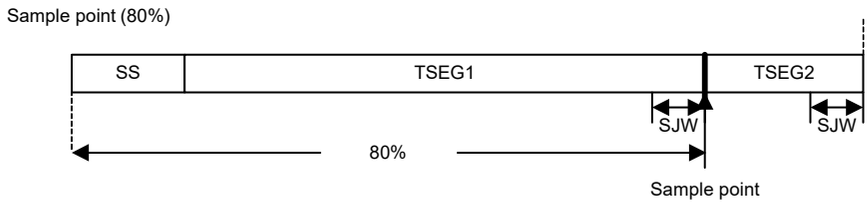
21.10.1.1 Clock Setting

Set the CAN clock (fCAN) as a clock source of the RS-CAN module. Select the clk_xincan or clkc using the DCS bit in the RSCAN0GCFG register.

21.10.1.2 Bit Timing Setting

In the CAN protocol, one bit of a communication frame consists of three segments SS, TSEG1, and TSEG2. Two of the segments, TSEG1 and TSEG2, can be set by the RSCAN0CmCFG register for each channel. Sample point timing can be determined by setting these two segments. This timing can be adjusted in units of 1 Time Quantum (referred to as Tq hereinafter). 1 Tq is equal to one CANmTq clock cycle. The CANmTq clock is obtained by selecting the clock source with the DCS bit in the RSCAN0GCFG register and selecting the clock division ratio with the BRP[9:0] bits in the RSCAN0CmCFG register.

Figure 21.17 shows the bit timing chart. Table 21.91 shows an example of bit timing setting.



SS = 1 Tq fixed
Set TSEG1 to a range of 4 Tq to 16 Tq
Set TSEG2 to a range of 2 Tq to 8 Tq
Set SJW to a range of 1 Tq to 4 Tq
Set SS + TSEG1 + TSEG2 to a range of 8 Tq to 25 Tq
TSEG1 > TSEG2 ≥ SJW

- SS (synchronization segment):
  The SS is a segment that performs synchronization by monitoring the edge from recessive to dominant bits in the Interframe Space.
  Interframe Space consists of Intermission, Suspend Transmission, and Bus Idle. All nodes can start transmission during Bus Idle.
- TSEG1 (Time segment 1):
  TSEG1 is a segment that absorbs physical delay on the CAN network. The physical delay on the CAN network is twice the total of the delay on the CAN bus, the delay in the input comparator, and the delay in the output driver.
- TSEG2 (Time segment 2):
  TSEG2 is a segment that compensates phase error due to an error in frequency.
- SJW (Resynchronization jump width)
  The SJW is a length to extend or reduce the time segment to compensate for an error in phase due to phase error.

Figure 21.17 Bit Timing Chart

Table 21.91 Example of Bit Timing Setting

1 Bit	Set Value (Tq)				Sample Point (%) Note: See Figure 21.17.
	SS	TSEG1	TSEG2	SJW	
8Tq	1	4	3	1	62.50
	1	5	2	1	75.00
12Tq	1	8	3	1	75.00
	1	9	2	1	83.33
16Tq	1	10	5	1	68.75
	1	11	4	1	75.00
24Tq	1	14	9	1	62.50
	1	15	8	1	66.66

chanhvinh.com

21.10.1.3 Communication Speed Setting

Set the CAN communication speed for each channel using the fCAN, baud rate prescaler division value (BRP[9:0] bits in the RSCAN0CmCFG register), and Tq count per bit time.

Figure 21.18 shows the CAN clock control block diagram, and Table 21.92 shows an example of the communication speed setting.

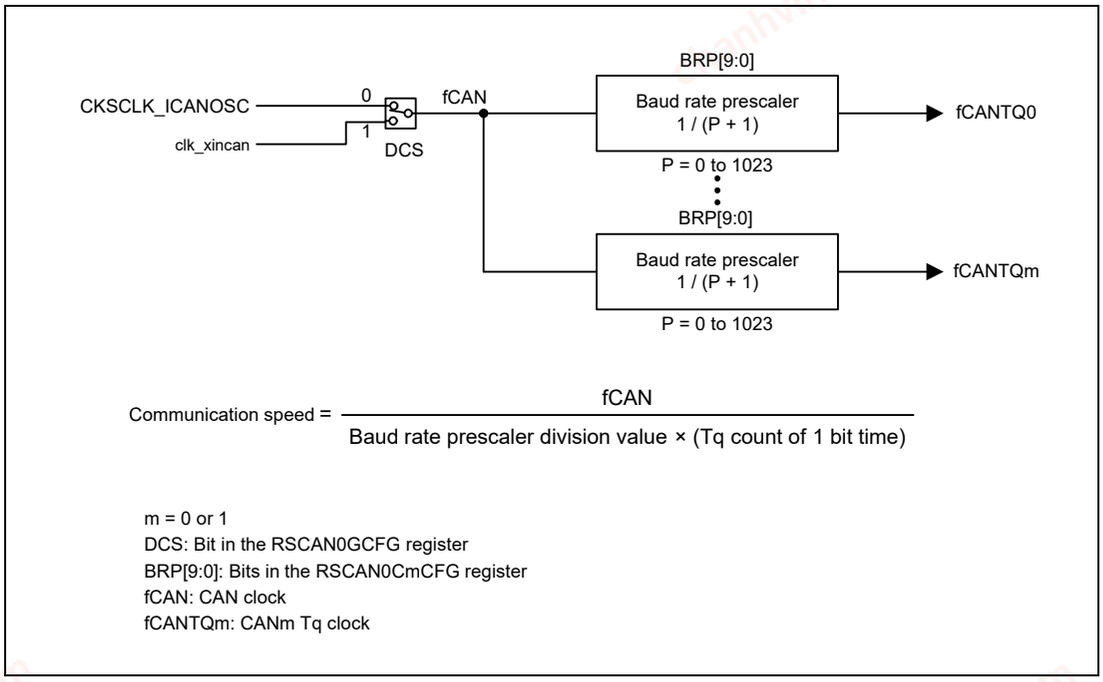


Figure 21.18 CAN Clock Control Block Diagram

Table 21.92 Example of Communication Speed Setting

fCAN / Communication speed	32MHz	24MHz	8MHz
1Mbps	8Tq (4) 16Tq (2)	8Tq (3) 12Tq (2) 24Tq (1)	8Tq (1)
500Kbps	8Tq (8) 16Tq (4)	8Tq (6) 12Tq (4) 24Tq (2)	8Tq (2) 16Tq (1)
125Kbps	8Tq (32) 16Tq (16)	8Tq (24) 12Tq (16) 24Tq (8)	8Tq (8) 16Tq (4)

Note: Values in ( ) are baud rate prescaler division values.

chanhvinh.com

21.10.1.4 Receive Rule Setting

Receive rules can be set using receive rule-related registers.

Up to 16 receive rules can be registered per page. Specify pages 0 to 7 by the AFLPN[4:0] bits in the RSCAN0GAFLECTR register.

Set receive rule table write enable/disable using the AFLDAE bit.

Figure 21.19 shows the receive rule setting procedure.

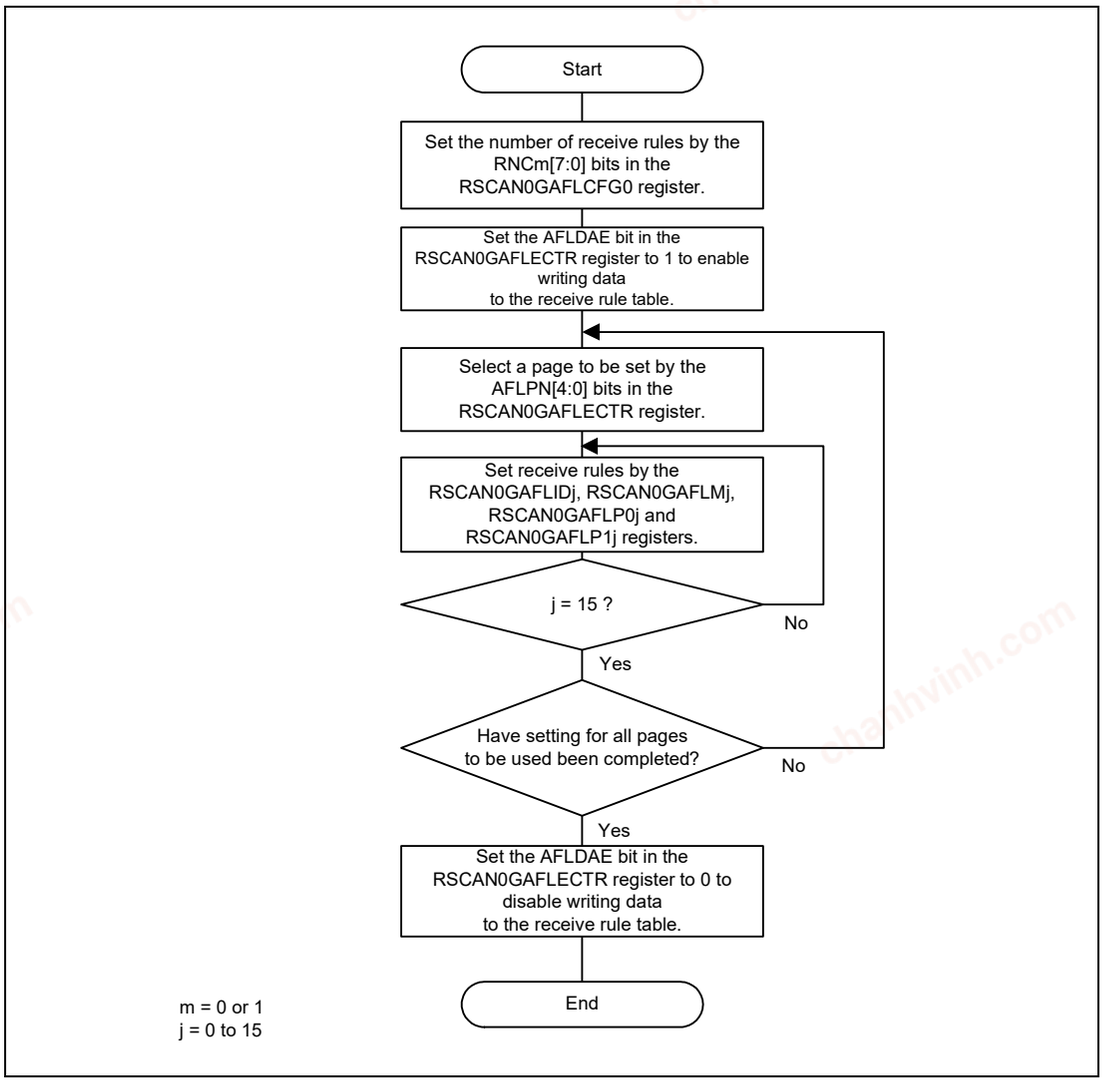


Figure 21.19 Receive Rule Setting Procedure

chanhvinh.com

21.10.1.5 Buffer Setting

Set sizes and interrupt sources of buffers. For transmit/receive FIFO buffers that are set to transmit mode, set transmit buffers to be linked.

Figure 21.20 shows the buffer configuration. Figure 21.21 shows the buffer setting procedure.

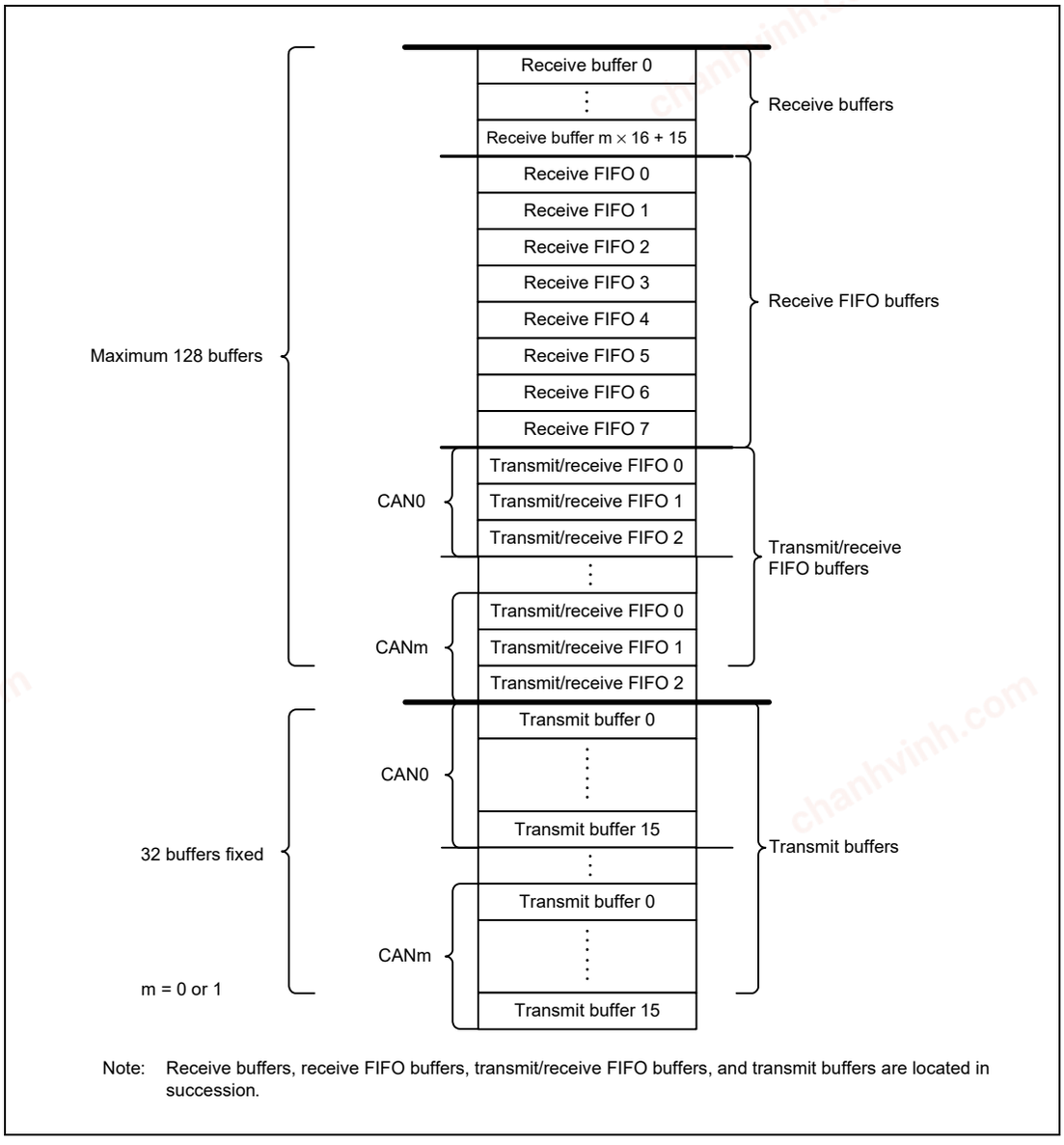


Figure 21.20 Buffer Configuration

chanhvinh.com

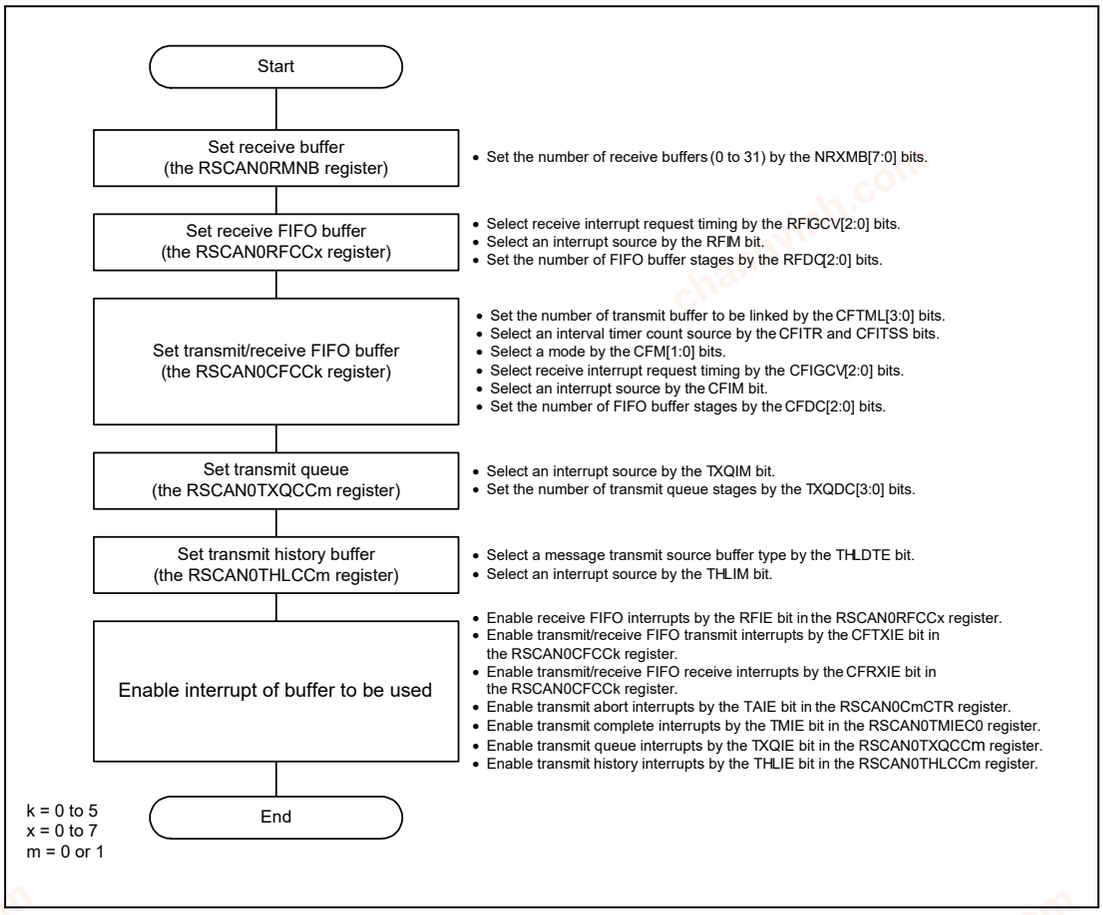


Figure 21.21 Buffer Setting Procedure

chanhvinh.com

21.10.2 Reception Procedure

21.10.2.1 Receive Buffer Reading Procedure

When the processing to store received messages in a receive buffer starts, the RMNSq flag in the RSCAN0RMNDy register (y = 0, q = 0 to 31) is set to 1 (receive buffer q contains a new message). Messages can be read from the RSCAN0RMIDq, RSCAN0RMPTRq, RSCAN0RMDF0q, and RSCAN0RMDF1q registers. If the next message has been received before the current message is read from the receive buffer, the message is overwritten. Figure 21.22 shows the receive buffer reading procedure.

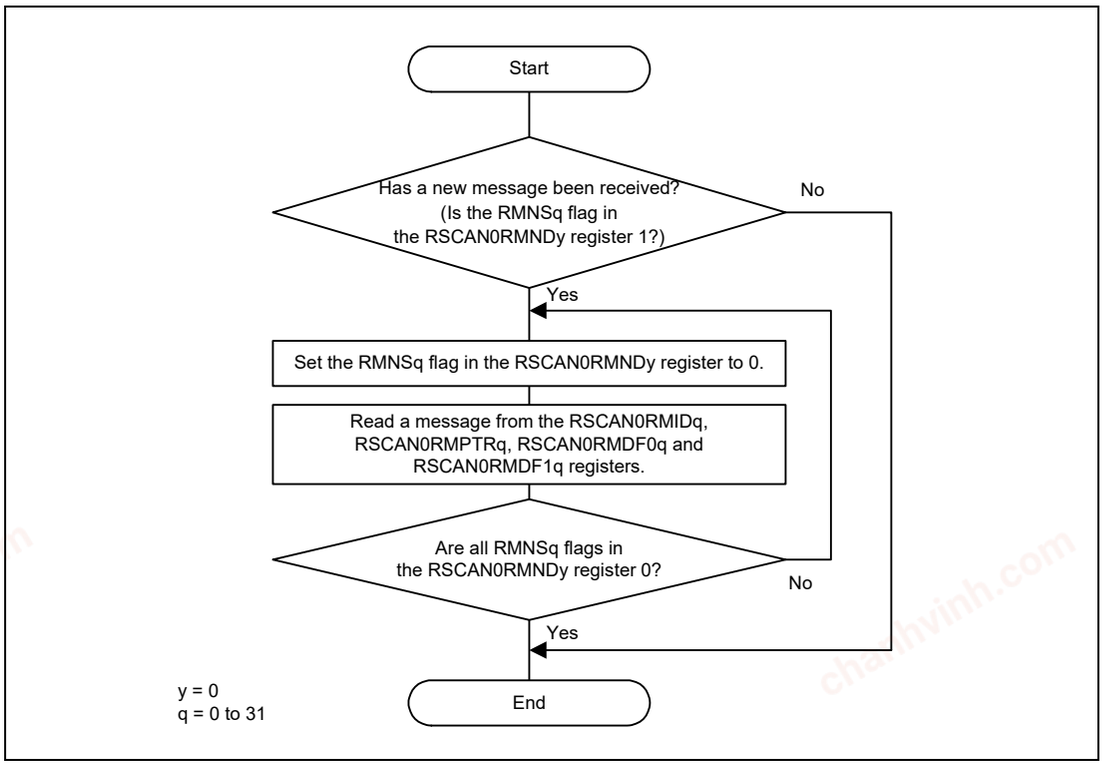


Figure 21.22 Receive Buffer Reading Procedure

chanhvinh.com

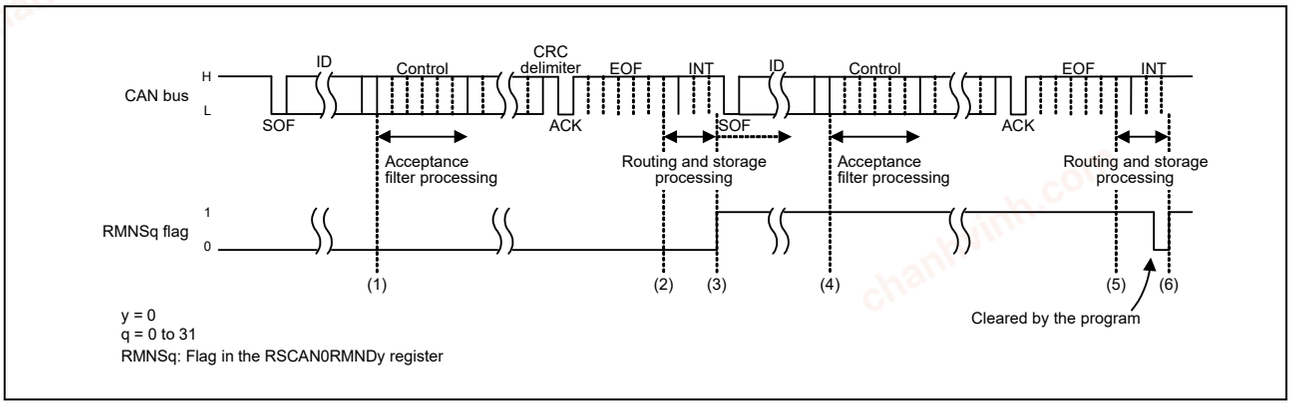


Figure 21.23 Receive Buffer Reception Timing Chart

(1) When the ID field in a message has been received, the acceptance filter processing starts.

(2) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCAN0GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(3) When the message has passed through the DLC filter processing, the processing to store the message in the specified receive buffer starts.
When the message storage processing starts, the RMNSq flag in the corresponding the RSCAN0RMNDy register is set to 1 (receive buffer n contains a new message). If other channels are performing filter processing or transmit priority determination processing, the routing processing and the storage processing may be delayed.

(4) When the ID field of the next message has been received, the acceptance filter processing starts.

(5) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCAN0GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(6) When the corresponding RMNSq flag is cleared to 0 (receive buffer n contains no new message), this flag is set to 1 again when the message storage processing starts. Even if the RMNSq flag remains 1, a new message is overwritten to the receive buffer. The RMNSq flag should not be cleared to 0 during storage of messages.

chanhvinh.com

21.10.2.2 FIFO Buffer Reading Procedure

When received messages have been stored in one or more receive FIFO buffers or a transmit/receive FIFO buffer that is set to receive mode or gateway mode, the corresponding message count display counter (RFMC[7:0] bits in the RSCAN0RFSTSx register (x = 0 to 7) or CFMC[7:0] bits in the RSCAN0CFSTSk register (k = 0 to 5)) is incremented. At this time, when the RFIE bit (receive FIFO interrupt is enabled) in the RSCAN0RFCCx register or the CFRXIE bit (transmit/receive FIFO receive interrupt is enabled) in the RSCAN0CFCCk register is set to 1, an interrupt request is generated. Received messages can be read from the RSCAN0RFIDx, RSCAN0RFPTRx, RSCAN0RFDF0x, and RSCAN0RFDF1x registers for receive FIFO buffers, or from the RSCAN0CFIDk, RSCAN0CFPTRk, RSCAN0CFDF0k, and RSCAN0CFDF1k registers for transmit/receive FIFO buffers. Messages in FIFO buffers can be read sequentially on a first-in, first-out basis.

When the message count display counter value matches the FIFO buffer depth (a value set by the RFDC[2:0] bits in the RSCAN0RFCCx register or the CFDC[2:0] bits in the RSCAN0CFCCk register), the RFFLL or CFFLL flag is set to 1 (the receive FIFO buffer is full).

When all messages have been read out of the FIFO buffer, the RFEMP flag in the RSCAN0RFSTSx register or the CFEMP flag in the RSCAN0CFSTSk register is set to 1 (the receive FIFO buffer contains no unread message (buffer empty)).

If the RFE bit or the CFE bit is cleared to 0 (no receive FIFO buffer is used) with the interrupt request flag (RFIF flag in the RSCAN0RFSTSx register or CFRXIF flag in the RSCAN0CFSTSk register) set to 1 (a receive FIFO interrupt request is present), the interrupt request flag is not automatically cleared to 0. The program must clear the interrupt request flag to 0.

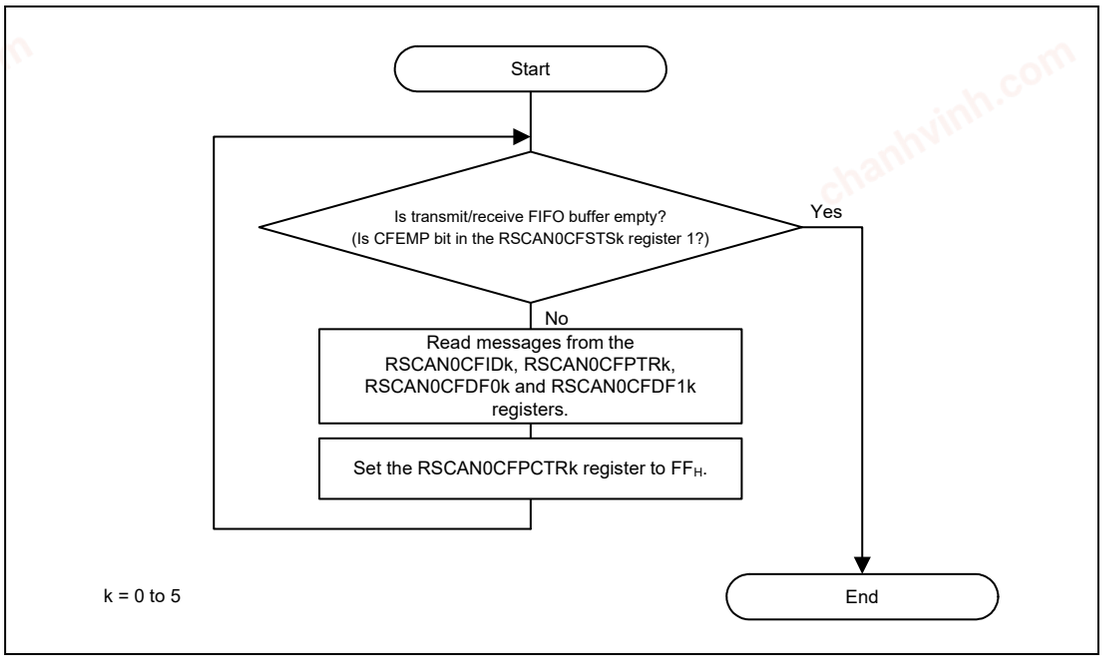


Figure 21.24 Transmit/Receive FIFO Buffer Reading Procedure

chanhvinh.com

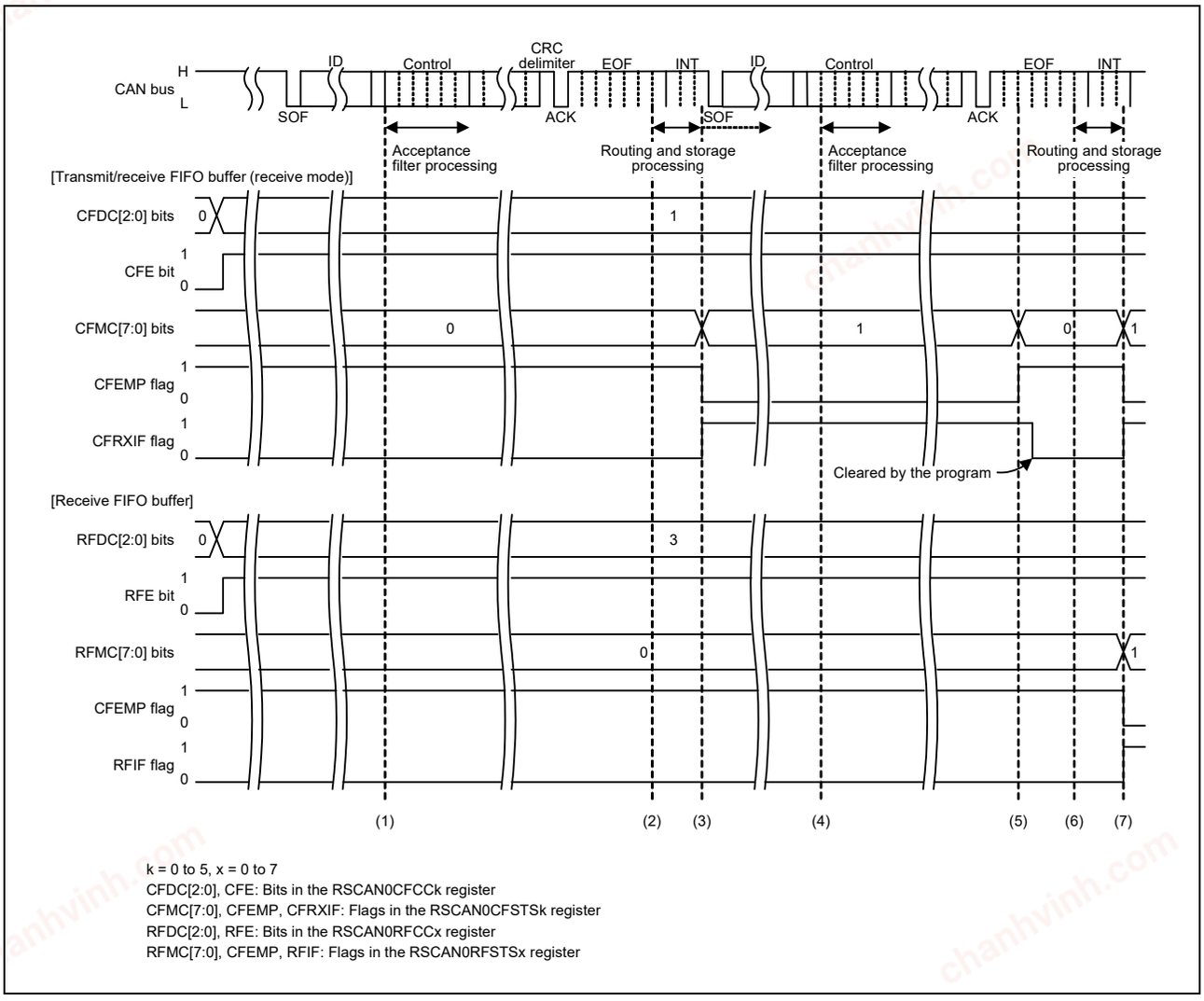


Figure 21.25 FIFO Buffer Reception Timing Chart

(1) When the ID field in a message has been received, the acceptance filter processing starts.

(2) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCAN0GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(3) When the message has passed through the DLC filter processing and the CFE value in the RSCAN0CFCCk register is 1 (transmit/receive FIFO buffers are used) and the CFDC[2:0] value in the RSCAN0CFCCk register is 001_B or more, the message is stored in the transmit/receive FIFO buffer that is set to receive mode. The CFMC[7:0] value in the RSCAN0CFSTSk register is incremented and becomes 01_H. When the CFIM bit in the RSCAN0CFCCk register is set to 1 (a FIFO receive interrupt request is generated each time a message has been received), the CFRXIF flag in the RSCAN0CFSTSk register is set to 1 (a transmit/receive FIFO receive interrupt request is present). The CFRXIF flag can be reset to 0 by the program.

(4) When the ID field of the next message has been received, the acceptance filter processing starts.

chanhvinh.com

(5) Read received messages from the RSCAN0CFIDk, RSCAN0CFPTRk, RSCAN0CFDF0k, and RSCAN0CFDF1k registers and write FF_H to the RSCAN0CFPCTRk register. This causes the CFMC[7:0] bits in the RSCAN0CFSTSk register to be decremented. When CFMC[7:0] becomes 00_H, the CFEMP flag in the RSCAN0CFSTSk register becomes 1 (the transmit/receive FIFO buffer contains no message (buffer empty)).

(6) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCAN0GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(7) The message is stored in the transmit/receive FIFO buffer set in receive mode when the message has passed through the DLC filter process if the CFE bit is set to 1 (transmit/receive FIFO buffers are used), the RFE bit in the RSCAN0RFCCx register is set to 1, and the CFDC[2:0] bits are set to 001_B or more. The CFMC[7:0] bit value is incremented by 1 to be 01_H. When the CFIM bit is set to 1 (an interrupt occurs each time a message has been received), the CFRXIF flag is set to 1 (a transmit/receive FIFO receive interrupt request is present).
The message is stored in the receive FIFO buffer if the RFE bit in the RSCAN0RFCCx register is set to 1 (receive FIFO buffers are used), and the RFDC[2:0] bits in the RSCAN0RFCCx register are set to 001_B or more. The RFMC[7:0] bits in the RSCAN0RFSTSx register are set to 01_H by being incremented by 1. When the RFIM bit in the RSCAN0RFCCx register is set to 1 (an interrupt occurs each time a message has been received), the RFIF flag in the RSCAN0RFSTSx register is set to 1 (a receive FIFO interrupt request is present).

chanhvinh.com

21.10.3 Transmission Procedure

21.10.3.1 Procedure for Transmission from Transmit Buffers

Figure 21.26 shows the procedure for transmission from transmit buffers.

Figure 21.27 shows a timing chart where messages are transmitted from two transmit buffers in the same channel and transmission has been successfully completed. Figure 21.28 shows a timing chart where messages are transmitted from two transmit buffers in the same channel and transmit abort has been completed.

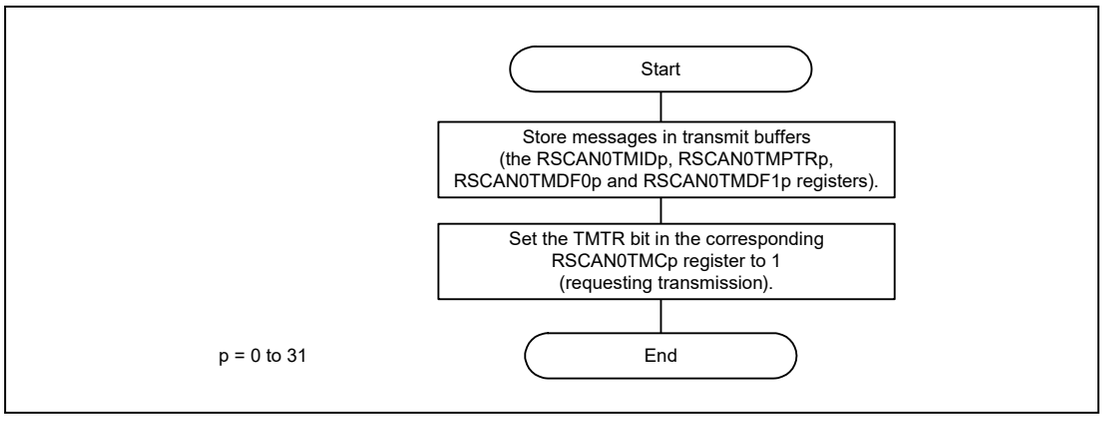


Figure 21.26 Procedure for Transmission from Transmit Buffers

chanhvinh.com

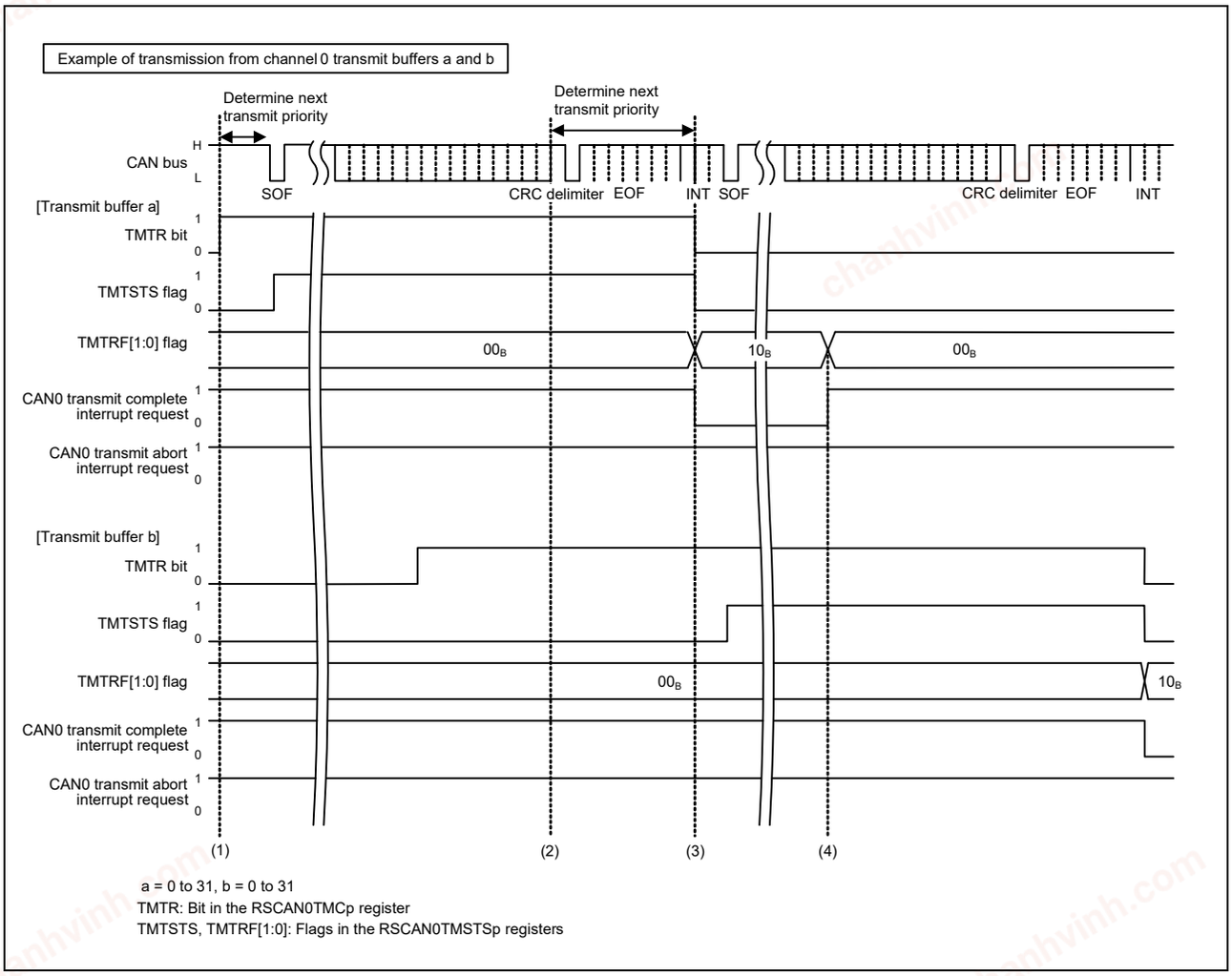


Figure 21.27 Transmit Buffer Transmission Timing Chart (Transmission Completed Successfully)

(1) When the TMTR bit in the RSCAN0TMCa register is set to 1 while the CAN bus is idle, the transmit priority determination processing starts to determine the highest-priority transmit buffer. If transmit buffer a is determined to be the highest-priority transmit buffer, the TMTSTS flag in the corresponding the RSCAN0TMCa register is set to 1 (transmission is in progress) and the CAN channel starts transmitting data.

(2) When a transmit request from a buffer is present, the priority determination starts with the CRC delimiter for the next transmission.

(3) When transmission completes successfully, the TMTRF[1:0] flag in the RSCAN0TMCa register is set to 10_B (transmission has been completed (without transmit abort request)) and the TMTSTS flag and the TMTR bit in the RSCAN0TMCa register are cleared to 0. When the TMIEa value in the RSCAN0TMIEC0 register is 1 (transmit buffer interrupt is enabled), a CAN0 transmit complete interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flag to 00_B (transmission is in progress or no transmit request is present).

(4) Before starting the next transmission, set the TMTRF[1:0] flag to 00_B. Write the next message to the transmit buffer, and then set the TMTR bit to 1 (transmission is requested). The TMTR bit can be set to 1 only when the TMTRF[1:0] flag value is 00_B.

chanhvinh.com

If an arbitration-lost has occurred after transmission is started, the TMTSTS flag is cleared to 0. The transmit priority determination is reexecuted at the beginning of the CRC delimiter to search the highest-priority transmit buffer. If an error has occurred during transmission or after arbitration loss, the priority determination processing is reexecuted during transmission of an error frame.

The timing the TMTSTS flag is set to 1 is not always the start timing of SOF. The start timing of SOF can be delayed up to the start timing of the basic ID due to the synchronization logic implemented for PLL bypass.

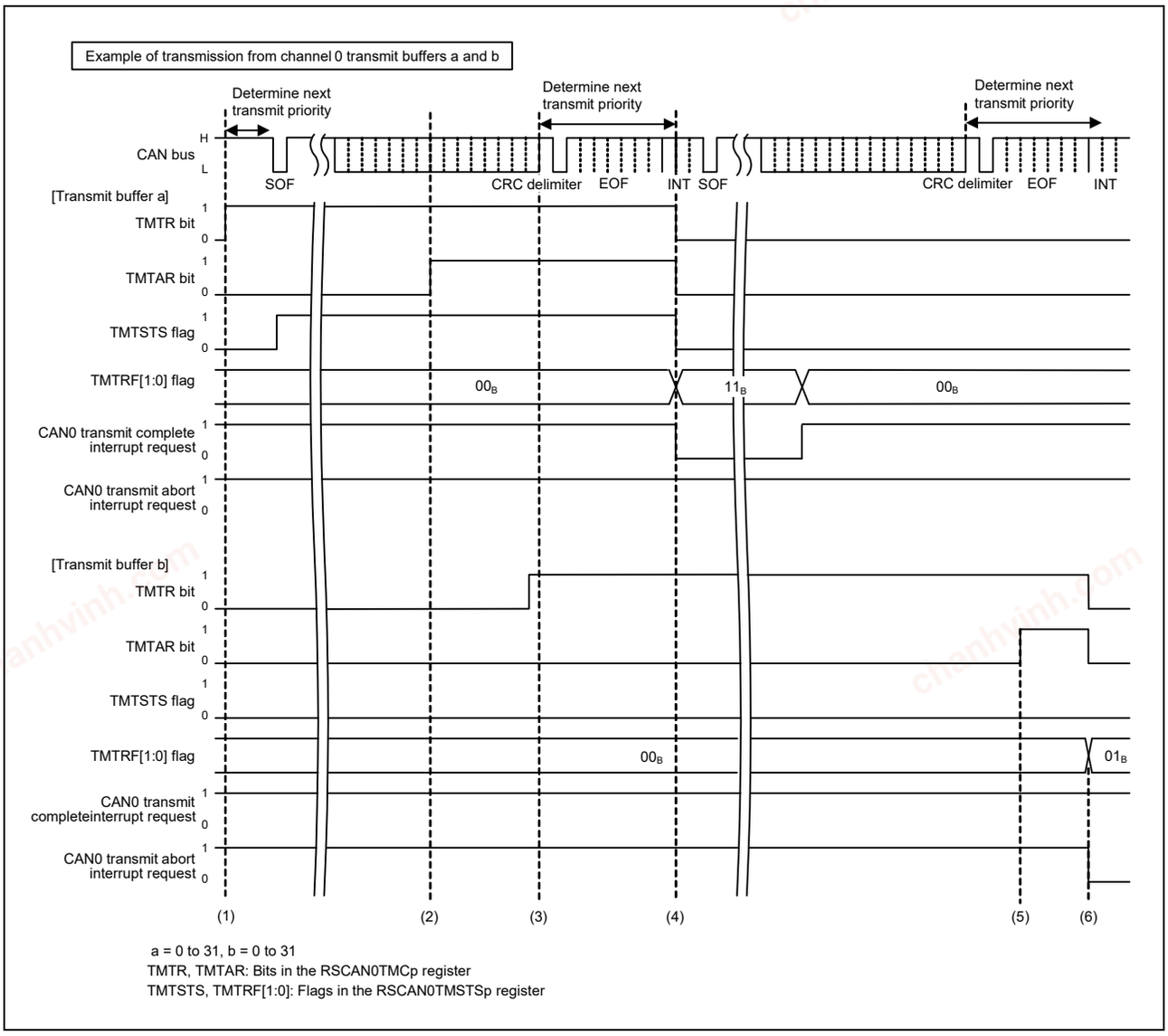


Figure 21.28 Transmit Buffer Transmission Timing Chart (Transmit Abort Completed)

(1) When the TMTR bit in the RSCAN0TMCa register is set to 1 while the CAN bus is idle, the transmit priority determination processing starts to determine the highest-priority transmit buffer. If transmit buffer a is determined to be the highest-priority transmit buffer, the TMTSTS flag in the corresponding RSCAN0TMCa register is set to 1 (transmission is in progress) and the CAN channel starts transmitting data.

(2) When it is determined that the transmit buffer is used for the next transmission or transmission is in progress, message transmission is not aborted unless an error or arbitration loss occurs even if the TMTAR bit is set to 1 (transmit abort is requested).

chanhvinh.com

(3) The priority determination starts with the CRC delimiter for the next transmission. In this timing chart, buffer b is not selected as the next transmit buffer.

(4) When transmission completes successfully, the TMTRF[1:0] flag in the RSCAN0TMCa register is set to 11_B (transmission has been completed (with transmit abort request)) and the TMTSTS flag and the TMTR bit in the RSCAN0TMCa register are cleared to 0. When the TMIEa value in the RSCAN0TMIEC0 register is 1 (transmit buffer interrupt is enabled), a CAN0 transmit complete interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flag to 00_B (transmission is in progress or no transmit request is present).

(5) While another CAN node is transmitting data on the CAN bus (TMTSTS flag = 0), if the TMTAR bit is set to 1 while the corresponding channel is determining transmit priority, the TMTR bit cannot be cleared to 0.

(6) After the internal processing time has passed, the transmission is terminated and the TMTRF[1:0] flag is set to 01_B. When the transmit buffer is not transmitting data and is not selected as the next transmit buffer and priority determination is not being made, an abort request is immediately accepted and the TMTRF[1:0] flag is set to 01_B. At this time, the TMTR and TMTAR bits are cleared to 0. When transmit abort is completed with the TAIE bit in the RSCAN0CmCTR register set to 1 (transmit abort interrupt is enabled), an interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flag to 00_B.

If an arbitration loss has occurred after the CAN channel started transmission, the TMTSTS bit is cleared to 0. The transmit priority determination is reexecuted at the beginning of the CRC delimiter to find the highest-priority transmit buffer. If an error has occurred during transmission or after arbitration loss, the priority determination processing is reexecuted during transmission of an error frame.

chanhvinh.com

21.10.3.2 Procedure for Transmission from Transmit/Receive FIFO Buffers

Figure 21.29 shows the procedure for transmission from transmit/receive FIFO buffers.

Figure 21.30 shows a timing chart where messages are transmitted from two transmit/receive FIFO buffers in the same channel and transmission has been successfully completed. Figure 21.31 shows a timing chart where messages are transmitted from two transmit/receive FIFO buffers in the same channel and transmit abort has been completed.

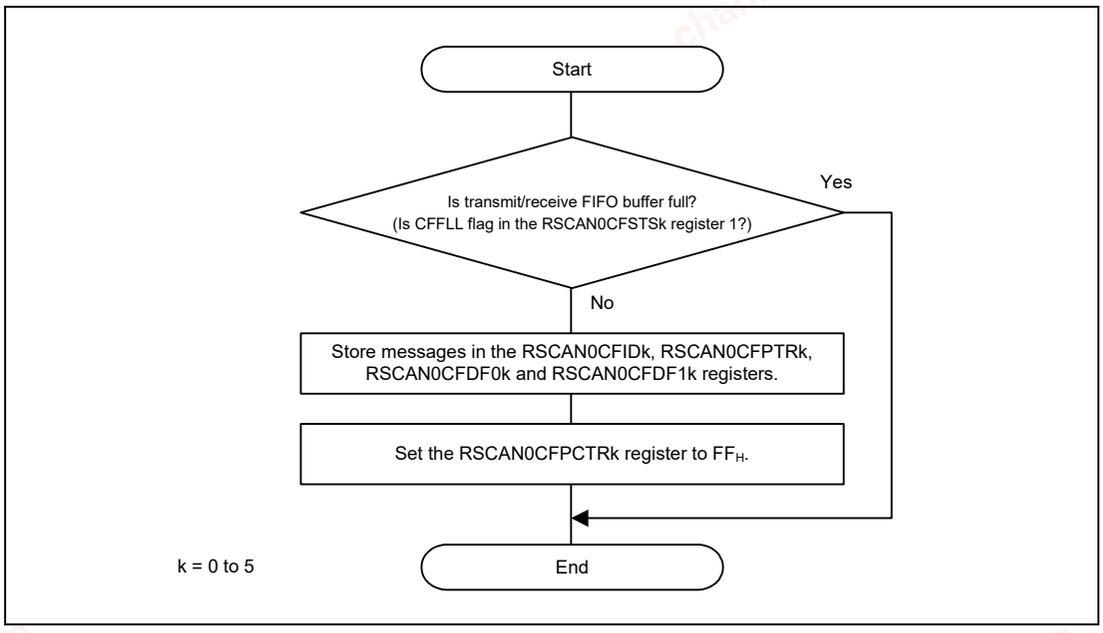


Figure 21.29 Procedure for Transmission from Transmit/Receive FIFO Buffers

chanhvinh.com

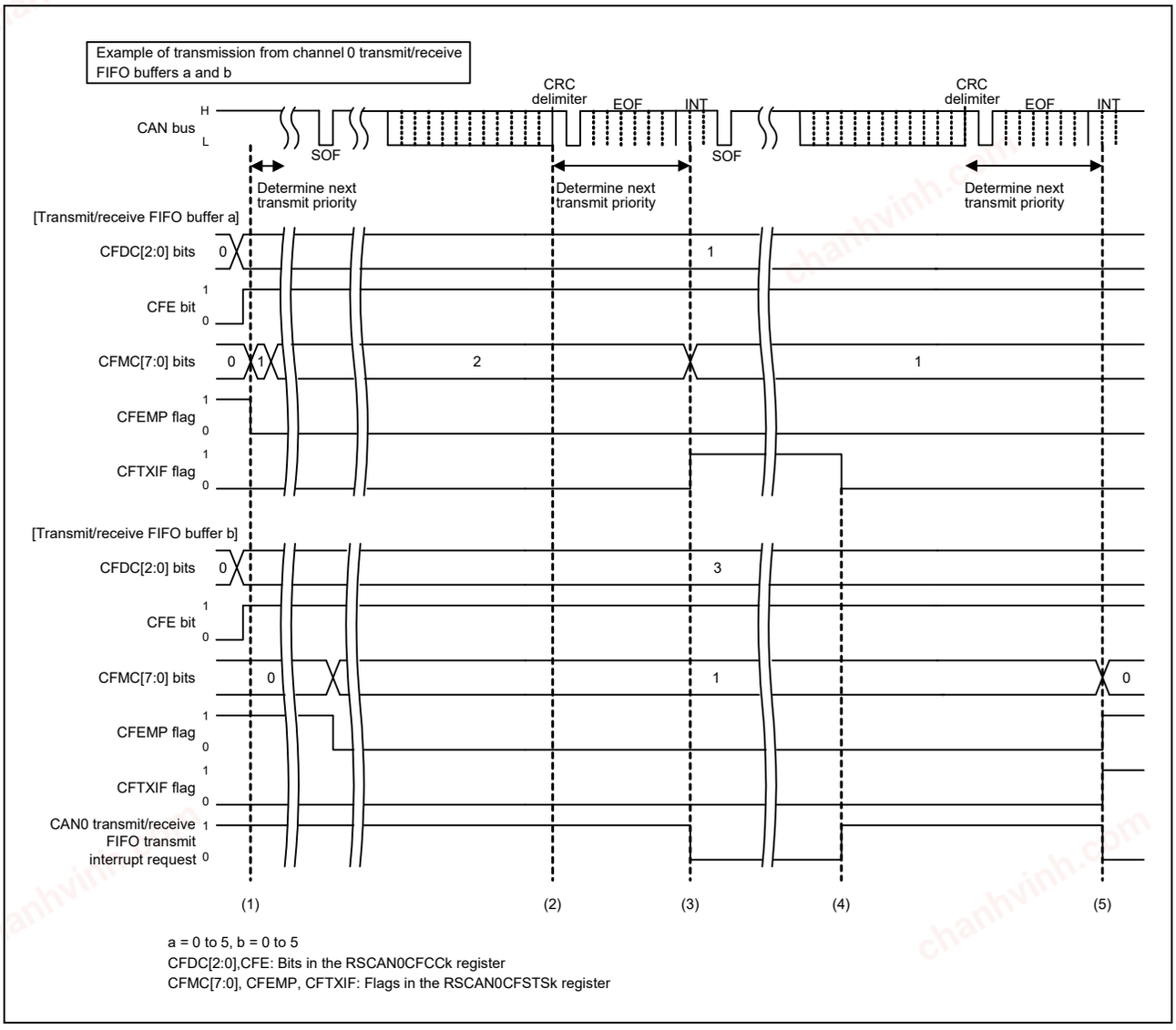


Figure 21.30 Transmit/Receive FIFO Buffer Transmission Timing Chart (Transmission Completed Successfully)

(1) While the CAN bus is idle, when the CFE value in the RSCA0CFCCa register is 1 (transmit/receive FIFO buffers are used) and the CFDC[2:0] value in the RSCAN0CFCCa register is 001_B (4 messages) or more and the CFMC[7:0] value in the RSCAN0CFSTSa register is 01_H or more, the priority determination processing starts to determine the highest-priority transmit message. When the highest-priority transmit message has been determined, transmission of the message starts. In this figure, the message is transmitted from transmit/receive FIFO buffer a of channel 0.

(2) When a transmit request from a buffer is present, the priority determination starts with the CRC delimiter for the next transmission.

(3) When transmission completes successfully, the CFMC[7:0] value in the RSCAN0CFSTSa register is decremented. Setting the CFIM bit in the RSCAN0CFCCa register to 1 (a FIFO transmit interrupt request is generated each time a message has been transmitted) sets the CFTXIF flag in the RSCAN0CFSTSk register to 1 (a transmit/receive FIFO transmit interrupt request is present).

(4) The program can clear the CFTXIF flag.

chanhvinh.com

(5) Message transmission from transmit/receive FIFO buffer b of channel 0 has been completed and the CFMC[7:0] value in the RSCAN0CFSTSb register is decremented. The CFMC[7:0] bits are cleared to 00_H and therefore the CFEMP flag in the RSCAN0CFSTSk register is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)).

Transmission is continued until the CFEMP flag is set to 1. It is possible to continuously store transmit messages in FIFO buffers until the CFFLL flag in the RSCAN0CFSTSa and RSCAN0CFSTSb register is set to 1 (the transmit/receive FIFO buffer is full).

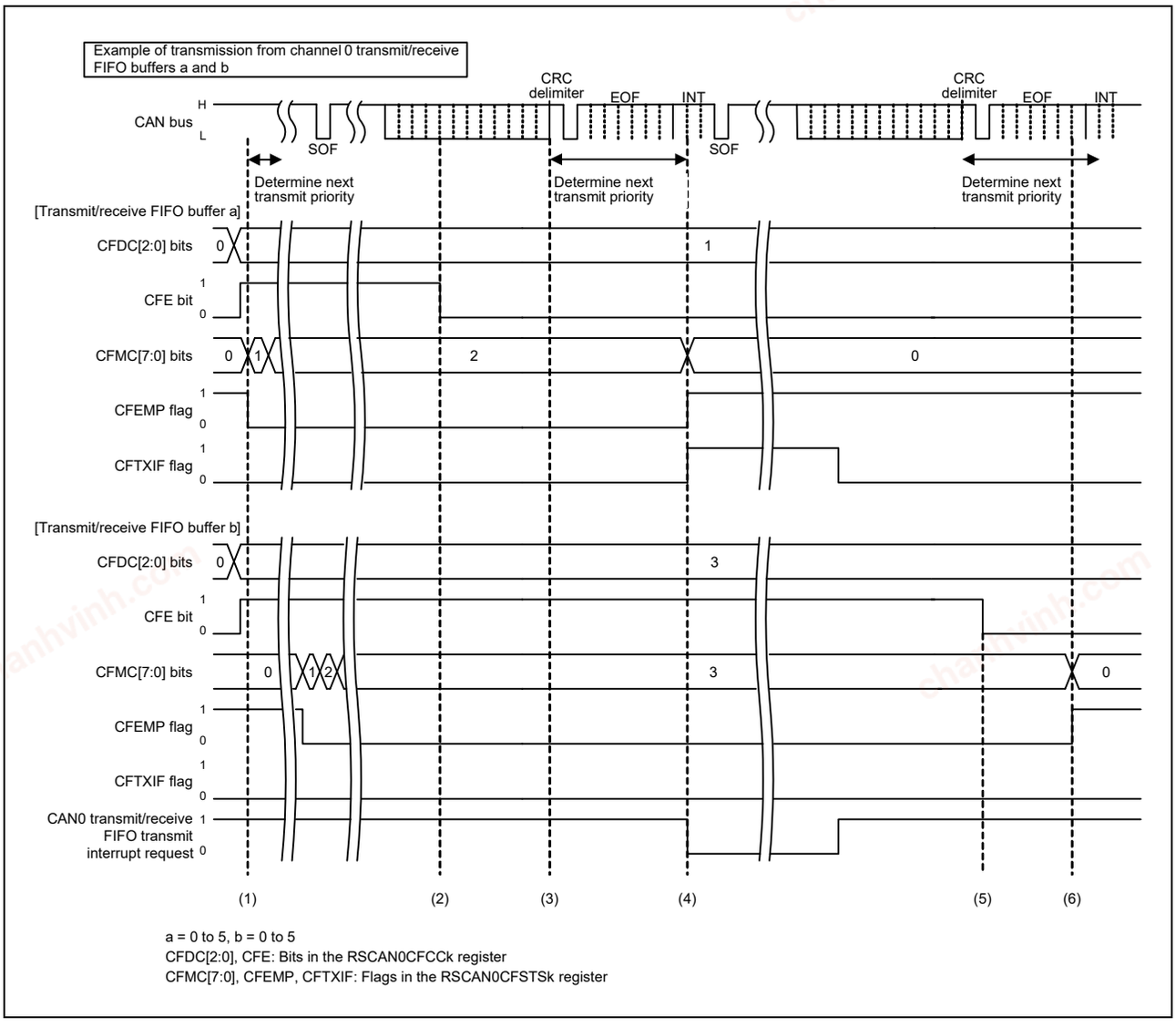


Figure 21.31 Transmit/Receive FIFO Buffer Transmission Timing Chart (Transmit Abort Completed)

(1) While the CAN bus is idle, when the CFE value in the RSCAN0CFCCa register (a = 0 to 5) is 1 (transmit/receive FIFO buffers are used) and the CFDC[2:0] value in the RSCAN0CFCCa register is 001_B (4 messages) or more and the CFMC[7:0] value in the RSCAN0CFSTSa register is 01_H or more, the priority determination processing starts to determine the highest-priority transmit message. When the highest-priority transmit message has been determined, transmission of the message starts. In this figure, the message is transmitted from transmit/receive FIFO buffer a of channel 0.

chanhvinh.com

(2) When transmission is in progress or it is determined that the transmit/receive FIFO buffer is used for the next transmission, message transmission is not aborted unless an error or arbitration loss occurs even if the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

(3) When a transmit request from a buffer is present, the priority determination starts with the CRC delimiter for the next transmission. In this figure, transmit/receive FIFO buffer b is not selected as a buffer for the next transmission.

(4) When transmit completes successfully, the CFMC[7:0] value is cleared to 00_H. Setting the CFIM bit to 1 (a FIFO transmit interrupt request is generated each time a message has been transmitted) sets the CFTXIF flag in the RSCAN0CFSTSa register to 1 (a transmit/receive FIFO transmit interrupt request is present). The program can clear the CFTXIF flag.

(5) If another CAN node on the CAN bus is transmitting data (not from transmit/receive FIFO buffer b), transmit/receive FIFO buffers cannot be disabled immediately even if the CFE bit in the RSCAN0CFCCb register is cleared to 0 (no transmit/receive FIFO buffer is used) during transmit priority determination. (The CFEMP flag in the RSCAN0CFSTSb register is not set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)) immediately.)

(6) After the internal processing time has passed, transmit/receive FIFO buffers are disabled and the CFMC[7:0] bits in the RSCAN0CFSTSb register are cleared to 00_H and the CFEMP flag is set to 1. When the transmit/receive FIFO buffer is not transmitting data and is not selected as the next transmit buffer and priority determination is not in progress, the transmit/receive FIFO buffer is immediately disabled. (The CFMC[7:0] bits are cleared to 00_H and the CFEMP flag is set to 1.)

21.10.3.3 Procedure for Transmission from the Transmit Queue

Figure 21.32 shows the procedure for transmission from the transmit queue.

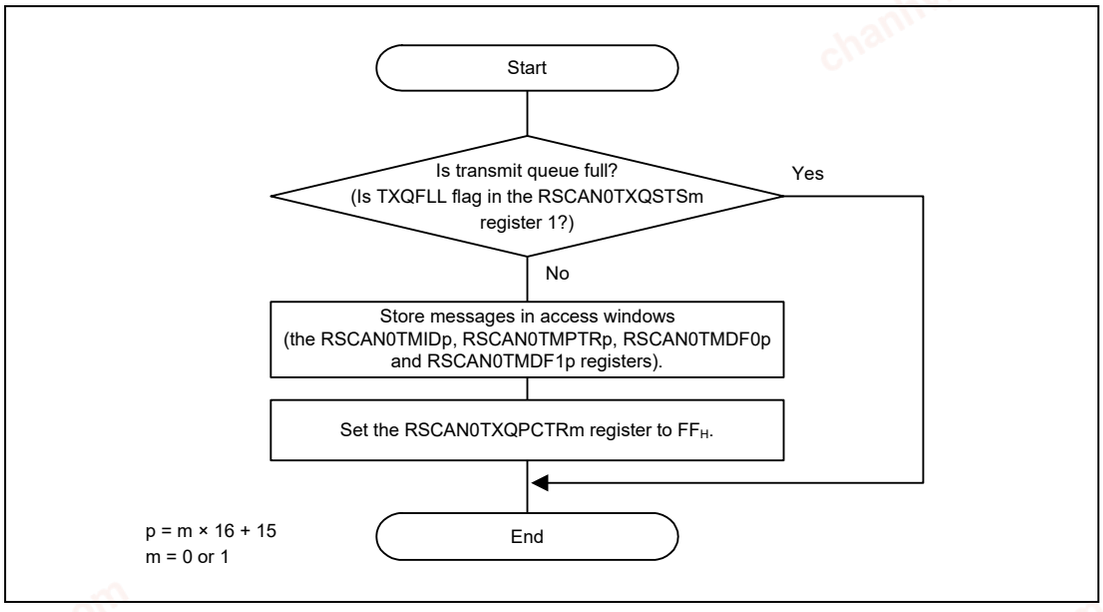


Figure 21.32 Procedure for Transmission from the Transmit Queue

chanhvinh.com

21.10.3.4 Transmit History Buffer Reading Procedure

Transmit history data can be read from the RSCAN0THLACCm register. The next data can be accessed by writing FF_H to the corresponding RSCAN0THLPCTRm register (m = 0 or 1) after reading a set of data. Figure 21.33 shows the transmit history buffer reading procedure.

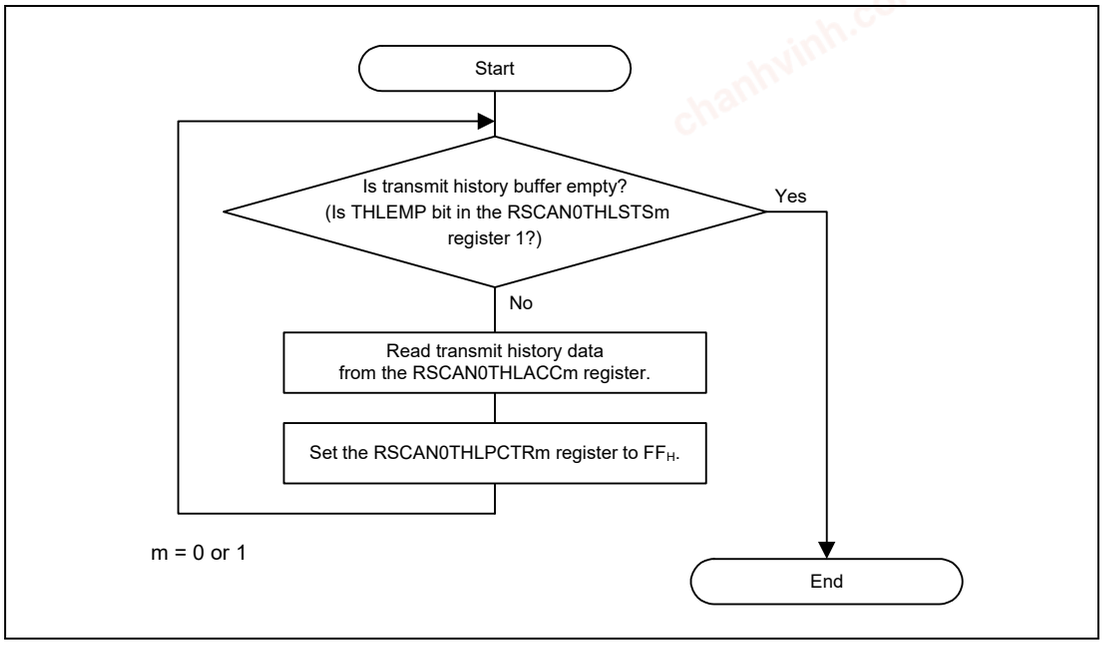


Figure 21.33 Transmit History Buffer Reading Procedure

chanhvinh.com

21.10.4 Test Settings

21.10.4.1 Self-Test Mode Setting Procedure

Self-test mode allows communication test on a channel basis by enabling a CAN node to receive its own transmitted messages.

Figure 21.34 shows the self-test mode setting procedure.

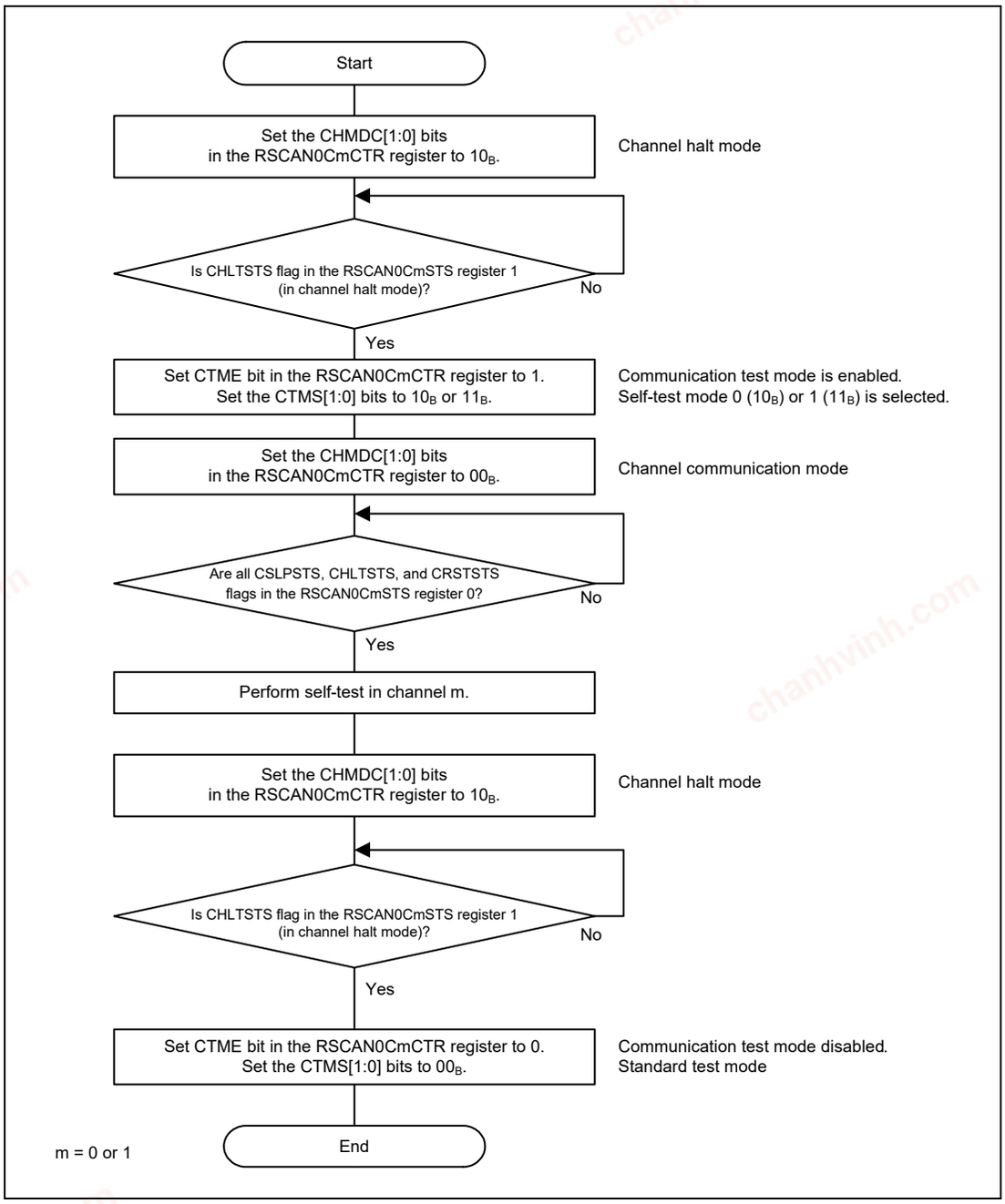


Figure 21.34 Self-Test Mode Setting Procedure

chanhvinh.com

21.10.4.2 Inter-Channel Communication Test Setting Procedure

Communication testing can be performed by transmitting and receiving data between different channels.

Figure 21.35 shows the inter-channel communication test setting procedure.

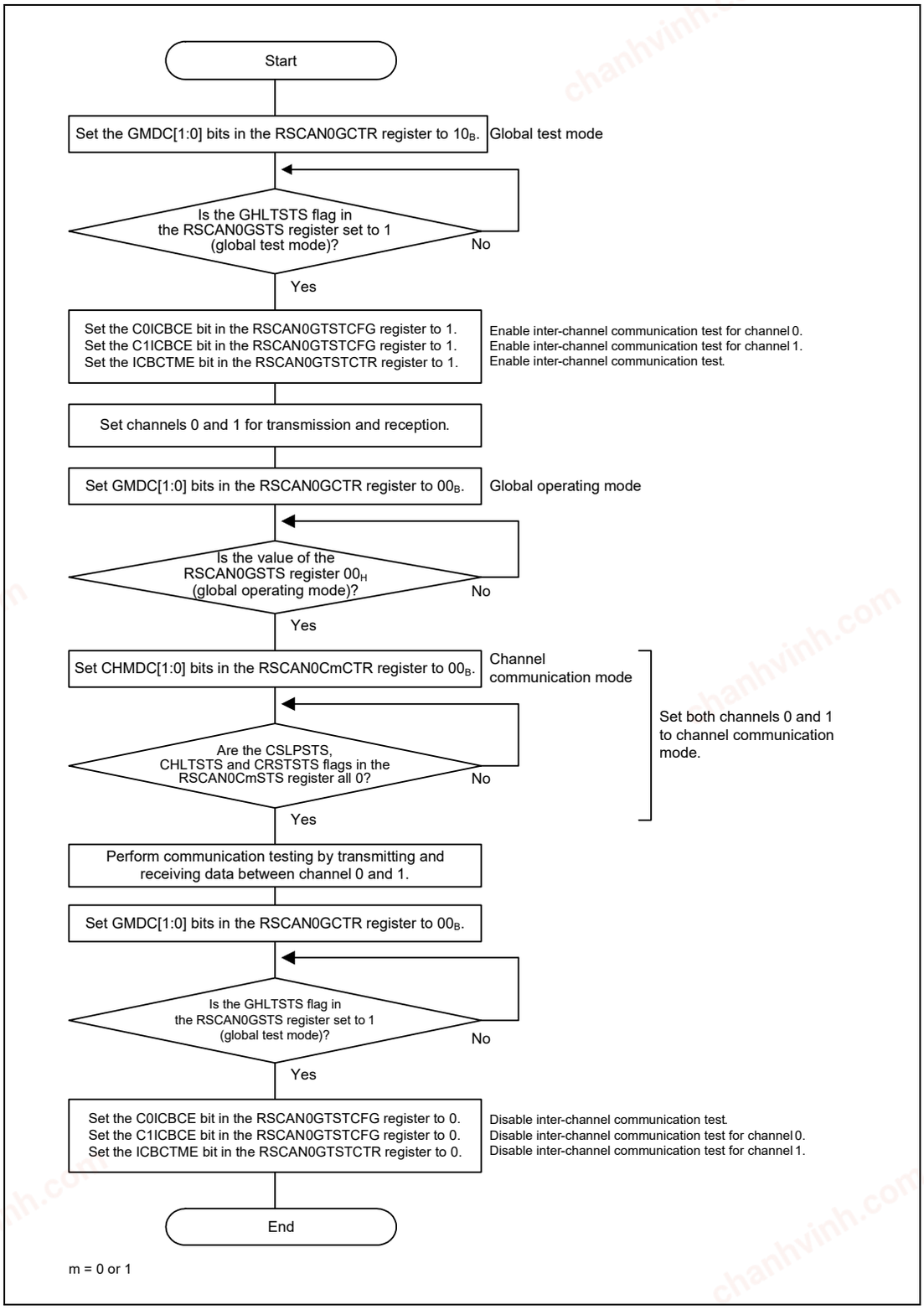


Figure 21.35 Inter-Channel Communication Test Setting Procedure(Example of Communication Test between Channel 0 and Channel 1)

chanhvinh.com

21.11 Notes on the RS-CAN Module

- When changing a global mode, check the GSLPSTS, GHLTSTS, and GRSTSTS flags in the RSCAN0GSTS register for transitions. When changing a channel mode, check the CSLPSTS, CHLTSTS, and CRSTSTS flags in the RSCAN0CmSTS register (m = 0 or 1) for transitions.
- The acceptance filter processing checks receive rules sequentially in ascending order from the minimum rule number. If the same ID, IDE bit, or RTR bit value is set for multiple receive rules, the minimum number of receive rule is used for the acceptance filter processing. If the message does not pass through the subsequent DLC filter processing, the data processing is terminated without returning to the acceptance filter processing and the message is not stored in the buffer.
- When linking transmit buffers to transmit/receive FIFO buffers or allocating transmit buffers to transmit queues, set the control register (RSCAN0TMCp) of the corresponding transmit buffer to 00_H. The status register (RSCAN0TMSTSp) of the corresponding transmit buffer should not be used. Flags in other status registers (registers RSCAN0TMTRSTS0, RSCAN0TMTARSTS0, RSCAN0TMTCSTS0, and RSCAN0TMTASTS0), which correspond to transmit buffers linked to transmit/receive FIFO buffers or allocated to transmit queues remain unchanged. Set the enable bit in the corresponding interrupt enable register (the RSCAN0TMIEC0 register) to 0 (transmit buffer interrupt is disabled).
- Transmit buffers that are linked to transmit/receive FIFO buffers must not bet allocated to transmit queues.
- Only a single transmit/receive FIFO buffer can be linked to a transmit buffer. Do not link two or more transmit/receive FIFO buffers to transmit buffers of the same number.
- When the CANm bit time clock is selected as a timestamp counter clock source, the timestamp counter stops when the corresponding channel has transitioned to channel reset mode or channel halt mode.
- In case of an attempt to store a new received message when the receive FIFO buffer and the transmit/receive FIFO buffer are full, the new message is discarded. If you wish to store a new transmit message in the transmit/receive FIFO buffer or the transmit queue, check that the transmit/receive FIFO buffer or the transmit queue is not full.
- The values of unused receive buffer registers (RSCAN0RMIDq, RSCAN0RMPTRq, RSCAN0RMDF0q, and RSCAN0RMDF1q registers), receive FIFO buffer access registers (RSCAN0RFIDx, RSCAN0RFPTRx, RSCAN0RFDF0x, and RSCAN0RFDF1x registers), and transmit/receive FIFO buffer access registers (RSCAN0CFIDk, RSCAN0CFPTRk, RSCAN0CFDF0k, and RSCAN0CFDF1k registers) are undefined when the RS-CAN module transitions to global operation mode or global test mode after exiting from global reset mode.

chanhvinh.com

22. IEBus Controller

This module is only incorporated in the RZ/A1L.

IEBus (Inter Equipment Bus) is a small-scale digital data transfer system that transfers data between units. To use IEBus with this LSI, an external IEBus driver and receiver are necessary because they are not provided.

The internal IEBus controller of this LSI is of negative logic.

22.1 IEBB Features

Channels This microcontroller has the following number of channels of the IEBB.

Table 22-1 Channels of IEBB

IEBB	
Number of channels	1
Name	IEBB0

Channel index n Throughout this chapter, the individual channels of IEBB are identified by the index "n" (n = 0), for example, IEBBnBCR for the IEBBn bus control register.

Register addresses All IEBBn register addresses are given as addresses offset from the base address <IEBBn_base>.

The base address <IEBBn_base> of each IEBBn is listed in the following table:

Table 22-2 Register base address <IEBBn_base>

IEBBn	<IEBBn_base>
IEBB0	FCFEF000

Clock supply The following clock is supplied to the IEBBn:

Table 22-3 IEBBn clock supply

IEBBn	Clock	Connected to:
IEBB0	P0ϕ	Clock pulse generator

chanhvinh.com

Interrupts and DMA The IEBB can generate the following interrupt requests and DMA requests:

Table 22-4 IEBBn interrupts and DMA requests

Interrupt request signal	Function	Direct memory access controller activation
IEBBTD	Data interrupt request	√
IEBBTV	Vector interrupt request	√
IEBBTERR	Error interrupt request	—
IEBBTSTA	Status interrupt request	—

I/O signals The I/O signals of the IEBB0 are listed in the following table:

Table 22-5 IEBBn I/O signals

Pin name	Function	Input/output
IERxD	IEBB0 reception data	Input
IETxD	IEBB0 transmission data	Output

chanhvinh.com

22.2 Configuration

22.2.1 Function overview

Features summary

- **The data transfer system complies with the IEBus (communication mode 1/ communication mode 2) protocol.**
- **The IEBus (Inter Equipment Bus) controller is mainly intended to transfer data between automotive devices by using a two-line serial bus interface.**
- Effective transmission speed: Approximately 18 kbps (communication mode 1), or approximately 27 kbps (communication mode 2)
- The single mode or FIFO mode can be selected.
- Maximum number of transferred bytes:

  32 bytes/frame (communication mode 1)
  128 bytes/frame (communication mode 2)
- To implement IEBus, an external IEBus driver and receiver are necessary. The driver and receiver are not built in.
- This IEBus controller uses negative logic.
- Operation clock: 8 MHz (input P0ϕ = 32 MHz to this module)
- Interrupt request signals
  - Data interrupt (IEBBTD)

    For transmission data write processing (single mode, FIFO mode)

    For reception data read processing (single mode)
  - Error interrupt (IEBBTERR)

    For error processing
  - Status interrupt (IEBBTSTA)

    Start interrupt processing (single mode)

    Status transmission interrupt (single mode)

    Communication completion interrupt (single mode, FIFO mode)

    Frame completion interrupt (single mode, FIFO mode)
  - Vector interrupt (IEBBTV)

    Occurs at the same time as IEBBTERR or IEBBTSTA (single mode)

    For reception data read processing (FIFO mode)
- Pin configuration
  - IERxD: IEBus reception data input signal
  - IETxD: IEBus transmission data output signal

chanhvinh.com

22.2.2 Block diagram

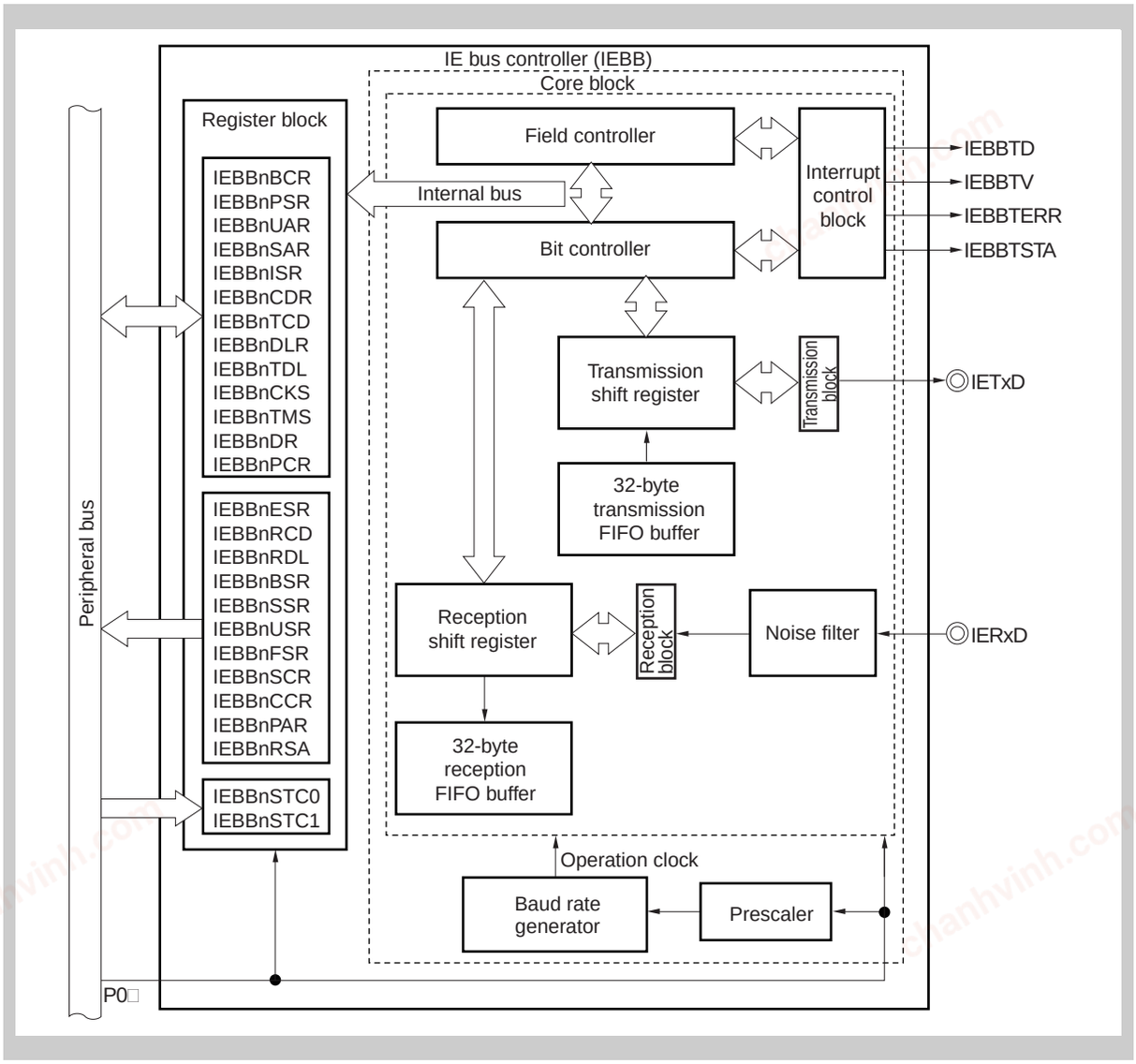


Figure 22-1 IEBBn block diagram

chanhvinh.com

22.3 Registers

22.3.1 IEBBn register overview

The IEBBn is controlled by the following registers:

Table 22-6 IEBBn registers

Register name	Symbol	Address
IEBBn bus control register	IEBBnBCR	<IEBBn_base> + 0000_H
IEBBn power save register	IEBBnPSR	<IEBBn_base> + 0004_H
IEBBn unit address register	IEBBnUAR	<IEBBn_base> + 0008_H
IEBBn slave address register	IEBBnSAR	<IEBBn_base> + $000C_H$
IEBBn partner address register	IEBBnPAR	<IEBBn_base> + 0010_H
IEBBn reception slave address register	IEBBnRSA	<IEBBn_base> + 0014_H
IEBBn control data register	IEBBnCDR	<IEBBn_base> + 0018_H
IEBBn transmission control data register	IEBBnTCD	<IEBBn_base> + $001C_H$
IEBBn reception control data register	IEBBnRCD	<IEBBn_base> + 0020_H
IEBBn message length register	IEBBnDLR	<IEBBn_base> + 0024_H
IEBBn transmission message length register	IEBBnTDL	<IEBBn_base> + 0028_H
IEBBn reception message length register	IEBBnRDL	<IEBBn_base> + $002C_H$
IEBBn clock selection register	IEBBnCKS	<IEBBn_base> + 0030_H
IEBBn transfer mode setting register	IEBBnTMS	<IEBBn_base> + 0034_H
IEBBn pointer clear register	IEBBnPCR	<IEBBn_base> + 0038_H
IEBBn buffer status register	IEBBnBSR	<IEBBn_base> + $003C_H$
IEBBn slave status register	IEBBnSSR	<IEBBn_base> + 0040_H
IEBBn unit status register	IEBBnUSR	<IEBBn_base> + 0044_H
IEBBn interrupt status register	IEBBnISR	<IEBBn_base> + 0048_H
IEBBn error status register	IEBBnESR	<IEBBn_base> + $004C_H$
IEBBn field status register	IEBBnFSR	<IEBBn_base> + 0050_H
IEBBn success count register	IEBBnSCR	<IEBBn_base> + 0054_H
IEBBn communication count register	IEBBnCCR	<IEBBn_base> + 0058_H
IEBBn status clear register 0	IEBBnSTC0	<IEBBn_base> + $005C_H$
IEBBn status clear register 1	IEBBnSTC1	<IEBBn_base> + 0060_H
IEBBn data register	IEBBnDR	<IEBBn_base> + 0064_H

<IEBBn_base> The IEBBn base address <IEBBn_base> is defined in Table 22-2.

chanhvinh.com

22.3.2 IEBBn control register details

(1) IEBBnBCR - IEBBn bus control register

The IEBBnBCR register is used to control the operation of IEBBn.

Access This register can be read or written in 8-bit units.

Address <IEBBn_base> + 0000_H

Initial value 00_H

The IEBBnMSRQ, IEBBnALRQ, IEBBnSTXE, and IEBBnSRXE bits are reset by writing 0 to the IEBBnPW bit.

Cautions

1. When operation is enabled (when the IEBBnPW bit = 1), writing 1 to the IEBBnMSRQ bit is prohibited while the IEBBnMSRQ bit = 1. To write 1 to this bit, first clear it to 0.
2. Note the following when accessing the register:
   - When the IEBBnPW bit = 0, it is not possible to write to the IEBBnMSRQ, IEBBnALRQ, IEBBnSTXE, and IEBBnSRXE bits.
   - Because the IEBBnMSRQ, IEBBnALRQ, IEBBnSTXE, and IEBBnSRXE bits are reset at the same time by writing 0 to the IEBBnPW bit, even if an 8-bit write that results in the IEBBnPW bit being cleared to 0 is performed, the IEBBnMSRQ, IEBBnALRQ, IEBBnSTXE, and IEBBnSRXE bits are not written to. When an 8-bit write that results in the IEBBnPW bit being set to 1 is performed, the IEBBnMSRQ, IEBBnALRQ, IEBBnSTXE, and IEBBnSRXE bits can be written to.

     Example: If 78H is written to the IEBBnBCR register and then the register is read, 00H is returned.

     If F8H is written to the IEBBnBCR register and then the register is read, F8H is returned.

7	6	5	4	3	2	1	0
IEBBn PW	IEBBn MSRQ	IEBBn ALRQ	IEBBn STXE	IEBBn SRXE	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R

chanhvinh.com

Table 22-7 IEBBnBCR register contents

Bit position	Bit name	Function
7	IEBBnPW	Communication enable flag 0: Stop IEBBn unit operation. 1: Enable IEBBn unit operation. Caution When the IEBBnPW bit is set (to 1), the IEBBnPSR, IEBBnUAR, IEBBnCKS, and IEBBnTMS registers below cannot be overwritten. Therefore, these registers must be set up before setting the IEBBnPW bit.
6	IEBBnMSRQ	Master request flag 0: Do not request the IEBBn unit as the master. 1: Request the IEBBn unit as the master.
5	IEBBnALRQ	Broadcast request flag 0: Request individual communication. 1: Request broadcast communication.
4	IEBBnSTXE	Slave transmission enable flag 0: Disable slave transmission. 1: Enable slave transmission.
3	IEBBnSRXE	Slave reception enable flag 0: Disable slave reception. 1: Enable slave reception.

chanhvinh.com

(a) Communication enable flag (IEBBnPW): Bit 7

- Set/clear condition

  Set: By software (Write 1 to the IEBBnPW bit.)
  Clear: By software (Write 0 to the IEBBnPW bit.)

Depending on when the IEBBnPW bit is set (to 1), the IEBBn communication participation method differs.

Table 22-8 IEBBnPW bit setting timing and communication participation method

Timing for setting the IEBBnPW bit (to 1)	IEBBn communication participation method
When communication is not being performed on IEBus	Communication is participated in starting at the next frame or communication is started.
When communication is being performed on IEBus, and start bit communication is being performed by another bus master	Participates in communication from that frame if the start bit is detected. If the start bit is not detected, participates in communication from the next frame.
When communication is being performed on IEBus, and post-start bit communication is being performed by another bus master	Participates in communication from the next frame.

If the IEBBnPW bit is cleared (to 0), communication is immediately stopped even if it is in progress, and the internal flags and registers are reset, with some exceptions. The registers that are not reset by the IEBBnPW bit are shown below.

When the IEBBnPW = 0, even if another unit starts communication, IEBBn does not respond.

Table 22-9 Registers that are not reset by the IEBBnPW bit

Registers that are not reset by the IEBBnPW bit	Remark
IEBBnPSR	Not reset
IEBBnUAR	Not reset
IEBBnSAR	Not reset
IEBBnCDR	Data written from the CPU is not reset but data received during communication is.
IEBBnTCD	Not reset
IEBBnDLR	Data written from the CPU is not reset but data received during communication is.
IEBBnTDL	Not reset
IEBBnCKS	Not reset
IEBBnTMS	Not reset
IEBBnPCR	Not reset
IEBBnSTC0	Not reset
IEBBnSTC1	Not reset
IEBBnDR	Data written from the CPU is not reset but data received during communication is.

chanhvinh.com

(b) Master request flag (IEBBnMSRQ): Bit 6

- Set/clear condition

  Set: By software

  Clear:

  – Single mode:

    The flag is cleared (to 0) by hardware when master communication is started and when the start interrupt of the master occurs.

    The flag is cleared (to 0) by hardware when a communication error interrupt occurs (when the IEBBnISR.IEBBnIEBE bit = 1).

    The flag is cleared (to 0) by hardware when arbitration loss occurs.

    The flag is cleared (to 0) when the IEBBnPW bit is cleared.

  – FIFO mode:

    The flag is cleared (to 0) by hardware after master communication starts, communication is performed without arbitration loss occurring, and the parity bit of the slave address field output by the unit is transmitted.

    The flag is cleared (to 0) by hardware when a communication error interrupt occurs (when the IEBBnISR.IEBBnIEBE bit = 1).

    The flag is cleared (to 0) by hardware if arbitration losses consecutively occur the number of times specified by the IEBBnTMS.IEBBnALC2 to IEBBnALC0 bits.

    The flag is cleared (to 0) when the IEBBnPW bit is cleared.

When the IEBBnMSRQ bit is set (to 1), the IEBus controller starts communication on IEBus as the master.

If communication is in progress on IEBus (if the start bit cannot be detected while the start bit is being communicated or if communication is in progress after the start bit has been detected), however, the controller waits until the current frame ends (holds the master request pending), outputs the start bit after the frame has ended, and starts communication as the master.

Cautions

1. Only set the IEBBnMSRQ bit after clearing the IEBBnSTXE bit to 0. After setting the IEBBnSTXE bit to 1, if arbitration loss occurs and the slave is selected, the transmission data prepared for the master might be used as slave transmission data.
2. Reissue master requests in the single mode and FIFO mode as described below.
   - Single mode:
     When arbitration is lost, use software to reissue master requests.
   - FIFO mode:
     When arbitration is lost, use hardware to reissue master requests.

     However, if arbitration losses consecutively occur the number of times specified by the IEBBnTMS.IEBBnALC2 to IEBBnALC0 bits, the request must be reissued by using software.

chanhvinh.com

(c) Broadcast request flag (IEBBnALRQ): Bit 5

- Set/clear condition

  Set: By software
  Clear: By software

Cautions
1. The IEBBnMSRQ bit is cleared (to 0) by hardware, but the IEBBnALRQ bit is not. Therefore, if the next master request is for individual communication, clear the IEBBnALRQ bit (to 0).
2. Be sure to change the value of the IEBBnALRQ bit before setting the IEBBnMSRQ bit (to 1).

(d) Slave transmission enable flag (IEBBnSTXE): Bit 4

- Set/clear condition

  Set: By software
  Clear: By software

Slave transmission is controlled by the value of the slave transmission enable flag, but whether IEBBn performs slave transmission (whether there is an $\overline{\text{ACK}}$ signal response for the control field) is determined by other conditions.

The $\overline{\text{ACK}}$ signal response conditions for the control field are shown below.

Table 22-10 Control field $\overline{\text{ACK}}$ signal response conditions (when the received control data is 0H, 3H, 4H, 5H, 6H, or 7H)

Communication target (IEBBnUSR. IEBBnSRQF bit) Slave specification = 1 No specification = 0	Lock status (IEBBnUSR. IEBBnLCKF bit) Lock = 1 No lock = 0	Master unit judgment (IEBBnPAR register match) Lock request unit = 1 Other = 0	Slave transmission enabled (IEBBnBCR. IEBBnSTXE bit)	Slave reception enabled (IEBBnBCR. IEBBnSRXE bit)	Received Control Data					
					0H	3H	4H	5H	6H	7H
1	0	don't care	0	don't care	A	N	N	N	A	N
			1		A	A	N	N	A	A
	1	0	don't care		A	N	A	A	N	N
		1	0		A	N	A	A	A	N
			1		A	A	A	A	A	A
Other than the above					N					

Note A: Slave transmission is performed. (The $\overline{\text{ACK}}$ signal is returned.)
N: Slave transmission is not performed. (The NACK signal is returned.)

chanhvinh.com

Slave transmission is not performed if the received control data is AH, BH, EH, or FH.

Table 22-11 Control field $\overline{\text{ACK}}$ signal response conditions (when the received control data is AH, BH, EH, or FH)

Communication target (IEBBnUSR. IEBBnSRQF bit) Slave specification = 1 No specification = 0	Lock status (IEBBnUSR. IEBBnLCKF bit) Lock = 1 No lock = 0	Master unit judgment (IEBBnPAR register match) Lock request unit = 1 Other = 0	Slave transmission enabled (IEBBnBCR. IEBBnSTXE bit)	Slave reception enabled (IEBBnBCR. IEBBnSRXE bit)	Received control data			
					AH	BH	EH	FH
1	0	don't care	don't care	1	A			
	1	1						
Other than the above					N			

Note A: The $\overline{\text{ACK}}$ signal is returned.
N: The NACK signal is returned.

Cautions
1. Set the IEBBnSTXE bit before the control field parity bit is received.
2. When there is a master request, clear the IEBBnSTXE bit (to 0) before setting the IEBBnMSRQ bit (to 1). This is to avoid transmission of the data of the IEBBnDR register that tries master transmission if the controller loses arbitration after master operation and if slave transmission is requested by the master.

chanhvinh.com

(e) Slave reception enable flag (IEBBnSRXE): Bit 3

- Set/clear condition

  Set: By software
  Clear: By software

When the IEBBnSRXE bit = 1 and the reception control data for the communicated control field addressed to the unit is AH, BH, EH, or FH (or when the lock status is specified and the master unit address of the communication matches the address for which a lock was requested), the ACK signal is returned for the control field, and a slave reception operation is performed.

When the IEBBnSRXE bit = 0 and the reception control data for the communicated control field addressed to the unit is AH, BH, EH, or FH, the NACK signal is returned for the control field, and no slave reception operation is performed.

Cautions

1. Set the IEBBnSRXE bit before the control field parity bit is received.
2. The IEBBnSRXE bit is used to enable or disable slave reception for both individual and broadcast communication. For individual communication, a NACK signal can be returned for the control field to end communication by clearing the IEBBnSRXE bit to 0 (thereby prohibiting slave reception), but, for broadcast communication, although communication cannot be ended by clearing this bit because no $\overline{\text{ACK}}$/NACK signal is transmitted, no data interrupt occurs because IEBBn does not respond to the broadcast communication.

chanhvinh.com

(2) IEBBnPSR - IEBBn power save register

The IEBBnPSR register is used to operate and stop the IEBBn operation clock and to control the communication mode.

Access This register can be read or written in 8-bit units.

Address <IEBBn_base> + 0004_H

Initial value 00_H

Cautions
1. The IEBBnPSR register can only be set up when the IEBBnBCR.IEBBnPW bit = 0. Do not set up the register when this bit = 1. If an attempt is made to set up the register when the IEBBnPW bit = 1, the value is ignored.
2. To use IEBBn, first set the IEBBnCLKE bit (to 1) and enable the operation clock.

   To start the bus operation, specify the settings below.

   • When communication has started

   1. Set up the IEBBnCKS register.
   2. Set the IEBBnCLKE bit (to 1). (The operation clock operates.)

      Set the IEBBnCMD bit to 0 or 1 to specify the communication mode.
   3. Set up registers such as IEBBnUAR, IEBBnSAR, IEBn0TCD, IEBBnTDL, and IEBBnDR depending on the type of communication.
   4. Set the IEBBnBCR.IEBBnPW bit (to 1) to start communication.

   • When communication is stopped

   1. Clear the IEBBnPW bit to 0.
   2. Clear the IEBBnCLKE bit (to 0). (The operation clock stops.)

7	6	5	4	3	2	1	0
IEBBn CLKE	IEBBn CMD	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R

Table 22-12 IEBBnPSR register contents

Bit position	Bit name	Function
7	IEBBnCLKE	Operation clock enable flag 0: Stop the operation clock. (This makes it possible to reduce the power consumed by IEBBn.) Initialize the prescaler and baud rate generator. 1: Enable the operation clock. The operation clock starts operating one clock cycle after the IEBBnCLKE bit is set (to 1). (For details, see Figure 22-2.) Similarly, the operation clock stops operating one clock cycle after the IEBBnCLKE bit is cleared (to 0). (For details, see Figure 22-2.)
6	IEBBnCMD	IEBBn communication mode setting flag 0: Specify communication mode 1. 1: Specify communication mode 2.

chanhvinh.com

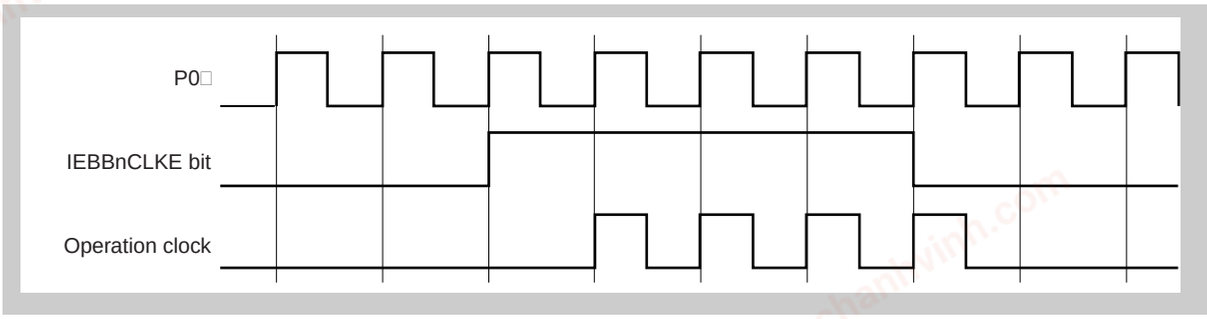


Figure 22-2 Starting and stopping the operation clock

chanhvinh.com

(3) IEBBnUAR - IEBBn unit address register

The IEBBnUAR register is used to specify the unit address of the IEBus unit.

This register must always be set before starting communication.

Specify the unit address (12 bits) for bits 11 to 0.

Access This register can be read or written in 16-bit units.

Address <IEBBn_base> + 0008_H

Initial value 0000_H

Cautions
1. The IEBBnUAR register can only be set up when the IEBBnBCR.IEBBnPW bit = 0. Do not set up the register when this bit = 1. If an attempt is made to set up the register when the IEBBnPW bit = 1, the value is ignored.
2. Writing to this register in 8-bit units is prohibited.

15	14	13	12	11	10	9	8
0	0	0	0				
R	R	R	R	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

chanhvinh.com

(4) IEBBnSAR - IEBBn slave address register

The IEBBnSAR register is used to specify the address of the communication-partner slave unit during master communication.

During a master request, the value of this register is transmitted as the slave address field data.

Specify the slave address (12 bits) for bits 11 to 0.

Access This register can be read or written in 16-bit units.

Address <IEBBn_base> + $000C_H$

Initial value 0000_H

Cautions

1. When the IEBBnSAR register is overwritten during communication (while the IEBBnBCR.IEBBnPW bit = 1), communication might not be correctly performed. Therefore, overwriting is prohibited from when a master request is issued until the communication or frame completion timing.

   Note that overwriting is enabled at the following times:

   - When the IEBBnPW bit = 0
   - From when the IEBBnPW bit is set to 1 until the first master request (when the IEBBnMSRQ bit = 1)
   - From the communication or frame completion timing (assuming the IEBBnPW bit = 1 and the IEBBnMSRQ bit = 0) until the next master request (when the IEBBnMSRQ bit = 1)

2. Writing to this register in 8-bit units is prohibited.

15	14	13	12	11	10	9	8
0	0	0	0				
R	R	R	R	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

chanhvinh.com

(5) IEBBnPAR - IEBBn partner address register

The IEBBnPAR register is used to store the reception master address in the master address field.

When operating the unit is enabled (when the IEBBnBCR.IEBBnPW bit = 1), the received master address is stored using the master address field regardless of whether the unit master is operating or the slave is operating.

- Storage in the single mode

  Upon completion of the parity period for the master address field, this is only performed if the parity value is normal and the unit is in the non-lock status.

- Storage in the FIFO mode

  If reading the data received during the previous communication has finished (if the IEBBnBSR.IEBBnRFLF bit = 0 and the IEBBnBSR.IEBBnSRFP4 to IEBBnSRFP0 bits = 00000), upon completion of the parity period for the master address field, storage is only performed if the parity value is normal and the unit is in the non-lock status. If reading the received data has not finished, the IEBBnPAR register is not updated until it finishes.

When there is a unit lock, because the address of the unit that requested the lock (the lock master) is retained, the IEBBnPAR register is not updated.

- Lock address transmission request reception in the single mode

  If a lock address transmission request is received from the master as a status transmission request, when the received control data receives the lock address (higher four bit) read request (5H), the value of the IEBBnPAR register is read by using software, and then the data in bits 15 to 8 of the IEBBnPAR register is written to the IEBBnDR register.
  In addition, if a lock address (lower 8 bits) read request (4H) is received, the value of the IEBBnPAR register is read by using software, and then the data in bits 7 to 0 of the IEBBnPAR register is written to the IEBBnDR register.

- Lock address transmission request reception in the FIFO mode

  If a lock address transmission request is received from the master as a status transmission request, the data in the IEBBnPAR register is automatically transmitted to the data field by using hardware.

Specify the partner address (12 bits) for bits 11 to 0.

Access This register is read-only, in 16-bit units.

Address <IEBBn_base> + 0010_H

Initial value 0000_H

This register is reset when the IEBBnBCR.IEBBnPW bit is overwritten.

15	14	13	12	11	10	9	8
0	0	0	0				
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
R	R	R	R	R	R	R	R

chanhvinh.com

(6) IEBBnRSA - IEBBn reception slave address register

The IEBBnRSA register is used to store the slave address value received using the slave address field.

When operating the unit is enabled (when the IEBBnBCR.IEBBnPW bit = 1), the received slave address is stored using the slave address field regardless of whether the unit master is operating or the slave is operating.

- Storage in the single mode

  This is performed upon the completion of the slave address field parity period if the parity value is normal.

- Storage in the FIFO mode

  If reading the data received during the previous communication has finished (if the IEBBnBSR.IEBBnRFLF bit = 0 and the IEBBnBSR.IEBBnSRFP4 to IEBBnSRFP0 bits = 00000), upon completion of the parity period for the slave address field, storage is only performed if the parity value is normal. Until reading the received data finishes, the IEBBnRSA register is not updated.

Specify the slave address (12 bits) for bits 11 to 0.

Access This register is read-only, in 16-bit units.

Address <IEBBn_base> + 0014_H

Initial value 0000_H

This register is reset when the IEBBnBCR.IEBBnPW bit is overwritten.

15	14	13	12	11	10	9	8
0	0	0	0				
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
R	R	R	R	R	R	R	R

chanhvinh.com

(7) IEBBnCDR - IEBBn control data register

The IEBBnCDR register is used to specify the control data transmitted using the control field.

After writing to the IEBBnCDR register, the IEBBnTCD register is written to.

After reading the IEBBnCDR register, the IEBBnRCD register value is read.

Access This register can be read or written in 8-bit units.

Address <IEBBn_base> + 0018_H

Initial value 00_H

The read value is reset when the IEBBnBCR.IEBBnPW bit is overwritten.

Caution When issuing a master request, be sure to set up the IEBBnCDR register before starting communication (when the IEBBnBCR.IEBBnMSRQ bit = 0).

Note The IEBBnCDR register consists of a write register and a read register. Therefore, data written to this register cannot be read as is. The data received during IEBus communication can be read.

7	6	5	4	3	2	1	0
0	0	0	0	IEBBn SLCD3	IEBBn SLCD2	IEBBn SLCD1	IEBBn SLCD0
R	R	R	R	R/W	R/W	R/W	R/W

chanhvinh.com

(8) IEBBnTCD - IEBBn transmission control data register

The IEBBnTCD register is used to specify the control data transmitted using the control field.

The value of the lower 4 bits of the value written to the IEBBnTCD register is transmitted as control data by using the control field during master transmission.

Access This register can be read or written in 8-bit units.

Address <IEBBn_base> + $001C_H$

Initial value 00_H

Cautions
1. When issuing a master request, be sure to set up the IEBBnTCD register before starting communication (when the IEBBnBCR.IEBBnMSRQ bit = 0).
2. Do not specify undefined values.
3. During broadcast transmission, specifying slave transmission control data is prohibited.

7	6	5	4	3	2	1	0
0	0	0	0	IEBBn SLTD3	IEBBn SLTD2	IEBBn SLTD1	IEBBn SLTD0
R	R	R	R	R/W	R/W	R/W	R/W

Table 22-13 IEBBnTCD register contents

Bit position	Bit name	Function
3 to 0	IEBBnSLTD [3-0]	Specify the control data transmitted by using the control field.

IEBBn SLTD3	IEBBn SLTD2	IEBBn SLTD1	IEBBn SLTD0	Function
0	0	0	0	Read slave status
0	0	0	1	Undefined
0	0	1	0	Undefined
0	0	1	1	Data reading and locking
0	1	0	0	Lock address reading (lower 8 bits)
0	1	0	1	Lock address reading (higher 4 bits)
0	1	1	0	Slave status reading and unlocking
0	1	1	1	Read data
1	0	0	0	Undefined
1	0	0	1	Undefined
1	0	1	0	Command writing and locking
1	0	1	1	Data writing and locking
1	1	0	0	Undefined
1	1	0	1	Undefined
1	1	1	0	Write command
1	1	1	1	Write data

chanhvinh.com

(9) IEBBnRCD - IEBBn reception control data register

The IEBBnRCD register is used to store the control data received using the control field.

The data received by using the control field is read to the lower 4 bits of the IEBBnRCD register. Data is stored in the IEBBnRCD register upon completion of the control field parity period if the parity value is normal.

- Storage in the single mode

  When a status transmission request is received, the user performs each process (settings for the transmission data of the IEBBnSSR register or the IEBBnPAR register) according to the value of the lower 4 bits of the IEBBnRCD register read value.

- Storage in the FIFO mode

  When a status transmission request is received, the hardware automatically performs the status transmission processing (settings for the transmission data of the IEBBnSSR register or the IEBBnPAR register).

Because it is necessary to judge whether the received data is a command or data, be sure to read the value of this register upon the completion of communication.

In the FIFO mode, if reading the data received during the previous communication has finished (if the IEBBnBSR.IEBBnRFLF bit = 0 and the IEBBnBSR.IEBBnSRFP4-SRFP0 bit = 00000), upon completion of the parity period for the control data field, storage is performed if the parity value is normal. Until reading the received data finishes, the IEBBnRCD register is not updated.

Access This register is read-only, in 8-bit units.

Address <IEBBn_base> + 0020_H

Initial value 00_H

This register is reset when the IEBBnBCR.IEBBnPW bit is overwritten.

7	6	5	4	3	2	1	0
0	0	0	0	IEBBn SLRD3	IEBBn SLRD2	IEBBn SLRD1	IEBBn SLRD0
R	R	R	R	R	R	R	R

chanhvinh.com

Table 22-14 IEBBnRCD register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3 to 0</td><td>IEBBnSLRD [3-0]</td><td>Specify the control data received by using the control field.
<table>
<tr><th>IEBBn SLRD3</th><th>IEBBn SLRD2</th><th>IEBBn SLRD1</th><th>IEBBn SLRD0</th><th>Function</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>Read slave status</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>Undefined</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>Undefined</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>Data reading and locking</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>Lock address reading (lower 8 bits)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>Lock address reading (higher 4 bits)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>Slave status reading and unlocking</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>Read data</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>Undefined</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>Undefined</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>Command writing and locking</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>Data writing and locking</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>Undefined</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>Undefined</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>Write command</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>Write data</td></tr>
</table>
</td></tr>
</table>

chanhvinh.com

(10) IEBBnDLR - IEBBn message length register

The IEBBnDLR register is used to specify the message length data transmitted using the message length field.

After writing to the IEBBnDLR register, the IEBBnTDL register is written to.

After reading the IEBBnDLR register, the IEBBnRDL register value is read.

Access This register can be read or written in 8-bit units.

Address <IEBBn_base> + 0024_H

Initial value 01_H

The read value is reset when the IEBBnBCR.IEBBnPW bit is overwritten.

Caution When issuing a master request, be sure to set up the IEBBnDLR register before starting communication (when the IEBBnBCR.IEBBnMSRQ bit = 0).

Note The IEBBnDLR register consists of a write register and a read register. Therefore, data written to this register cannot be read as is. The data received during IEBus communication can be read.

7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

chanhvinh.com

(11) IEBBnTDL - IEBBn transmission message length register

The IEBBnTDL register is used to specify the message length data transmitted using the message length field.

The value written to the IEBBnTDL register is transmitted as message length data by using the message length field if the unit is the transmission unit (master transmission, slave transmission).

However, when a status transmission request is received, 0H is transmitted as the message length data regardless of the IEBBnTDL register setting.

Access This register can be read or written in 8-bit units.

Address <IEBBn_base> + 0028_H

Initial value 01_H

Cautions

1. Be sure to set up the IEBBnTDL register before starting communication (when the IEBBnBCR.IEBBnMSRQ bit = 0).
2. The maximum number of bytes that can be transferred per frame is determined according to the communication mode. For example, when transferring 48 bytes in communication mode 1, perform communication by dividing the data among multiple frames. In this case, when performing the second communication, use the IEBBnSCR register to check the number of data bytes transmitted during the first communication, subtract the number of bytes that were successfully transmitted from the number of bytes you want to transmit, and then specify the result for the IEBBnTDL register. Write the next data to the IEBBnDR register at the same time, and then issue a master request.

7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22-15 IEBBnTDL register contents

Bit								Setting	Number of remaining communication data bytes
7	6	5	4	3	2	1	0		
0	0	0	0	0	0	0	1	01H	1 byte
0	0	0	0	0	0	1	0	02H	2 bytes
...	...	...	...	...	...	...	...	...	...
0	0	0	1	0	1	0	0	20H	32 bytes
...	...	...	...	...	...	...	...	...	...
1	1	1	1	1	1	1	1	FFH	255 bytes
0	0	0	0	0	0	0	0	00H	256 bytes

chanhvinh.com

(12) IEBBnRDL - IEBBn reception message length register

The IEBBnRDL register is used to specify the message length data received using the message length field.

The IEBBnRDL register read value is the data received using the message length field.

- Storage in the single mode

  Storage proceeds if the parity value is normal at the end of the parity period for the message length field.

- Storage in the FIFO mode

  If reading the data received during the previous communication has finished (IEBBnBSR.IEBBnRFLF bit = 0 and the IEBBnBSR.IEBBnSRFP4-SRFP0 bit = 00000), upon completion of the parity period for the message length field, storage is performed if the parity value is normal. Until reading the received data finishes, the IEBBnRDL register is not updated.

Access This register is read-only, in 8-bit units.

Address <IEBBn_base> + $002C_H$

Initial value 01_H

This register is reset when the IEBBnBCR.IEBBnPW bit is overwritten.

7	6	5	4	3	2	1	0
R	R	R	R	R	R	R	R

Table 22-16 IEBBnRDL register contents

Bit								Setting	Number of remaining communication data bytes
7	6	5	4	3	2	1	0		
0	0	0	0	0	0	0	1	01H	1 byte
0	0	0	0	0	0	1	0	02H	2 bytes
...	...	...	...	...	...	...	...	...	...
0	0	0	1	0	1	0	0	20H	32 bytes
...	...	...	...	...	...	...	...	...	...
1	1	1	1	1	1	1	1	FFH	255 bytes
0	0	0	0	0	0	0	0	00H	256 bytes

chanhvinh.com

(13) IEBBnCKS - IEBBn clock selection register

The IEBBnCKS register is used to control the clock selection of the IEBus controller.

Access This register can be read or written in 8-bit units.

Address <IEBBn_base> + 0030_H

Initial value 17_H

Caution The IEBBnCKS register can only be set up when the IEBBnBCR.IEBBnPW bit = 0 and the IEBBnPSR.IEBBnCLKE bit = 0. Do not set up the register when this bit = 1. If an attempt is made to set up the register when the IEBBnPW bit = 1, the value is ignored.

7	6	5	4	3	2	1	0
0	0	0	IEBBn PRS	0	IEBBn BRS2	IEBBn BRS1	IEBBn BRS0
R	R	R	R/W	R	R/W	R/W	R/W

Table 22-17 IEBBnCKS register contents

Bit position	Bit name	Function
4	IEBBnPRS	Specify the prescaler output (PRSOUT). 0: P0ϕ 1: P0ϕ2 Caution The conditions under which the prescaler is initialized are as follows: • When the IEBBnPRS bit is overwritten • When the IEBBnBCR.IEBBnPW bit = 0 and the IEBBnPSR.IEBBnCLKE bit = 1
2 to 0	IEBBnBRS [2-0]	Specify the operation clock output (MCK).

IEBBnBRS2	IEBBnBRS1	IEBBnBRS0	Operation clock output (MCK)
0	0	0	PRSOUT/1
0	0	1	PRSOUT/1
0	1	0	PRSOUT/2
0	1	1	PRSOUT/3
1	0	0	PRSOUT/4
1	0	1	PRSOUT/5
1	1	0	PRSOUT/6
1	1	1	PRSOUT/7

Caution The conditions under which the baud rate generator is initialized are as follows:
- When the IEBBnBRS2 to IEBBnBRS0 bits are overwritten
- When the IEBBnBCR.IEBBnPW bit = 0 and the IEBBnPSR.IEBBnCLKE bit = 1

chanhvinh.com

Table 22-18 Input clock specification example

P0φ	IEBBnPRS	IEBBnBRS2	IEBBnBRS1	IEBBnBRS0	Specified value
32 MHz	0	1	0	0	04H

Caution The IEBus controller is designed to operate at 8 MHz.

chanhvinh.com

(14) IEBBnTMS - IEBBn transfer mode setting register

The IEBBnTMS register is used to control the IEBus controller communication operations.

Access This register can be read or written in 8-bit units.

Address <IEBBn_base> + 0034_H

Initial value 01_H

Caution The IEBBnTMS register can only be set up when the IEBBnBCR.IEBBnPW bit = 0. Do not set up the register when this bit = 1. If an attempt is made to set up the register when the IEBBnPW bit = 1, the value is ignored.

7	6	5	4	3	2	1	0
IEBBn FMDE	IEBBn SLRI1	IEBBn SLRI0	IEBBn SLTI1	IEBBn SLTI0	IEBBn ALC2	IEBBn ALC1	IEBBn ALC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22-19 IEBBnTMS register contents (1/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7</td><td>IEBBnFMDE</td><td>Specify whether to enable or disable FIFO mode operation.
0: Disable FIFO mode operation (single mode).
1: Enable FIFO mode operation.</td></tr>
<tr><td>6, 5</td><td>IEBBnSLRI [1, 0]</td><td>Specify the IEBBTV occurrence timing during FIFO mode reception.
<table>
<tr><th>IEBBnSLRI1</th><th>IEBBnSLRI0</th><th>IEBBTV occurrence timing</th></tr>
<tr><td>0</td><td>0</td><td>When the reception data that has not been read from the reception FIFO buffer reaches 32 bytes</td></tr>
<tr><td>0</td><td>1</td><td>When the reception data that has not been read from the reception FIFO buffer reaches 24 bytes</td></tr>
<tr><td>1</td><td>0</td><td>When the reception data that has not been read from the reception FIFO buffer reaches 16 bytes</td></tr>
<tr><td>1</td><td>1</td><td>When the reception data that has not been read from the reception FIFO buffer reaches 8 bytes</td></tr>
</table></td></tr>
<tr><td>4, 3</td><td>IEBBnSLTI [1, 0]</td><td>Specify the IEBBTD occurrence timing during FIFO mode transmission.
<table>
<tr><th>IEBBnSLTI1</th><th>IEBBnSLTI0</th><th>IEBBTD occurrence timing</th></tr>
<tr><td>0</td><td>0</td><td>When the transmission FIFO buffer becomes empty</td></tr>
<tr><td>0</td><td>1</td><td>When the untransmitted data remaining in the transmission FIFO buffer reaches 2 bytes</td></tr>
<tr><td>1</td><td>0</td><td>When the untransmitted data remaining in the transmission FIFO buffer reaches 4 bytes</td></tr>
<tr><td>1</td><td>1</td><td>When the untransmitted data remaining in the transmission FIFO buffer reaches 8 bytes</td></tr>
</table></td></tr>
</table>

chanhvinh.com

Table 22-19 IEBBnTMS register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>2 to 0</td><td>IEBBnALC
[2-0]</td><td>Specify the arbitration loss number.
This is only valid in the FIFO mode. The settings of these bits are invalid in the single mode.
When arbitration is lost and the counter is set to 0H, the IEBBnBCR.IEBBnMSRQ bit is not retained.
The settings of the IEBBnALC2 to IEBBnALC0 bits are always retained. The settings are not changed each time arbitration is lost.
The arbitration loss counter is decremented separately from the IEBBnALC2 to IEBBnALC0 bits.
Overwriting the IEBBnALC2 to IEBBnALC0 bits while the arbitration loss counter is counting does not affect the counter. (The values of the bits are specified for the arbitration loss counter when the IEBBnMSRQ bit = 1.)
<table>
<tr><th>IEBBnALC2</th><th>IEBBnALC1</th><th>IEBBnALC0</th><th>Maximum arbitration loss count</th></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>3</td></tr>
<tr><td colspan="3">Other than the above</td><td>Setting prohibited</td></tr>
</table>
</td></tr>
</table>

chanhvinh.com

(a) FIFO operation enable bit (IEBBnFMDE): Bit 7

Differences between operation in the single mode and FIFO mode are shown below.

Table 22-20 Differences between operation in the single mode and FIFO mode

Transfer mode	Arbitration loss master request flag (IEBBnBCR.IEBBnMSRQ)	Arbitration loss error interrupt signal (IEBBTERR)	Support for slave status requests and lock address requests
Single mode	Clear	• The signal is not output. • The IEBBnESR.IEBBnABTE bit is fixed to 0.	During the slave status interrupt servicing, the value of the IEBBnSSR or IEBBnPAR register is written to the IEBBnDR register.
FIFO mode	The flag value is retained until arbitration is lost the number of times specified for the IEBBnALC2 to IEBBnALC0 bits. (The flag is cleared unless an error occurs during communication between third parties.)	• The signal is output if arbitration is lost the specified number of times. • The IEBBnESR.IEBBnABTE bit is simultaneously set (to 1).	The value of the IEBBnSSR or IEBBnPAR register is automatically sent by hardware by using the data field.

(b) FIFO mode reception IEBBTV occurrence timing specification bits (IEBBnSLRI1, IEBBnSLRI0): Bits 6, 5

In communication mode 1, the IEBBnSLRI1 and IEBBnSLRI0 bits are cleared to 00.

In communication mode 2, because data that exceeds 32 bytes is received, data must be read during reception. The occurrence timing of the data interrupt (IEBBTV) that requests that received data be read is specified by the IEBBnSLRI1 and IEBBnSLRI0 bits.

The IEBBTV interrupt occurs when received data is stored in the FIFO buffer and the number of unread bytes of data reaches the value specified for the IEBBnSLRI1 and IEBBnSLRI0 bits.

For the data no less than the number of bytes specified for the IEBBnSLRI1 and IEBBnSLRI0 bits, the data is read, and no interrupt occurs even if the number of unread bytes of received data exceeds the value specified for the IEBBnSLRI1 and IEBBnSLRI0 bits.

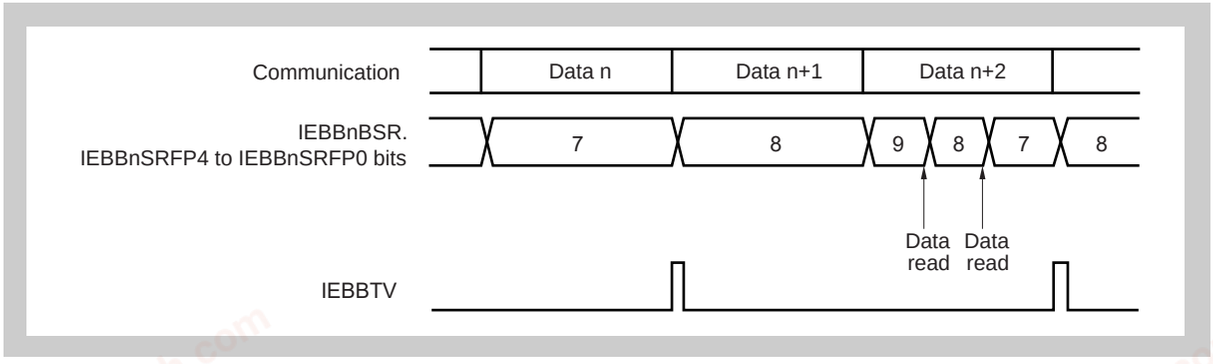


Figure 22-3 Example of operation in communication mode 2: When the IEBBnSLRI1 and IEBBnSLRI0 bits = 11

chanhvinh.com

(c) FIFO mode transmission IEBBTB occurrence timing specification bits (IEBBnSLTI1, IEBBnSLTI0): Bits 4, 3

In communication mode 1, clear the IEBBnSLTI1 and IEBBnSLTI0 bits to 00.

In communication mode 2, because data that exceeds 32 bytes is transmitted, transmission data must be written during transmission. The occurrence timing of the data interrupt (IEBBTD) that requests that transmission data be written is specified by the IEBBnSLTI1 and IEBBnSLTI0 bits.

When less than 32 bytes of data are written to the FIFO buffer, IEBBTD occurs for the remaining number of bytes of data to be transmitted.

The IEBBTD interrupt occurs when transmission data is transferred from the FIFO buffer to the shift register and the number of bytes of data that have not been transmitted reaches the value specified for the IEBBnSLTI1 and IEBBnSLTI0 bits.

For data less than the number of bytes specified for the IEBBnSLTI1 and IEBBnSLTI0 bits, the data is written, and no interrupt occurs even if the number of bytes of data that have not been transmitted reaches the value specified for the IEBBnSLTI1 and IEBBnSLTI0 bits.

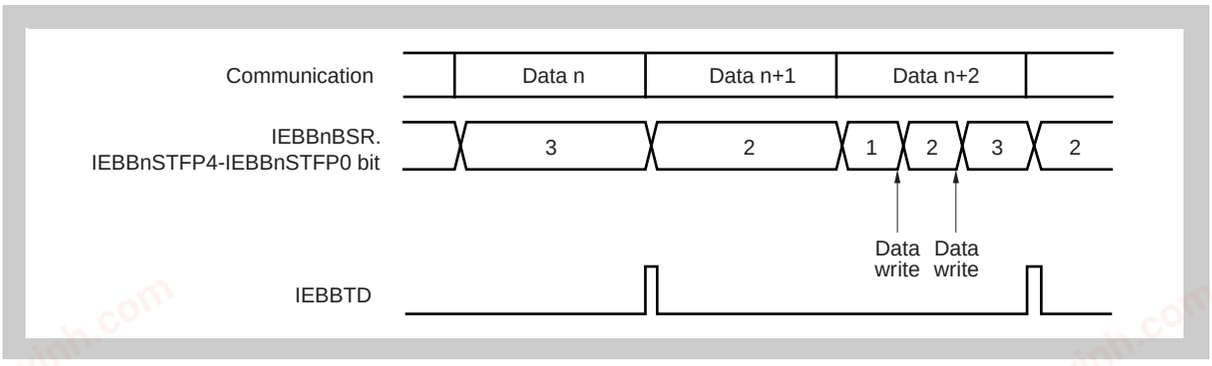


Figure 22-4 Example of operation in communication mode 2: When the IEBBnSLTI1 and IEBBnSLTI0 bits = 01

chanhvinh.com

(d) Arbitration loss count specification bits (IEBBnALC2 to IEBBnALC0): Bits 2 to 0

This is only valid in the FIFO mode. The settings of these bits are invalid in the single mode.
When arbitration is lost and the counter is set to 0H, the IEBBnBCR.IEBBnMSRQ bit is not retained.

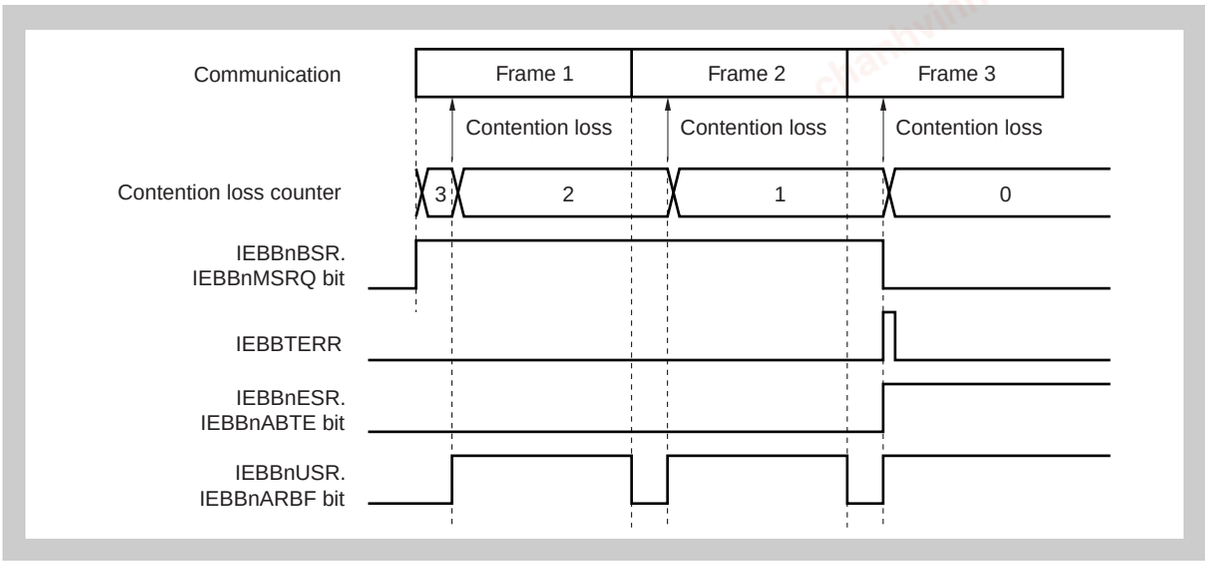


Figure 22-5 Example of operation in the FIFO mode: When the IEBBnALC2 to IEBBnALC0 bits = 011

chanhvinh.com

(15) IEBBnPCR - IEBBn pointer clear register

The IEBBnPCR register is the trigger bit register for clearing the FIFO buffer pointer.

Access This register is write-only, in 8-bit units.

Address <IEBBn_base> + 0038_H

Initial value 00_H

7	6	5	4	3	2	1	0
IEBBn CRPT	IEBBn CTPT	0	0	0	0	0	0
W	W	R	R	R	R	R	R

Table 22-21 IEBBnPCR register contents

Bit position	Bit name	Function
7	IEBBnCRPT	Clear trigger bit for the store pointer and read pointer of the reception FIFO buffer 0: No operation 1: Clear the store pointer and read pointer of the reception FIFO buffer. The bit value can only be changed by setting the bit (to 1). Attempting to clear the bit (to 0) does not change the bit value. When the bit is read, 0 is always returned. Caution During reception or before reading received data, if the IEBBnCRPT bit is set to 1, the received data cannot be read. Except when discarding received data, only write 1 to the IEBBnCRPT bit after receiving and then reading data.
6	IEBBnCTPT	Clear the trigger bit for the write pointer and load pointer of the transmission FIFO buffer 0: No operation 1: Clear the write pointer and load pointer of the transmission FIFO buffer. The bit value can only be changed by setting the bit (to 1). Attempting to clear the bit (to 0) does not change the bit value. When the bit is read, 0 is always returned. Cautions 1. If the IEBBnCTPT bit = 1 while specifying data for the FIFO buffer or during transmission, the specified data cannot be transmitted. Except when discarding transmission data, only write 1 to the IEBBnCTPT bit after transmitting the data written to the FIFO buffer. 2. To discard transmission data, set the IEBBnCTPT bit to 1 during transmission. Because the data is lost in this case, transmission is not performed and an underrun error occurs.

chanhvinh.com

(16) IEBBnBSR - IEBBn buffer status register

The IEBBnBSR register indicates the FIFO buffer status.

Access This register is read only, in 16-bit units.

Address <IEBBn_base> + $003C_H$

Initial value 0000_H

This register is reset when 0 is written to the IEBBnBCR.IEBBnPW bit.

15	14	13	12	11	10	9	8
IEBBn RFLF	IEBBn FOVR	0	IEBBn SRFP4	IEBBn SRFP3	IEBBn SRFP2	IEBBn SRFP1	IEBBn SRFP0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
IEBBn TFLF	IEBBn FOVW	0	IEBBn STFP4	IEBBn STFP3	IEBBn STFP2	IEBBn STFP1	IEBBn STFP0
R	R	R	R	R	R	R	R

Table 22-22 IEBBnBSR register contents (1/2)

Bit position	Bit name	Function
15	IEBBnRFLF	Reception FIFO buffer full status flag 0: There are 31 bytes of data or less that have not been read in the reception FIFO buffer. 1: There are 32 bytes of data that have not been read in the reception FIFO buffer. This flag indicates that there are 32 bytes of data that have not been read in the reception FIFO buffer and that the buffer is full.
14	IEBBnFOVR	This flag indicates whether reception FIFO buffer over-reading has occurred. 0: Reception FIFO buffer over-reading has not occurred. 1: Reception FIFO buffer over-reading has occurred.
12 to 8	IEBBnSRFP [4-0]	This flag indicates the number of data bytes that have not been read remaining in the reception FIFO buffer. The (store pointer - read pointer) value can be read. However, the following values are used when IEBBnSRFP4 to IEBBnSRFP0 = 00000. • When the IEBBnRFLF bit = 0 and the IEBBnSRFP4 to IEBBnSRFP0 bits = 00000 Number of remaining data bytes that have not been read = 0 bytes • When the IEBBnRFLF bit = 1, and the IEBBnSRFP4 to IEBBnSRFP0 bits = 00000 Number of remaining data bytes that have not been read = 32 bytes
7	IEBBnTFLF	Transmission FIFO buffer full status flag 0: There are 31 bytes of data or less that have not been transferred in the transmission FIFO buffer. 1: There are 32 bytes of data that have not been transferred in the transmission FIFO buffer. This flag indicates that there are 32 bytes of data that have not been transferred in the transmission FIFO buffer and that the buffer is full.
6	IEBBnFOVW	Transmission FIFO buffer overwrite flag 0: No transmission FIFO buffer overwrite has occurred. 1: A transmission FIFO buffer overwrite has occurred.

chanhvinh.com

Table 22-22 IEBBnBSR register contents (2/2)

Bit position	Bit name	Function
4 to 0	IEBBnSTFP [4-0]	This flag indicates the number of data bytes that have not been transferred remaining in the transmission FIFO buffer. The (write pointer - load pointer) value can be read. However, the following values are used when the IEBBnSTFP4 to STFPSTFP0 bits = 00000. • When the IEBBnTFLF bit = 0 and the IEBBnSTFP4 to IEBBnSTFP0 bits = 00000 Number of remaining data bytes that have not been transferred = 0 bytes • When the IEBBnTFLF bit = 1 and the IEBBnSTFP4 to IEBBnSTFP0 bits = 00000 Number of remaining data bytes that have not been transferred = 32 bytes

chanhvinh.com

(a) Reception FIFO buffer over-read indicating flag (IEBBnFOVR): Bit 14

- Set/clear condition

  Set:

  – Single mode:

    The IEBBnFOVR bit is not set (to 1).

  – FIFO mode:

    When an over-read (a read of the IEBBnDR register while the store pointer = the read pointer) occurs for the reception FIFO buffer

  Clear:

  – Single mode or FIFO mode:

    By software (The flag is cleared when 1 is written to the IEBBnPCR.IEBBnCRPT bit.)

After reading the reception FIFO buffer, detect whether an over-read occurred by reading the IEBBnFOVR bit. (The data read during the over-read is the last bytes read during multiple operations.)

Even if an over-read is detected, because the data cannot be read again, perform software processing such as requesting retransmission. The IEBBnFOVR bit is cleared by writing 1 to the IEBBnPCR.IEBBnCRPT bit. (If an over-read occurred, the FIFO buffer pointers have already been cleared.)

(b) Transmission FIFO buffer overwrite flag (IEBBnFOVW): Bit 6

- Set/clear condition

  Set:

  – Single mode:

    The IEBBnFOVW bit is not set (to 1).

  – FIFO mode:

    When there are 32 bytes of that have not been transmitted in the transmission FIFO buffer and a 33rd byte of data is written

  Clear:

  – Single mode or FIFO mode:

    By software (The flag is cleared when 1 is written to the IEBBnPCR.IEBBnCTPT bit.)

After writing the required data to the transmission FIFO buffer, read the IEBBnFOVW bit before setting the master request flag. (IEBBnBCR.IEBBnMSRQ) (to 1) to detect whether an overwrite has occurred. If an overwrite is detected, clear the FIFO buffer pointers by writing 1 to the IEBBnPCR.IEBBnCTPT bit and then specify the data again.

chanhvinh.com

(17) IEBBnSSR - IEBBn slave status register

The IEBBnSSR register indicates the communication status of the slave unit.

When a slave status transmission request interrupt is received form the master and the received control data is 0H or 6H, the IEBBnSSR register value is automatically written to the IEBBnDR register, and the slave status is transmitted.

In addition, when transmitting the slave status, because 01H is automatically transmitted as the message length, the IEBBnTDL register does not have to be set up.

Because bits 7 and 6 indicate the highest mode supported by the unit, they are fixed to 10 (which indicates communication mode 2).

Access This register is read-only, in 8-bit units.

Address <IEBBn_base> + 0040_H

Initial value 81_H

The IEBBnSSLF and IEBBnSTLF bits are reset by writing 0 to the IEBBnBCR.IEBBnPW bit.

The IEBBnSRXF and IEBBnSTXF bits are reset when the value of the IEBBnPW bit is overwritten with a different value.

7	6	5	4	3	2	1	0
1	0	0	IEBBn SSLF	0	IEBBn STLF	IEBBn SRXF	IEBBn STXF
R	R	R	R	R	R	R	R

Table 22-23 IEBBnSSR register contents (1/2)

Bit position	Bit name	Function
4	IEBBnSSLF	Slave transmission status flag 0: Slave transmission is stopped. 1: Slave transmission is enabled.
2	IEBBnSTLF	Lock status flag 0: Unlocked 1: Locked
1	IEBBnSRXF	IEBBnDR register or FIFO buffer reception status flag In the single mode 0: Received data has not been stored in the IEBBnDR register. 1: Received data has been stored in the IEBBnDR register. In the FIFO mode 0: Received data has not been stored in the FIFO buffer. 1: Received data has been stored in the FIFO buffer.

chanhvinh.com

Table 22-23 IEBBnSSR register contents (2/2)

Bit position	Bit name	Function
0	IEBBnSTXF	IEBBnDR register or FIFO buffer transmission status flag • When communication is not being performed 1: The flag is always this value. • Master In the single mode (transmission) 0: The data specified for the IEBBnDR register has been transferred to the transmission shift register, and the next transmission data has not been written to the IEBBnDR register. 1: Transmission data remains in the IEBBnDR register. (This is the status up until the contents of the IEBBnDR register are transferred to the transmission shift register.) In the single mode (reception) 1: The flag is always this value. In the FIFO mode (transmission) 0: The number of bytes of data written to the FIFO buffer have been transferred from the FIFO buffer to the transmission shift register, and the next transfer data has not been written to the FIFO buffer. 1: This is the status from when communication starts until data is transferred to the transmission shift register and the number of bytes of data written to the FIFO buffer disappear. In the FIFO mode (reception) 1: The flag is always this value. • Slave In the single mode (transmission) 0: The data specified for the IEBBnDR register has been transferred to the transmission shift register, and the next transmission data has not been written to the IEBBnDR register. 1: This is the status from when communication starts until the first transmission data is transferred from the IEBBnDR register to the transmission shift register. This is also the status after writing data to the IEBBnDR register until the data is transferred to the transmission shift register. In the single mode (reception) 1: The flag is always this value. In the FIFO mode (transmission) 0: The number of bytes of data written to the FIFO buffer have been transferred from the FIFO buffer to the transmission shift register, and the next transfer data has not been written to the FIFO buffer. 1: This is the status from when communication starts until data is transferred to the transmission shift register and the number of bytes of data written to the FIFO buffer disappear. In the FIFO mode (reception) 1: The flag is always this value. • Third party 1: The flag is always this value.

chanhvinh.com

(a) Slave transmission status flag (IEBBnSSLF): Bit 4

The value of the slave transmission enable flag (the IEBBnBCR.IEBBnSTXE bit) is applied as is.

(b) Lock status flag (IEBBnSTLF): Bit 2

The value of the lock status flag (the IEBBnUSR.IEBBnLCKF bit) is applied as is.

(c) IEBBnDR register or FIFO buffer reception status flag (IEBBnSRXF): Bit 1

- Set/clear condition

  Set:

  – Single mode:

    When received data is stored in the IEBBnDR register

  – FIFO mode:

    When received data is stored in the FIFO buffer

  Clear:

  – Single mode:

    When the IEBBnDR register contents are read

  – FIFO mode:

    When all the received data stored in the FIFO buffer has been read

In the single mode, when the IEBBnSRXF bit is set (to 1), the data interrupt IEBBTD occurs.

In the single mode, when IEBBTD occurs on data reception, the IEBBnDR register must be read before the next data is received. During broadcast communication, an overrun error occurs if the IEBBnDR register is not read regardless of whether IEBBTD has occurred. For details, see the description of the IEBBnOVRE bit in 22.3.2 (20) IEBBnESR - IEBBn error status register.

In the FIFO mode, even if the IEBBnSRXF bit is set (to 1), the interrupt signal IEBBTV is not generated. For details about the IEBBTV generation timing, see the description of the IEBBnSLRI1, IEBBnSLRI0 bits in 22.3.2 (14) IEBBnTMS - IEBBn transfer mode setting register.

chanhvinh.com

(d) IEBBnDR register or FIFO buffer transmission status flag (IEBBnSTXF): Bit 0

- Set/clear condition

  Set:

  – When communication finishes

  – Single mode:

    When the IEBBnDR register is written to

  – FIFO mode:

    When transmission data is written to the FIFO buffer

  Clear:

  – Single mode:

    When the contents of the IEBBnDR register are written to the transmission shift register

  – FIFO mode:

    When the number of bytes of data written to the FIFO buffer are transferred from the FIFO buffer to the transmission shift register

In the single mode, when the IEBBnSTXF bit is cleared, the data interrupt IEBBTD occurs.

In the single mode, when IEBBTD occurs on data transmission, the next transmission data must be written to the IEBBnDR register.

Regardless of whether IEBBTD occurs, an underrun error occurs if IEBBnDR is not written to. For details, see the description of the IEBBnUNRE bit in 22.3.2 (20) IEBBnESR - IEBBn error status register.

In the FIFO mode, even if the IEBBnSRXF bit is set (to 1), the interrupt signal IEBBTD is not generated. For details about the IEBBTD generation timing, see the description of the IEBBnSLTI1, IEBBnSLTI0 bits in 22.3.2 (14) IEBBnTMS - IEBBn transfer mode setting register.

chanhvinh.com

(18) IEBBnUSR - IEBBn unit status register

The IEBBnUSR register indicates the unit status.

Access This register is read-only, in 8-bit units.

Address <IEBBn_base> + 0044_H

Initial value 00_H

This register is reset when the value of the IEBBnPW bit is overwritten with a different value.

7	6	5	4	3	2	1	0
0	IEBBn SRQF	IEBBn ARBF	IEBBn ALTF	IEBBn ACKF	IEBBn LCKF	0	0
R	R	R	R	R	R	R	R

Table 22-24 IEBBnUSR register contents

Bit position	Bit name	Function
6	IEBBnSRQF	Slave request flag for the unit 0: There are no slave requests. 1: There is a slave request.
5	IEBBnARBF	Arbitration result flag 0: Arbitration loss did not occur. 1: Arbitration loss occurred.
4	IEBBnALTF	Broadcast communication flag 0: Individual communication status 1: Broadcast communication status
3	IEBBnACKF	Acknowledge transmission flag 0: A NACK signal is transmitted. 1: An ACK signal is transmitted.
2	IEBBnLCKF	Lock status flag 0: Unlocked 1: Locked

chanhvinh.com

(a) Slave request flag for the unit (IEBBnSRQF): Bit 6

- Set/clear condition

  Set:

  When the unit is requested as a slave (if the condition in Table 22-25 is satisfied), this flag is set (to 1) by hardware when the parity bit communication of the slave address field ends[a].

  Clear:

  This flag is cleared (to 0) by hardware when the unit is not requested as a slave (if the condition in Table 22-25 is not satisfied). The timing[a] is the same as that for setting the flag.

a) The bit is updated when the communication of the slave address field parity bit finishes without an error such as a parity error occurring. For example, if the slave address reception parity is incorrect, the IEBBnSRQF bit is not updated and the previous value is retained.

Table 22-25 Slave request conditions (conditions for setting the IEBBnSRQF bit)

Status of unit	Received master address	Communication mode	Received slave address
Not locked	don't care	Individual	IEBBnUAR match
		Broadcast	Group matching
			FFFH match
Locked	Locked master matching	Individual	IEBBnUAR match
		Broadcast	Group matching
			FFFH match

Note IEBBnUAR match: When the reception slave address and unit IEBBnUAR register match

Group match: When the reception-slave-address group address and unit IEBBnUAR-register group address match

FFFH match: When the reception slave address is FFFH

Table 22-26 $\overline{\text{ACK}}$ signal response condition for the slave address field

Status of unit	Received master address	Communication mode	Received slave address
don't care	don't care	Individual	IEBBnUAR match

Note IEBBnUAR match: When the reception slave address and unit IEBBnUAR register match

Caution If a unit other than the locked master communicates with the unit while the unit is locked, the IEBBnSRQF bit is not set but the $\overline{\text{ACK}}$ signal is returned to the slave address field. This is because communication must be continued, even for communication of a unit other than the locked master, if the control data received using the control field is a slave status transmission request.

chanhvinh.com

(b) Arbitration result flag (IEBBnARBF): Bit 5

- Set/clear condition

  Set:
  When the data output by the unit does not match the received data during the arbitration period[a]
  Clear: After each communication frame start bit is transmitted or received

a) In the FIFO mode, the arbitration loss flag is set (to 1) based on the same condition even while the master request flag (IEBBnBCR.IEBBnMSRQ) is retained.

Note that, if arbitration is lost in the FIFO mode, a NACK reception error is detected by returning the NACK signal after control data reception, and software processing by outputting IEBBTERR is required.

In the case of a status transmission request, the NACK signal is not returned because there is an automatic response.

The IEBBnARBF bit is set (to 1) if there is inter-unit-data contention during the arbitration period (the broadcast field and master address field period) and the unit loses arbitration.

Arbitration loss is judged to have occurred when the unit output data does not match the received data.

Because the IEBus controller uses AND logic, the unit that outputs 0 wins arbitration.

In other words, among units that output broadcast data (0) for the broadcast field, the unit that has the smallest master address wins arbitration.

Caution In the single mode, when the start interrupt occurs upon issuing a master request, use the IEBBnARBF bit to check for arbitration loss. To transfer data again in the case of arbitration loss, perform software processing to issue another master request.

In the FIFO mode, another master request is issued even if arbitration is lost the number of times specified by the IEBBnTMS.IEBBnALC0 to IEBBnALC2 bits. If the number of arbitration losses exceeds this setting, an interrupt occurs to indicate an arbitration loss error. To transfer data again, perform software processing to issue another master request.

chanhvinh.com

(c) Broadcast communication flag (IEBBnALTF): Bit 4

Flag indicating whether the unit is performing broadcast communication. The contents of the flag are updated in the broadcast field of each frame.

- Set/clear condition

  Set: When “broadcast” is received by the broadcast field

  Clear: When individual is received by the broadcast field

Caution The broadcast flag is updated regardless of whether IEBus is the communication target.

In the FIFO mode, storage is performed upon header completion if reading the data received during the previous communication has finished (if the IEBBnBSR.IEBBnRFLF bit = 0 and the nBSR.IEBBnSRFP4 to IEBBnSRFP0 bits = 00000). Until reading the received data finishes, the IEBBnALTF register is not updated.

(d) Acknowledge transmission flag (IEBBnACKF): Bit 3

This flag indicates whether the $\overline{\text{ACK}}$ signal was transmitted during the acknowledge bit period of the acknowledge bit field when IEBus is the receiving unit.

- Set/clear condition

  Set:

  When $\overline{\text{ACK}}$ is transmitted upon the completion of the acknowledge bit period for each field

  Clear:

  When NACK is transmitted upon the completion of the acknowledge bit period for each field

Cautions
1. If a communication error occurs and the unit returns to the initial status, no update is performed at the end of the acknowledge bit period for the corresponding field. For example, if the reception parity for the control field is incorrect, because IEBBn changes to the initial status (the communication standby status) after parity reception due to the parity error, a NACK signal is returned by using the control field (more accurately, no $\overline{\text{ACK}}$ signal is returned), but this is not applied to the IEBBnACKF bit, and the previous value is retained.
2. In the single mode, because the occurrence timing (when IEBBTSTA or IEBBTV becomes active) for the start interrupt and status transmission interrupt is when parity bit reception ends, the reading of the IEBBnUSR register by the previously described interrupt in the slave mode during interrupt handler processing might overlap with the changing of the IEBBnACKF bit.

chanhvinh.com

(e) Lock status flag (IEBBnLCKF): Bit 2

A flag that indicates whether the unit is locked.

- Set/clear condition

  Set:

  When an individual communication frame ends, lock-related data (3H, 6H, AH, and BH) is received by using the control field, the communication completion flag (the IEBBnISR.IEBBnETRF bit) is cleared (to 0), and the frame completion flag (the IEBBnISR.IEBBnEFMF bit) is set (to 1).

  Clear:

  When an individual communication frame ends, lock-related data (3H, 6H, AH, and BH) is received by using the control field, and the communication completion flag (the IEBBnISR.IEBBnETRF bit) is set (to 1)

Cautions
1. Locking and unlocking are performed only during individual communication, not during broadcast communication.
2. While the master is locked, communication from a unit other than the master is not generally acknowledged. However, as an exception, if the control data (0H, 4H, or 5H) of a slave status transmission request is received, the communication is acknowledged even if it is not from the locked master. Note that, at this time, only a status request interrupt occurs, not a start interrupt or completion interrupt.

chanhvinh.com

(19) IEBBnISR - IEBBn interrupt status register

The IEBBnISR status register indicates the interrupt source when an IEBBTSTA, IEBBTERR, or IEBBTV interrupt occurs.

Each time the IEBBTSTA, IEBBTERR, and IEBBTV interrupts occur, the IEBBnISR register is read, and the specified interrupt servicing is performed.

Access Only bit 6 can be read or written in 8-bit units.

Bits other than bit 6 are read-only, in 8-bit units.

Address <IEBBn_base> + 0048_H

Initial value 00_H

The IEBBnIEBE bit is reset when 0 is written to the IEBBnBCR.IEBBnPW bit.

Bits other than IEBBnIEBE are reset when the value of the IEBBnPW bit is overwritten with a different value.

Caution Be sure to set bits 1, and 7 to 0.

7	6	5	4	3	2	1	0
0	IEBBn IEBE	IEBBn STRF	IEBBn STSF	IEBBn ETRF	IEBBn EFMF	0	IEBBn FOVE
R	R/W[a]	R	R	R	R	R	R

a) Only the IEBBnIEBE bit can be written. Note that, when writing to the IEBBnIEBE bit, the bit can only be cleared (to 0). Even if 1 is written, the IEBBnIEBE bit is not set (to 1).

Table 22-27 IEBBnISR register contents

Bit position	Bit name	Function
6	IEBBnIEBE	Communication error flag 0: No communication error has occurred. 1: A communication error has occurred.
5	IEBBnSTRF	Start interrupt flag 0: No start interrupt has occurred. 1: A start interrupt has occurred.
4	IEBBnSTSF	Status transmission flag (slave) 0: There is no status transmission request. 1: There is a status transmission request.
3	IEBBnETRF	Communication completion flag 0: Communication of the number of transmission bytes specified by the message length field has not finished. 1: Communication of the number of transmission bytes specified by the message length field has finished.
2	IEBBnEFMF	Frame completion flag 0: The frame (communication of the maximum number of transmission bytes[a]) has not finished. 1: The frame (communication of the maximum number of transmission bytes[a]) has finished.
0	IEBBnFOVE	Frame over error flag 0: No frame over error has occurred. 1: A frame over error has occurred.

a) Communication mode 1: 32 bits, communication mode 2: 128 bits

chanhvinh.com

(a) Communication error flag (IEBBnIEBE): Bit 6

A flag that indicates a communication error has occurred.

- Set/clear condition

  Set:

  – Single mode:

    When a timing error, parity error (except in the data field during individual reception), NACK reception error, underrun error, or overrun error (during broadcast reception) occurs

  – FIFO mode:

    When a timing error, parity error (except in the data field during individual reception), NACK reception error, underrun error, overrun error (during broadcast reception), or arbitration loss error occurs

  Clear:

  – Single mode or FIFO mode:

    By software (The flag is cleared (to 0) when 0 is written to the IEBBnIEBE bit.)

When a communication error occurs, IEBBTERR IEBBTV occur in the single mode, and IEBBTERR occurs in the FIFO mode.

It is possible to determine what caused the error by reading the IEBBnESR and IEBBnISR registers.

chanhvinh.com

(b) Start interrupt flag: (IEBBnSTRF): Bit 5

A flag that indicates the start interrupt.

- Set/clear condition

  Set:

  – Single mode:

  The flag is set (to 1) during master unit operation, regardless of whether arbitration is won or lost.

  The flag is set (to 1) during slave unit operation if there is a slave request (when the IEBBnUSR.IEBBnSRQF bit = 1) from the master (only the locked master when the unit is locked).

  The flag is set upon the completion of the slave address field parity period in all cases.

  – FIFO mode: The IEBBnSTRF bit is not set (to 1).

  Clear:

  – Single mode:

  If the unit is the communication target (during communication with the master unit or slave unit), the flag is cleared (to 0) by hardware when a status transmission interrupt, communication completion interrupt, frame completion interrupt, transmission data write request interrupt, reception data read interrupt, or communication error interrupt occurs.

  – FIFO mode: The IEBBnSTRF bit is normally cleared.

In the single mode, IEBBTSTA and IEBBTV occur when a start interrupt occurs. In the FIFO mode, the start interrupt, IEBBTSTA, and IEBBTV do not occur.

Cautions
1. When a start interrupt occurs, read the IEBBnUSR register to check the slave request flag (IEBBnSRQF) and arbitration result flag (IEBBnARBF) for the unit.
2. If the arbitration result flag (IEBBnARBF) is set (to 1) when a start interrupt occurs after the unit issues a master request, perform software processing to reissue the master request.

chanhvinh.com

(c) Status transmission flag (slave) (IEBBnSTSF): Bit 4

This flag indicates that the master requested transmission of the slave status and lock address (higher 4 bits and lower 8 bits) when the controller was serving as a slave.

- Set/clear condition

  Set:

  – Single mode:

  When the unit is not locked, there is a slave request from any master, and 0H or 6H is received by using the control field

  When the unit is locked, there is a slave request from the locked master, and 0H, 4H, 5H, or 6H is received by using the control field, or when 0H, 4H, or 5H is received from a unit other than the locked master by using the control field

  The flag is set upon the completion of the control field parity period in all cases.

  For details, see Table 22-28.

  – FIFO mode: The IEBBnSTSF bit is not set (to 1).

  Clear:

  – Single mode:

  If the unit is the communication target (during communication with the master unit or slave unit), the flag is cleared (to 0) by hardware when a start interrupt, communication completion interrupt, frame completion interrupt, transmission data write request interrupt, reception data read interrupt, or communication error interrupt occurs.

  – FIFO mode: The IEBBnSTSF bit is normally cleared.

In the single mode, IEBBTSTA and IEBBTV occur when there is a status transmission request. In the FIFO mode, no interrupt signal is generated even if there is a status transmission request.

Caution Even if the slave transmission enable flag (the IEBBnBCR.IEBBnSTXE bit) is set to the prohibited value (0), the IEBBnSTSF bit is set (to 1).

chanhvinh.com

Table 22-28 Conditions for setting the status transmission request flag (slave)

Various statuses					Value received using the control field				
equa	lockf	eqpa	IEBBn STXE	IEBBn SRXE	0H	3H, 7H	4H, 5H	6H	AH, BH, EH, FH
1	0	0	Any	Any	Set	Not set	Not set	Set	Not set
1	0	1	Any	Any	Set	Not set	Not set	Set	Not set
1	1	0	Any	Any	Set	Not set	Set	Not set	Not set
1	1	1	Any	Any	Set	Not set	Set	Set	Not set

Note equa: Unit match (during individual communication, IEBBnUAR register match

lockf: Whether there is a lock

eqpa: Lock master match

IEBBnSTXE: Slave transmission enable flag (IEBBnBCR register bit 4)

IEBBnSRXE: Slave reception enable flag (IEBBnBCR register bit 3)

If a slave status transmission request interrupt occurs in the single mode, read the IEBBnCDR register to check the received control data contents, and then write the required slave status information to the IEBBnDR register.

The received control data and data written to the IEBBnDR register are shown below.

Table 22-29 Received control data and data written to the IEBBnDR register

Received control data	Function	Data written to the IEBBnDR register
0H, 6H	Slave status transmission	Value read from the IEBBnSSR register
4H	Transmission of the lower 8 bits of the lock address	Lower 8 bits of the IEBBnPAR register
5H	Transmission of the higher 4 bits of the lock address	Higher 8 bits of the IEBBnPAR register

Caution After a slave status transmission request interrupt occurs, be sure to write the appropriate data to the IEBBnDR register before the completion of the message length field.

chanhvinh.com

(d) Communication completion flag (IEBBnETRF): Bit 3

A flag that indicates whether communication ends after the number of bytes set in the message length field have been transferred.

- Set/clear condition

  Set:

  – Single mode:

    When the unit is the communication target (during communication with the master unit or slave unit) and the value of the IEBBnSCR register becomes 0 at the end of the data field acknowledge period

  Clear:

  – Single mode:

    The flag is cleared (to 0) by hardware when a start interrupt, status transmission interrupt, frame completion interrupt (when a communication completion interrupt does not occur), transmission data write request interrupt, reception data read interrupt, or communication error interrupt occurs.

In the single mode, IEBBTSTA and IEBBTV occur when the communication completion flag is set (to 1).

In the FIFO mode, the IEBBTSTA interrupt occurs at the same timing as in the single mode.

In the FIFO mode, reading the IEBBnETRF bit is prohibited. If the bit is read, the returned value is undefined.

In the FIFO mode, after IEBBTSTA occurs, transmission and reception can be controlled by performing the software processing below.

- When reception ends

1. Detecting the completion of communication based on the occurrence of the IEBBTSTA interrupt
2. Checking whether communication or a frame has finished by using the IEBBnFSR.IEBBnRTRF bit
3. Checking the number of received data bytes by using the IEBBnBSR.IEBBnSRFP4 to IEBBnSRFP0 bits (the number of bytes that have not been read)
4. Reading the received data from the IEBBnDR register (the reception FIFO buffer)

- When transmission ends

1. Detecting the completion of communication based on the occurrence of the IEBBTSTA interrupt
2. Checking whether communication or a frame has finished by using the IEBBnFSR.IEBBnTTRF bit
3. Checking the number of transmission data bytes by using the IEBBnBSR.IEBBnSTFP4 to IEBBnSTFP0 bits (the number of bytes that have not been transmitted)
4. Proceeding to retransmission processing if there is data that has not been transmitted

chanhvinh.com

(e) Frame completion flag (IEBBnEFMF): Bit 2

This flag indicates whether communication ends after the maximum number of bytes (communication mode 1: 32 bytes, communication mode 2: 128 bytes) have been transferred.

- Set/clear condition

  Set:

  – Single mode:

    When the unit is the communication target (during communication with the master unit or slave unit) and the value of the IEBBnCCR register becomes 0 at the end of the data field acknowledge period

  Clear:

  – Single mode:

    The flag is cleared (to 0) by hardware when a start interrupt, status transmission interrupt, communication completion interrupt (when a frame completion interrupt does not occur), transmission data write request interrupt, reception data read interrupt, or communication error interrupt occurs.

In the single mode, IEBBTSTA and IEBBTV occur when the frame completion flag is set (to 1).

Cautions
1. In the single mode, if the IEBBnSCR and IEBBnCCR registers are both cleared to 00H at the end of the data field acknowledge period, the IEBBnETRF and IEBBnEFMF bits are set (to 1) at the same time.
2. In the single mode, if the last data field is a NACK signal when the maximum number of bytes that can be transmitted is reached during retransmission, the IEBBnEFMF and IEBBnIEBE bits (NACK reception error) are set (to 1) at the same time.

In the FIFO mode, the IEBBTSTA interrupt occurs at the same timing as in the single mode.

In the FIFO mode, reading the IEBBnEFMF bit is prohibited. If the bit is read, the returned value is undefined.

After IEBBTSTA occurs, transmission and reception can be controlled by performing the software processing below.

- When reception ends

1. Detecting the completion of communication based on the occurrence of the IEBBTSTA interrupt
2. Checking whether communication or a frame has finished by using the IEBBnFSR.IEBBnRTRF bit
3. Checking the number of received data bytes by using the IEBBnBSR.IEBBnSRFP4 to IEBBnSRFP0 bits (the number of bytes that have not been read)
4. Reading the received data from the IEBBnDR register (the reception FIFO buffer)

chanhvinh.com

- When transmission ends

1. Detecting the completion of communication based on the occurrence of the IEBBTSTA interrupt
2. Checking whether communication or a frame has finished by using the IEBBnFSR.IEBBnTTRF bit
3. Checking the number of transmission data bytes by using the IEBBnBSR.IEBBnSTFP4 to IEBBnSTFP0 bits (the number of bytes that have not been transmitted)
4. Proceeding to retransmission processing if there is data that has not been transmitted

(f) Frame over error flag (IEBBnFOVE): Bit 0

This flag indicates that a frame over error has occurred.

- Set/clear condition

  Set:

  – Single mode: The IEBBnFOVE bit is not set (to 1).

  – FIFO mode:

    If the next broadcast communication is received as a slave unit before reading the data received during the previous communication finishes, the flag is set (to 1) after the first data field parity period finishes.

  Clear:

  – Single mode or FIFO mode:

    By software

    (The flag is cleared (to 0) when 1 is written to the IEBBnSTC1.IEBBnCLFF bit.)

In the single mode, the IEBBnFOVE bit is not set (to 1), and no interrupt requests occur.

In the FIFO mode, IEBBTERR occurs when the IEBBnFOVE bit is set (to 1).

Even if a frame over error occurs for the current frame, the data received during the previous frame is valid.

chanhvinh.com

(20) IEBBnESR - IEBBn error status register

The IEBBnESR register is used to indicate the cause when an IEBus controller communication error interrupt occurs. Each bit of the IEBBnESR register is set (to 1) as soon as the communication error flag of the IEBBnISR register (IEBBnIEBE) is set (to 1). The cause of a communication error, if any, can be identified by checking the contents of the IEBBnESR register. (If the IEBBnISR.IEBBnIEBE bit is already set to 1, only the bits of the IEBBnESR register are set (to 1).)

Note that the bits can only be set (to 1) by hardware and are cleared by writing 1 to the IEBBnSTC0 register.

When 1 is written to the IEBBnSTC0 register, if there is contention with the hardware trying to specify (1), the hardware is prioritized.

Note that writing to the IEBBnESR register is invalid.

Access This register is read-only, in 8-bit units.

Address <IEBBn_base> + $004C_H$

Initial value 00_H

This register is reset when the value of the IEBBnBCR.IEBBnPW bit is overwritten with a different value.

Caution When a communication error occurs, IEBBn returns to the initial status and prepares for the next communication. Regardless of whether an error occurs, if the next communication is started without handling errors, any error flags that have been set remain set. (If the system returns to the initial status due to a timing error, and a parity error is received during the next communication, both the timing error and parity error bits of the IEBBnESR register are set to 1.)
Therefore, handle any errors that occur before the next communication starts.

7	6	5	4	3	2	1	0
IEBBn TIME	IEBBn PARE	IEBBn NACE	IEBBn UNRE	IEBBn OVRE	0	IEBBn ABTE	IEBBn TRDE
R	R	R	R	R	R	R	R

Table 22-30 IEBBnESR register contents (1/2)

Bit position	Bit name	Function
7	IEBBnTIME	Timing error occurrence flag 0: No timing error has occurred. 1: A timing error has occurred.
6	IEBBnPARE	Parity error occurrence flag 0: No parity error has occurred. 1: A parity error has occurred.
5	IEBBnNACE	NACK reception error flag 0: No NACK reception error has occurred. 1: A NACK reception error has occurred.
4	IEBBnUNRE	Underrun error occurrence flag 0: No underrun error has occurred. 1: An underrun error has occurred.

chanhvinh.com

Table 22-30 IEBBnESR register contents (2/2)

Bit position	Bit name	Function
3	IEBBnOVRE	Overrun error occurrence flag 0: No overrun error has occurred. 1: An overrun error has occurred.
1	IEBBnABTE	Arbitration loss error occurrence flag 0: The number of arbitration losses specified for the arbitration loss count setting bits (IEBBnTMS.IEBBnALC2 to IEBBnALC0) have not occurred. 1: The number of arbitration losses specified for the arbitration loss count setting bits (IEBBnTMS.IEBBnALC2 to IEBBnALC0) have occurred.
0	IEBBnTRDE	Inter-third-party communication error occurrence flag 0: An error occurred during communication targeting the unit. 1: An error occurred during inter-third-party communication.

chanhvinh.com

(a) Timing error occurrence flag (IEBBnTIME): Bit 7

- Set condition

  Set: This flag is set (to 1) if a timing error occurs.

A timing error occurs if the high-/low-level width of the communication bit is not the defined value.

The defined value of the high- and low-level width is set to the bit processing block and monitored by the internal timer.

(b) Parity error occurrence flag (IEBBnPARE): Bit 6

- Set condition

  Set: This flag is set (to 1) if a parity error occurs.

When the unit is the reception unit (including while communication between others is being monitored), a parity error occurs when the parity data generated by the data received using the master address field, slave address field, control data field, or message length field does not match the received parity data.

However, when there is a data field mismatch, a NACK signal is returned and a data retransmission request is issued during individual communication, but a parity error occurs during broadcast communication.

Note During the above parity period, if the parity data received on the transmission side is inverted for some reason, a timing error occurs and communication ends.

Table 22-31 Operation when the parity data does not match

Field	Communication mode	Operation when the parity data does not match
Master address field	Individual/broadcast	A parity error occurs.
Slave address field	Individual/broadcast	A parity error occurs.
Control data field	Individual/broadcast	A parity error occurs.
Message length field	Individual/broadcast	A parity error occurs.
Data field	Individual	A NACK signal is returned to request retransmission.
	Broadcast	A parity error occurs.

chanhvinh.com

(c) NACK reception error flag (IEBBnNACE): Bit 5

- Set condition

  Set: This flag is set (to 1) if a NACK reception error occurs.

A NACK reception error occurs if a NACK signal is received during the acknowledge bit period of the slave address field, control data field, or message length field during individual communication, regardless of whether the controller is operating as the master or slave.

When a NACK signal is received for the data field, no NACK reception error generally occurs because this reception signals a data retransmission request. However, if the last data field is a NACK signal, a NACK reception error does occur.

During reception, a NACK reception error is judged to have occurred if an output NACK signal is received for the last data of the slave address field, control data field, message length field, or data field.

Note that, during broadcast communication, no NACK reception errors occur because ACK/NACK signal judgment is not performed.

No NACK reception errors occur during inter-third-party communication because only timing/parity errors are detected as errors. However, for the slave address field, NACK reception error judgment is performed because communication is participated in as a slave.

Table 22-32 NACK reception error judgment period

Communication mode		Slave address field	Control data field Data field	Message length field	Data field (last)
Individual communication	Master transmission	Occurs	Occurs	Occurs	Occurs
	Master reception	—	—	Occurs	Occurs
	Slave transmission	—	—	Occurs	Occurs
	Slave reception	Occurs	Occurs	Occurs	Occurs
	Inter-third-party communication	Occurs	Does not occur	Does not occur	Does not occur
Broadcast communication	All	Does not occur	Does not occur	Does not occur	Does not occur

chanhvinh.com

(d) Underrun error occurrence flag (IEBBnUNRE): Bit 4

- Set condition

  Set:

  – Single mode:

  During data transmission using the data field, an underrun error occurs if writing the next data to be transmitted to the IEBBnDR register does not finish before the end of the data-field acknowledge bit period following the occurrence of IEBBTD, and this flag is set (to 1). However, if the NACK signal is received during the acknowledge bit period, no underrun error occurs because retransmission is performed.

  During inter-third-party communication, underrun errors do not occur because only timing/parity errors are detected as errors.

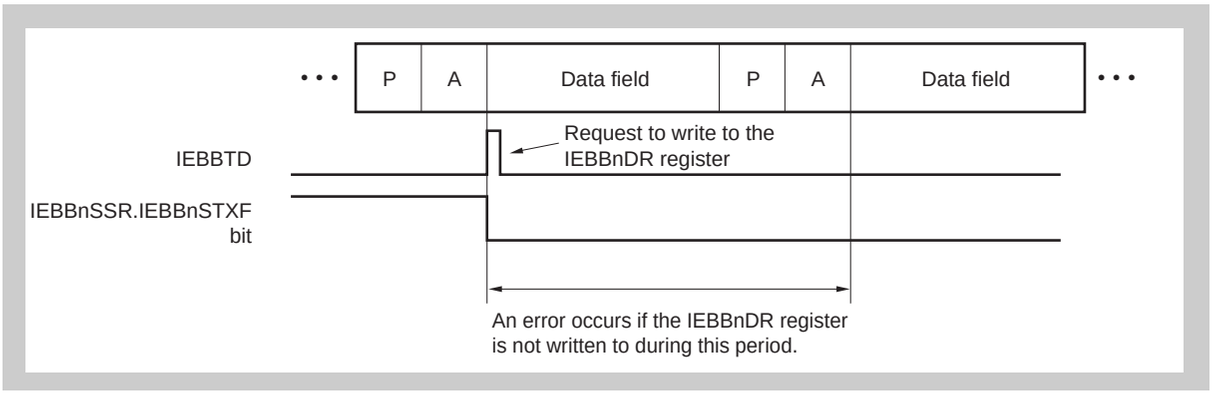


Figure 22-6 Underrun error occurrence timing

– FIFO mode:

Before data of the message length specified by the IEBBnTDL register is transmitted, if there are 0 data items to be transmitted remaining in the transmission FIFO buffer (if the IEBBnSSR.IEBBnSTXF bit = 0), an underrun error occurs if writing the next data to be transmitted to the IEBBnDR register does not finish before the end of the next data-field acknowledge bit period, and this flag is set (to 1). However, if the NACK signal is received during the acknowledge bit period, no underrun error occurs because retransmission is performed.

During inter-third-party communication, underrun errors do not occur because only timing/parity errors are detected as errors.

chanhvinh.com

(e) Overrun error occurrence flag (IEBBnOVRE): Bit 3

- Set condition

  Set:

  – Single mode:

  When the data field is used to receive data during broadcast communication, an overrun error occurs if reading the IEBBnDR register does not finish between when IEBBTD occurs and when the parity period of the data field finishes, and this flag is set (to 1).

  During individual communication, data retransmission is requested by returning a NACK signal without an error, and returning the NACK signal continues until the IEBBnDR register is read. (However, the frame ends when the maximum number of bytes that can be transferred is reached.)

  During inter-third-party communication, overrun errors do not occur because only timing/parity errors are detected as errors.

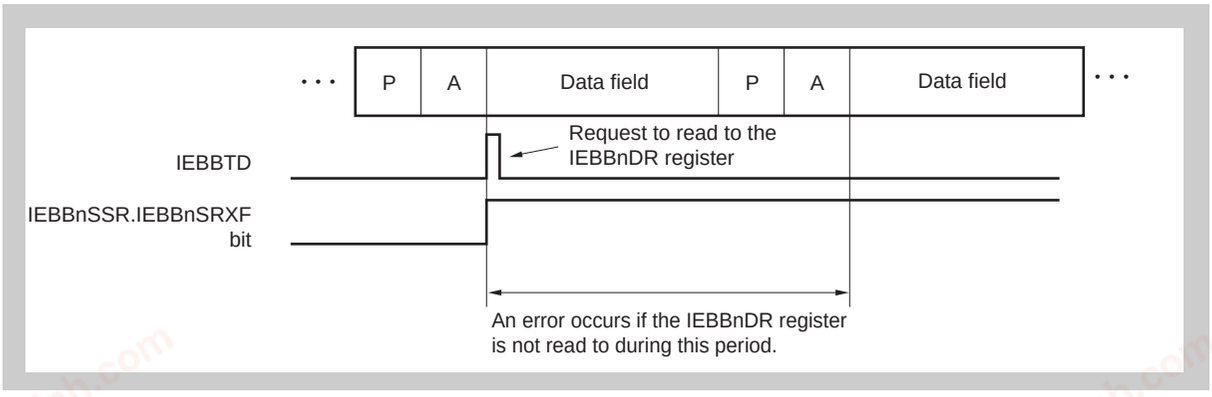


Figure 22-7 Overrun error occurrence timing

– FIFO mode:

During broadcast communication in communication mode 2, if there are 32 bytes of unread data in the reception FIFO buffer, an overrun error occurs if a 33rd byte of data is received, and this flag is set (to 1).

During individual communication, data retransmission is requested by returning a NACK signal without an error, and returning the NACK signal continues until the IEBBnDR register is read. (However, the frame ends when the maximum number of bytes that can be transferred is reached.)

If an overrun error occurs, do not read data by servicing interrupts.

If an interrupt occurs while reading data, it might no longer be possible to read the correct data.

chanhvinh.com

(f) Arbitration loss error occurrence flag (IEBBnABTE): Bit 1

- Set condition

  Set:

  If arbitration losses occur the number of times specified for the IEBBnTMS.IEBBnALC2 to IEBBnALC0 bits, this flag is set (to 1) when the last arbitration loss error occurs.

  For details about arbitration loss, see the description of the IEBBnARBF bit in 22.3.2 (18) IEBBnUSR - IEBBn unit status register.

(g) Inter-third-party communication error occurrence flag (IEBBnTRDE): Bit 0

- Set condition

  Set:

  If an inter-third-party communication error occurs at the same time as a timing error or parity error that occurs during communication that is not related to the unit (inter-third-party communication), this flag is set (to 1) at the same time as the IEBBnTIME or IEBBnPARE bit.

Caution If an error occurs before the inter-third-party communication starts even when the slave address field does not match that of the unit (for example, if the NACK signal is received when the received address does not match that of the unit in the slave address field (if the IEBBnNACE bit is set (to 1))), the IEBBnTRDE bit is not set (to 1).

Note Communication between third parties may take place in the following two cases.

1. If the address received in the slave address field does not match that of the unit (during individual communication: matching with the IEBBnUAR register, during broadcast communication: matching with the group or FFFH) and communication continues after the ACK signal has been received, the unit monitors that communication.
2. If the unit cannot respond to the received control data in the control field during broadcast communication and if communication continues, the unit monitors that communication. For example, this happens when the unit receives the control data FH from the master during broadcast communication but the slave reception enable flag of the unit is disabled (IEBBnBCR.IEBBnSRXE bit = 0). (During individual communication, the NACK signal is returned and communication ends.)

chanhvinh.com

(21) IEBBnFSR - IEBBn field status register

The IEBBnFSR register is used to store the field status state of the IEBus controller when various interrupts (IEBBTD, IEBBTSTA, IEBBTERR, and IEBBTV) occur.

Access This register is read-only, in 8-bit units.

Address <IEBBn_base> + 0050_H

Initial value 00_H

The IEBBnSSFS1 and IEBBnSSFS0 bits are reset when the IEBBnBCR.IEBBnPW bit is overwritten with a different value.

Cautions

1. If a different interrupt occurs before the IEBBnFSR register is read, the status information used at the time of the previous interrupt is overwritten with the status information used at the time of the new interrupt.
2. If an interrupt occurs during communication between third parties (during the reception of communication between other units), the IEBBnSSFS1 and IEBBnSSFS0 bits are cleared to 00. However, because the only interrupts that occur during communication between third parties are interrupts caused by errors, an inter-third-party communication error can be judged to have occurred by reading the inter-third-party communication error occurrence flag (the IEBBnTRDE bit) of the IEBBnESR register.
3. Even if the field status signal (an internal signal) changes, the IEBBnSSFS1 and IEBBnSSFS0 bits retain their previous values until an interrupt occurs.

7	6	5	4	3	2	1	0
IEBBn RTRF	IEBBn TTRF	0	0	0	0	IEBBn SSFS1	IEBBn SSFS0
R	R	R	R	R	R	R	R

Table 22-33 IEBBnFSR register contents

Bit position	Bit name	Function
7	IEBBnRTRF	Reception communication completion flag 0: Communication did not finish during reception. 1: Communication finished during reception.
6	IEBBnTTRF	Transmission communication completion flag 0: Communication did not finish during transmission. 1: Communication finished during transmission.
1, 0	IEBBnSSFS [1, 0]	For details about the IEBBnSSFS1 and IEBBnSSFS0 bits, see Table 22-34.

chanhvinh.com

Table 22-34 Field status

Field status	Description		
	Master/slave	Field	Transmission/ reception
Slave reception status IEBBnSSFS1 and IEBBnSSFS0 bits = 00 (IEBBnFSR register = 00H)	Slave operation	Start bit	Reception
		Master address field	
		Slave address field	
		Control data field	
		Message length field	
		Data field	
Slave transmission status IEBBnSSFS1 and IEBBnSSFS0 bits = 01 (IEBBnFSR register = 01H)	Slave operation	Message length field	Transmission
		Data field	
Master reception status IEBBnSSFS1 and IEBBnSSFS0 bits = 10 (IEBBnFSR register = 02H)	Master operation	Message length field	Reception
		Data field	
Master transmission status IEBBnSSFS1 and IEBBnSSFS0 bits = 11 (IEBBnFSR register = 03H)	Master operation	Start bit	Transmission
		Master address field	
		Slave address field	
		Control data field	
		Message length field	
		Data field	

chanhvinh.com

(a) Reception communication completion flag (IEBBnRTRF): Bit 7

This flag indicates that communication equivalent to the number of bytes specified by the message length has finished during reception.

- Set/clear condition

  Set:

  – Single mode: The IEBBnRTRF bit is not set (to 1).

  – FIFO mode:

    During reception, the flag is set (to 1) when the IEBBnISR.IEBBnETRF bit is set (to 1).

  Clear:

  – Single mode: The IEBBnRTRF bit always has the clear status.

  – FIFO mode:

    During reception, the flag is cleared (to 0) when the IEBBnISR.IEBBnEFMF bit is set (to 1).

Cautions
1. In the FIFO mode, if the set and clear conditions are both satisfied, setting the flag is prioritized.
2. The IEBBnRTRF bit is not cleared (to 0) by writing 1 to the IEBBnPCR.IEBBnCRPT bit.

chanhvinh.com

(b) Transmission communication completion flag (IEBBnTTRF): Bit 6

This flag indicates that communication equivalent to the number of bytes specified by the message length has finished during transmission.

- Set/clear condition

  Set:

  – Single mode: The IEBBnTTRF bit is not set (to 1).

  – FIFO mode:

    During transmission, the flag is set (to 1) when the IEBBnISR.IEBBnETRF bit is set (to 1).

  Clear:

  – Single mode: The IEBBnTTRF bit always has the clear status.

  – FIFO mode:

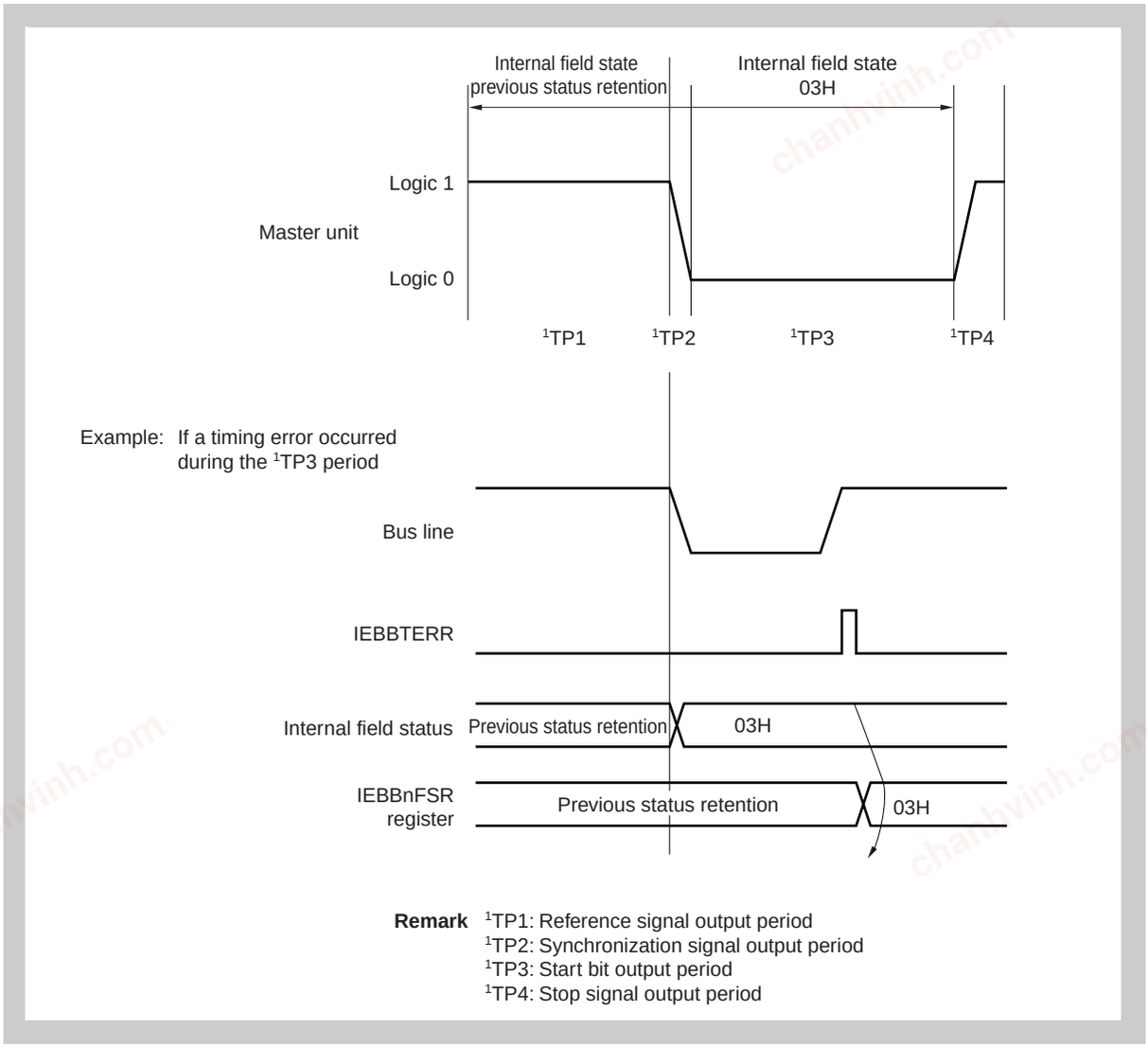
    During transmission, the flag is cleared (to 0) when the IEBBnISR.IEBBnEFMF bit is set (to 1).

Cautions
1. In the FIFO mode, if the set and clear conditions are both satisfied, setting the flag is prioritized.
2. The IEBBnTTRF bit is not cleared (to 0) by writing 1 to the IEBBnPCR.IEBBnCTPT bit.

chanhvinh.com

(c) Field status flags (IEBBnSSFS1 and IEBBnSSFS0): Bits 1 and 0

These flags store the state of the IEBus controller field status when various interrupts (IEBBTD, IEBBTSTA, IEBBTERR, and IEBBTV) occur.



Remark 1TP1: Reference signal output period
1TP2: Synchronization signal output period
1TP3: Start bit output period
1TP4: Stop signal output period

Figure 22-8 Start bit field status for the master (internal signal)

When the start bit shown in Figure 22-8 is output for the master, the previous field status value is retained until 1TP1. At point 1TP2 and after, the field status value is 03H. If a timing error occurs at point 1TP3 and IEBBTERR is output, 03H is stored in the IEBBnFSR register.

Because IEBBTERR does not occur if communication is performed normally, the field status value is not stored in the IEBBnFSR register, and the IEBBnFSR register retains the previous value at and after point 1TP2.

chanhvinh.com

(22) IEBBnSCR - IEBBn success count register

The IEBBnSCR register indicates the number of remaining communication bytes.

The value specified by the IEBBnDLR register is stored in the IEBBnSCR register after the message length field processing finishes, and the count value of the counter to be decremented according to the data field $\overline{ACK}$ signal is read.

In other words, because the number of successfully communicated bytes is subtracted from the number of data bytes to be communicated, the IEBBnSCR register indicates the remaining number of bytes to be communicated.

Note that the communication completion flag (the IEBBnISR.IEBBnETRF bit) is set (to 1) when the count value reaches 00H.

The data in the IEBBnSCR register is updated when the $\overline{ACK}$ signal is received at the end of the message length field parity period or data field acknowledge bit period.

Access This register is read-only, in 8-bit units.

Address <IEBBn_base> + 0054_H

Initial value 01_H

This register is reset when the value of the IEBBnBCR.IEBBnPW bit is overwritten with a different value.

Caution When 00H is read from the IEBBnSCR register, it is not possible to judge whether the remaining number of communication data bytes is 0 (indicating communication completion) or 256. Therefore, the communication completion flag (the IEBBnISR.IEBBnETRF bit) must also be used with this register to make this judgment.

7	6	5	4	3	2	1	0
R	R	R	R	R	R	R	R

Table 22-35 IEBBnSCR register contents

Bit								Setting	Number of remaining communication data bytes
7	6	5	4	3	2	1	0		
0	0	0	0	0	0	0	1	01H	1 byte
0	0	0	0	0	0	1	0	02H	2 bytes
...	...	...	...	...	...	...	...	...	...
0	0	0	1	0	1	0	0	20H	32 bytes
...	...	...	...	...	...	...	...	...	...
1	1	1	1	1	1	1	1	FFH	255 bytes
0	0	0	0	0	0	0	0	00H	0 bytes (communication completion) or 256 bytes

chanhvinh.com

(23) IEBBnCCR - IEBBn communication count register

The IEBBnCCR register indicates the number of bytes remaining from the communication byte number specified by the communication mode.

This register indicates the number of transfer bytes.

The maximum number of transmitted bytes per frame defined in each mode (communication mode 1: 32 bytes, communication mode 2: 128 bytes) is preset to this register. The count value of the counter that is decremented during the acknowledge bit period of the data field regardless of the $\overline{\text{ACK}}$/NACK signal is read from this register. In contrast with the IEBBnSCR register, which is decremented when there is normal communication (the $\overline{\text{ACK}}$ signal), the IEBBnCCR register is decremented when one byte is communicated, regardless of the $\overline{\text{ACK}}$/NACK signal. Note that the frame completion flag (the IEBBnISR.IEBBnEFMF bit) is set (to 1) when the counter reaches 00H.

The preset value of the maximum number of transmitted bytes per frame is 20H (32 bytes) in communication mode 1 and 80H (128 bytes) in communication mode 2.

Updating of data proceeds independently of parity at the end of the ACK interval for the data field on completion of transmission (or reception) of the start bit for the preset maximum number of bytes.

Access This register is read-only, in 8-bit units.

Address <IEBBn_base> + 0058_H

Initial value 20_H

This register is reset when the value of the IEBBnBCR.IEBBnPW bit is overwritten with a different value.

Caution The value of the IEBBnCCR register is not updated by writing to the IEBBnPSR.IEBBnCMD bit.

7	6	5	4	3	2	1	0
R	R	R	R	R	R	R	R

chanhvinh.com

(24) IEBBnSTC0 - IEBBn status clear register 0

The IEBBnSTC0 register is used to clear the IEBBnESR register.

Access This register is write-only, in 8-bit units.

Address <IEBBn_base> + $005C_H$

Initial value 00_H

7	6	5	4	3	2	1	0
IEBBn CLTM	IEBBn CLPA	IEBBn CLNC	IEBBn CLUR	IEBBn CLOV	0	IEBBn CLAB	IEBBn CLTR
W	W	W	W	W	R	W	W

Table 22-36 IEBBnSTC0 register contents

Bit position	Bit name	Function
7	IEBBnCLTM	This bit is used to clear the timing error flag (IEBBnESR.IEBBnTIME). 0: No operation 1: Clear the IEBBnTIME bit. Writing 1 is valid, and writing 0 does not change the internal status. When the bit is read, 0 is always returned.
6	IEBBnCLPA	This bit is used to clear the parity error flag (IEBBnESR.IEBBnPARE). 0: No operation 1: Clear the IEBBnPARE bit. Writing 1 is valid, and writing 0 does not change the internal status. When the bit is read, 0 is always returned.
5	IEBBnCLNC	This bit is used to clear the NACK reception error flag (IEBBnESR.IEBBnNACE). 0: No operation 1: Clear the IEBBnNACE bit. Writing 1 is valid, and writing 0 does not change the internal status. When the bit is read, 0 is always returned.
4	IEBBnCLUR	This bit is used to clear the underrun error flag (IEBBnESR.IEBBnUNRE). 0: No operation 1: Clear the IEBBnUNRE bit. Writing 1 is valid, and writing 0 does not change the internal status. When the bit is read, 0 is always returned.
3	IEBBnCLOV	This bit is used to clear the overrun error flag (IEBBnESR.IEBBnOVRE). 0: No operation 1: Clear the IEBBnOVRE bit. Writing 1 is valid, and writing 0 does not change the internal status. When the bit is read, 0 is always returned.
1	IEBBnCLAB	This bit is used to clear the arbitration loss error flag (IEBBnESR.IEBBnABTE). 0: No operation 1: Clear the IEBBnABTE bit. Writing 1 is valid, and writing 0 does not change the internal status. When the bit is read, 0 is always returned.
0	IEBBnCLTR	This bit is used to clear the inter-third-party communication error flag (IEBBnESR.IEBBnTRDE). 0: No operation 1: Clear the IEBBnTRDE bit. Writing 1 is valid, and writing 0 does not change the internal status. When the bit is read, 0 is always returned.

chanhvinh.com

(25) IEBBnSTC1 - IEBBn status clear register 1

The IEBBnSTC1 register is used to clear the IEBBnISR.IEBBnFOVE bit.

Access This register is write-only, in 8-bit units.

Address <IEBBn_base> + 0060_H

Initial value 00_H

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	IEBBn CLFF
R	R	R	R	R	R	R	W

Table 22-37 IEBBnSTC1 register contents

Bit position	Bit name	Function
0	IEBBnCLFF	This bit is used to clear the frame over error flag (IEBBnISR.IEBBnFOVE). 0: No operation 1: Clear the IEBBnFOVE bit. Writing 1 is valid, and writing 0 does not change the internal status. When the bit is read, 0 is always returned.

chanhvinh.com

(26) IEBBnDR - IEBBn data register

The IEBBnDR register is used to set up the communication data. Specify the communication data (8 bits) for bits 7 to 0.

Notes
1. The IEBBnDR register consists of a write register and a read register. Therefore, data written to this register cannot be read as is. The data received during IEBus communication can be read.
2. In the FIFO mode, the data in the FIFO buffer can be transferred by continuously accessing the IEBBnDR register. (See 22.5.1 (1) Transmission FIFO buffer for details about using the transmission unit or 22.5.1 (2) Reception FIFO buffer for details about using the reception unit.)

(a) For the transmission unit

If the unit is the transmission unit (during master or slave transmission), the bits in the data field are transmitted as data bits starting with the highest bits when writing to the IEBBnDR register.

Specify the first byte of transmission data before starting communication (IEBBnBCR.IEBBnMSRQ bit = 0).

Even in the FIFO mode, write at least one byte of transmission data to the FIFO buffer before starting communication (IEBBnMSRQ bit = 0).

In the FIFO mode, the contents of the FIFO buffer are not reset in sync with the IEBBnBCR.IEBBnPW bit. Clearing the pointer value (to 0) eliminates the remaining data visible to the user. The stored data becomes undefined.

During transmission (master or slave transmission), if a data interrupt (IEBBTD) occurs, the next transmission data is written to the IEBBnDR register.

In the single mode, if a status transmission interrupt (IEBBTSTA or IEBBTV) occurs, the status data is written to the IEBBnDR register according to the control data.

(b) For the reception unit

The 1 byte of data received using the data field is read from the IEBBnDR register if the unit is the reception unit (master or slave reception). Storage is performed at the end of the data field parity period if the parity value is normal. The read value is reset by clearing the IEBBnPW bit to 0.

During reception (master or slave reception), if a data interrupt (IEBBTD) occurs, received data is read from the IEBBnDR register.

Access This register can be read or written in 8-bit units.

Address <IEBBn_base> + 0064_H

Initial value 00_H

The read value is reset when the IEBBnBCR.IEBBnPW bit is overwritten.

chanhvinh.com

Cautions

1. If writing to the IEBBnDR register is not in time for transmission, an underrun error occurs and communication ends.
2. Write to the status data register IEBBnDR after a status transmission interrupt occurs and before the end of the message length field.
3. In the single mode, if the IEBBnDR register is not read before the next reception, the communication differs in the case of individual versus broadcast communication.
   - For individual communication, a NACK signal is returned for the field, and the master is requested to transmit the same data. Received data is not stored in the IEBBnDR register. If the NACK signal is returned again before the IEBBnDR register is read and the register has still not been read when the maximum number of transferable bytes is reached, frame completion (IEBBTSTA, IEBBTV) and a NACK reception error (IEBBTERR, IEBBTV) occur at the same time.
   - During broadcast communication, an overrun error occurs and communication ends. Received data is not stored in the IEBBnDR register. The overrun error flag (IEBBnOVRE) is set (to 1).
4. In the FIFO mode, if the next reception occurs before all the data received during the previous communication is read, the communication differs in the case of individual versus broadcast communication.
   - For individual communication, a NACK signal is returned for the data field, and the master is requested to transmit the same data. Received data is not stored in the FIFO buffer. If the NACK signal is returned again before all the data in the FIFO buffer is read and the FIFO buffer has still not been read when the maximum number of transferable bytes is reached, frame completion (IEBBTSTA) and a NACK reception error (IEBBTERR) occur at the same time.
   - During broadcast communication, an overrun error occurs and communication ends. Received data is not stored in the FIFO buffer. The frame over error flag (IEBBnISR.IEBBnFOVE) is set (to 1).
5. In the FIFO mode and communication mode 2, if there are 32 bytes of unread data and the next reception occurs, the communication differs in the case of individual versus broadcast communication.
   - For individual communication, a NACK signal is returned for the field, and the master is requested to transmit the same data. Received data is not stored in the FIFO buffer. If the NACK signal is returned again before the IEBBnDR register is read and reading the register has still not finished when the maximum number of transferable bytes is reached, frame completion (IEBBTSTA) and a NACK reception error (IEBBTERR) occur at the same time.
   - During broadcast communication, an overrun error occurs and communication ends. Received data is not stored in the FIFO buffer. The overrun error flag (IEBBnESR.IEBBnOVRE) is set (to 1).

7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

chanhvinh.com

22.4 Interrupt Operations

22.4.1 Interrupt request signals

Various interrupts occur in response to the eight interrupt requests below. The high level width of each interrupt signal is one P0ϕ clock cycle.

The interrupts that occur differ depending on whether the system is in the single or FIFO mode.

(1) Single mode

The causes of interrupts in the single mode are shown blow.

Table 22-38 Causes of interrupts in the single mode

Symbol	IEBBTD	IEBBTV	IEBBTERR	IEBBTSTA	Interrupt cause
IEBBnIEBE		Occurs	Occurs		Communication error (When the IEBBnISR.IEBBnIEBE bit = 1) Note that IEBBnIEBE occurs when the following bits of the IEBBnESR register = 1. • Timing error (IEBBnTIME) • Parity error (IEBBnPARE) • NACK reception error (IEBBnNACE)[a] • Underrun error (IEBBnUNRE) • Overrun error (IEBBnOVRE)
IEBBnSTRF		Occurs		Occurs	Start request (when the IEBBnISR.IEBBnSTRF bit = 1)
IEBBnSTSF		Occurs		Occurs	Status transmission request (when the IEBBnISR.IEBBnSTSF bit = 1)
IEBBnETRF		Occurs		Occurs	End of communication (When the IEBBnISR.IEBBnETRF bit = 1)
IEBBnEFMF		Occurs		Occurs	End of frame (When the IEBBnISR.IEBBnEFMF bit = 1)[a]
IEBBnFOVE					Frame over (When the IEBBnISR.IEBBnFOVE bit = 1)
WRREQ	Occurs				Transmission data write request (when the IEBBnSSR.IEBBnSTXF bit = 0)[b]
RDREQ	Occurs				Reception data read request (when the IEBBnSSR.IEBBnSRXF bit = 1) [c]

a) If the frame data ends with a NACK signal, IEBBTV and IEBBTSTA are triggered by setting the frame completion indicating bit IEBBnISR.IEBBnEFMF (to 1).
At this time, the IEBBTERR and IEBBTV interrupts are triggered by a NACK reception error.
Three interrupts (IEBBTV, IEBBTSTA, and IEBBTERR) occur at the same time.

chanhvinh.com

b) During master transmission:

1. IEBBTD occurs after receiving the $\overline{\text{ACK}}$ signal, which follows message length field transmission. However, if the transfer size is one byte (the IEBBnTDL register = 01H), IEBBTD does not occur.

2. IEBBTD occurs after receiving the $\overline{\text{ACK}}$ signal, which follows data field transmission. However, IEBBTD does not occur before transmitting the final data, or after transmitting the final data and then receiving the ACK signal. More specifically, if the message length is five bytes, IEBBTD does not occur after transmitting the 4th or 5th byte. In addition, when transmitting the maximum number of transferable bytes for one frame, IEBBTD does not occur after transmitting byte number (maximum number of transferrable bytes – 1) or byte number (maximum number of transferable bytes).

During slave transmission:

1. IEBBTD occurs after receiving the $\overline{\text{ACK}}$ signal, which follows message length field transmission. However, if the transfer size is one byte or the received control bit is a status request (0H, 4H, 5H, or 6H), IEBBTD does not occur (and a status transmission interrupt occurs instead).

2. After data field transmission, the operation is the same as for *2* under During master transmission.

c) RDREQ occurs after receiving the parity bit by using the data field. However, if the self-transmitted parity bit differs from the received parity bit, there is a timing error and no interrupt occurs.

chanhvinh.com

(2) FIFO mode

The causes of interrupts in the FIFO mode are shown below.

Table 22-39 Causes of interrupts in the FIFO mode

Symbol	IEBBTD	IEBBTV	IEBBTERR	IEBBTSTA	Interrupt cause
IEBBnIEBE			Occurs		Communication error (When the IEBBnISR.IEBBnIEBE bit = 1) Note that IEBBnIEBE occurs when the following bits of the IEBBnESR register = 1. • Timing error (IEBBnTIME) • Parity error (IEBBnPARE) • NACK reception error (IEBBnNACE)[a] • Underrun error (IEBBnUNRE) • Overrun error (IEBBnOVRE)[b] • Arbitration loss error (IEBBnABTE)
IEBBnSTRF					Start request (when the IEBBnISR.IEBBnSTRF bit = 1)
IEBBnSTSF					Status transmission request (when the IEBBnISR.IEBBnSTSF bit = 1)
IEBBnETRF		Occurs[c]		Occurs	End of communication (When the IEBBnISR.IEBBnETRF bit = 1)
IEBBnEFMF		Occurs[c]		Occurs	End of frame (When the IEBBnISR.IEBBnEFMF bit = 1)[a]
IEBBnFOVE			Occurs		Frame over (When the IEBBnISR.IEBBnFOVE bit = 1)[b]
WRREQ	Occurs[d]				Transmit data write request
RDREQ		Occurs[e]			Receive data write request Parity

a) If the frame data ends with a NACK signal, IEBBTV and IEBBTSTA are triggered by setting the frame completion indicating bit IEBBnISR.IEBBnEFMF (to 1).
Note that the IEBBTERR interrupt is triggered by a NACK reception error.
Three interrupts (IEBBTV, IEBBTSTA, and IEBBTERR) occur at the same time, regardless of transmission or reception.

b) If data is received during the broadcast communication for the next frame while the FIFO buffer is full due to the reception of the previous frame and the data has not been read, the IEBBnISR.IEBBnFOVE and IEBBnESR.IEBBnOVRE bits are set (to 1) at the same time.

chanhvinh.com

c) IEBBTV occurs based on the same conditions and at the same timing as IEBBTSTA. (For details, see Table 22-40 $\overline{\text{ACK}}$/NACK for IEBBnETRF and IEBBnEFMF.)
For individual communication:
1. Transmission-side device

IEBBnETRF:
This is set (to 1) after receiving the $\overline{\text{ACK}}$ signal. If the NACK signal is received, communication does not end.

IEBBnEFMF: This is set (to 1) after reception regardless of the $\overline{\text{ACK}}$/NACK signal.

2. Reception-side device

IEBBnETRF:
This is set (to 1) after transmitting the $\overline{\text{ACK}}$ signal. If the NACK signal is transmitted, communication does not end.

IEBBnEFMF:
This is set (to 1) after transmission regardless of the $\overline{\text{ACK}}$/NACK signal.
The NACK signal is output when there is no room in the reception FIFO buffer or when there is a data retransmission request due to a parity mismatch. Not returning an ACK signal to the reception side as the bus status after exiting communications due to the detection of an error does not constitute a NACK signal. In this case, IEBBTSTA and IEBBTV do not occur because the system does not enter the frame completion status.

For broadcast communication:
For broadcast communication, no $\overline{\text{ACK}}$ signal is returned from the slave. Therefore, it is judged that the NACK signal was successfully returned regardless of master transmission or slave reception, and the IEBBnETRF or IEBBnEFMF interrupt occurs.

d) This occurs when the conditions specified by the IEBBnTMS.IEBBnSLTI1 and IEBBnSLTI0 bits are satisfied. The occurrence timing is after receiving the $\overline{\text{ACK}}$ signal, which follows data field transmission. However, if the transfer size is one byte or the received control bit is a status request (0H, 4H, 5H, or 6H), IEBBTD does not occur.

e) 1. Transmission-side device
RDREQ is not set under this condition.
2. Reception-side device
RDREQ is set (to 1) when the conditions specified by the IEBBnTMS.IEBBnSLRI1 and IEBBnSLRI0 bits are satisfied (after confirming that a normal parity bit value has been received).
If there is a parity bit mismatch, RDREQ is not set (to 1) because the conditions are not satisfied.
On normal completion or frame completion, RDREQ interrupt sources are masked.

Table 22-40 $\overline{\text{ACK}}$/NACK for IEBBnETRF and IEBBnEFMF

Field status	Individual communication				Broadcast communication			
	IEBBnETRF		IEBBnEFMF		IEBBnETRF		IEBBnEFMF	
	$\overline{\text{ACK}}$	NACK	$\overline{\text{ACK}}$	NACK	$\overline{\text{ACK}}$	NACK	$\overline{\text{ACK}}$	NACK
Master transmission	Occurs	Does not occur	Occurs	Occurs	–	Occurs	–	Occurs
Master reception	Occurs	Does not occur	Occurs	Occurs	–	–	–	–
Slave transmission	Occurs	Does not occur	Occurs	Occurs	–	–	–	–
Slave reception	Occurs	Does not occur	Occurs	Occurs	–	Occurs	–	Occurs

chanhvinh.com

Note The IEBBTV usage method is shown below.

1. Generating IEBBTV before communication finishes (when the IEBBnTMS.IEBBnSLRI1 and IEBBnSLRI0 bit settings are 32 bytes or less)
   This operation is not generally performed in communication mode 1. (It can be performed but is not.) The operation is performed in communication mode 2.
   – Use IEBBTV to check the number of data bytes received by the FIFO buffer, and then perform a readout operation. (Be sure to check the number because the IEBBTV interrupt servicing is assumed to be late.)
   – For IEBBTSTA (communication completion/frame completion), check the status.
   – Because the status is changed after receiving data, the interrupt priority is IEBBTV followed by IEBBTSTA.
2. If not generating IEBBTV during communication (if the IEBBnSLRI1 and IEBBnSLRI0 bit settings are 32 bytes)

   Use communication mode 1. (Communication mode 2 can also be used.)
   – Communication mode 1

   Mask IEBBTV (so it is not used).
   Use IEBBTSTA (communication completion/frame completion) to check the number of data bytes in the FIFO buffer, and then check the readout and status changes.
   – Communication mode 2

   Perform the same operations as in 1.

chanhvinh.com

22.4.2 Interrupt judgment examples

Interrupt judgment examples for the single mode are shown below.

(1) When using IEBBTD

IEBBn transmission/reception must be checked by issuing an IEBBTD interrupt.

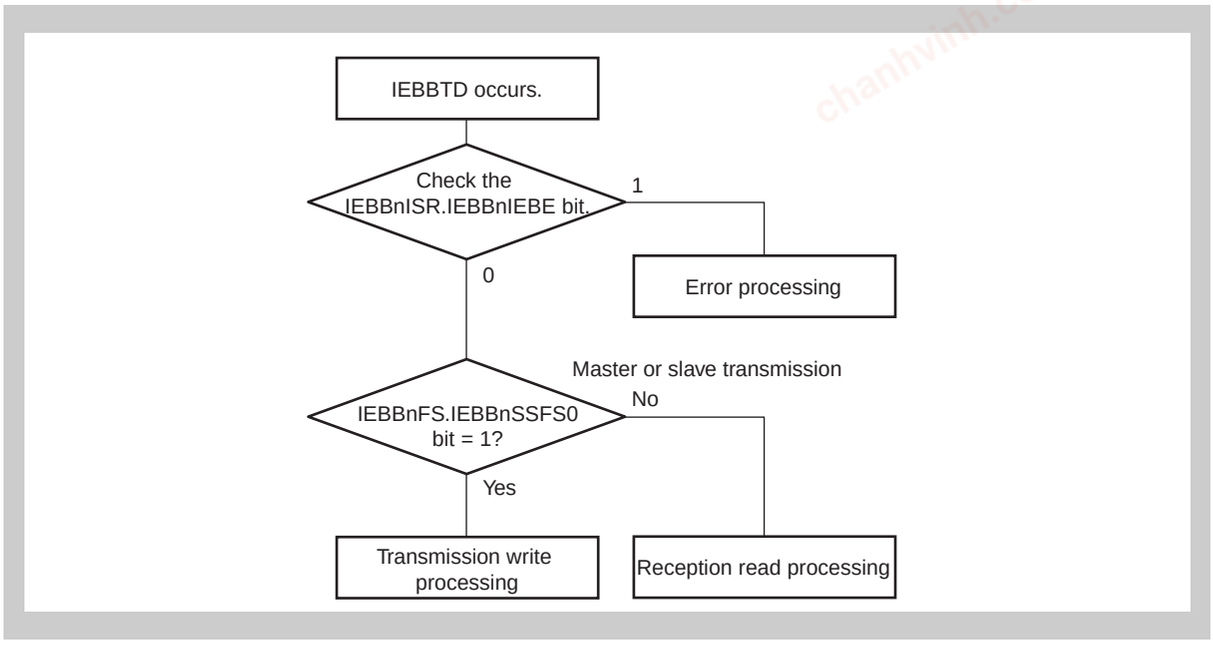


Figure 22-9 IEBBTD interrupt judgment example

Caution Even if IEBBTD occurs, an error might occur depending on when the interrupt is handled.

Such errors include timing errors after IEBBTD occurs. To increase data processing reliability, only handle data after using the IEBBnISR.IEBBnIEBE bit to make sure that no error has occurred.

(2) When using IEBBTERR

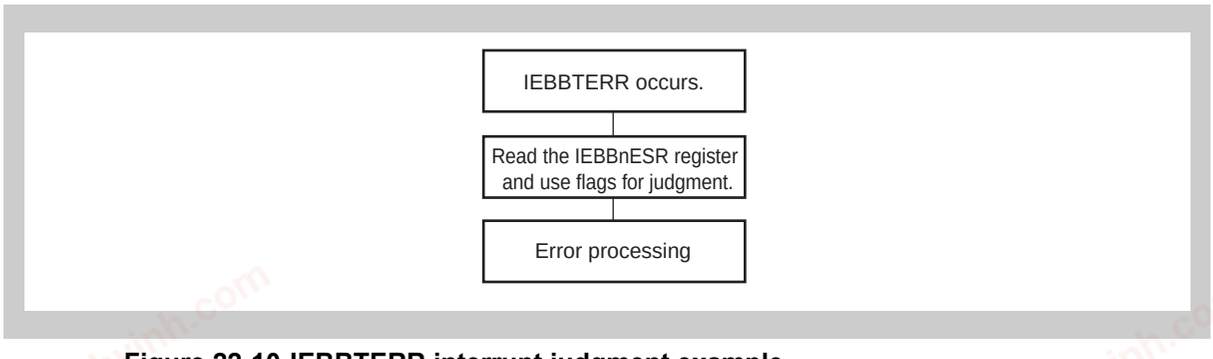


Figure 22-10 IEBBTERR interrupt judgment example

chanhvinh.com

(3) When using IEBBTSTA or IEBBTV

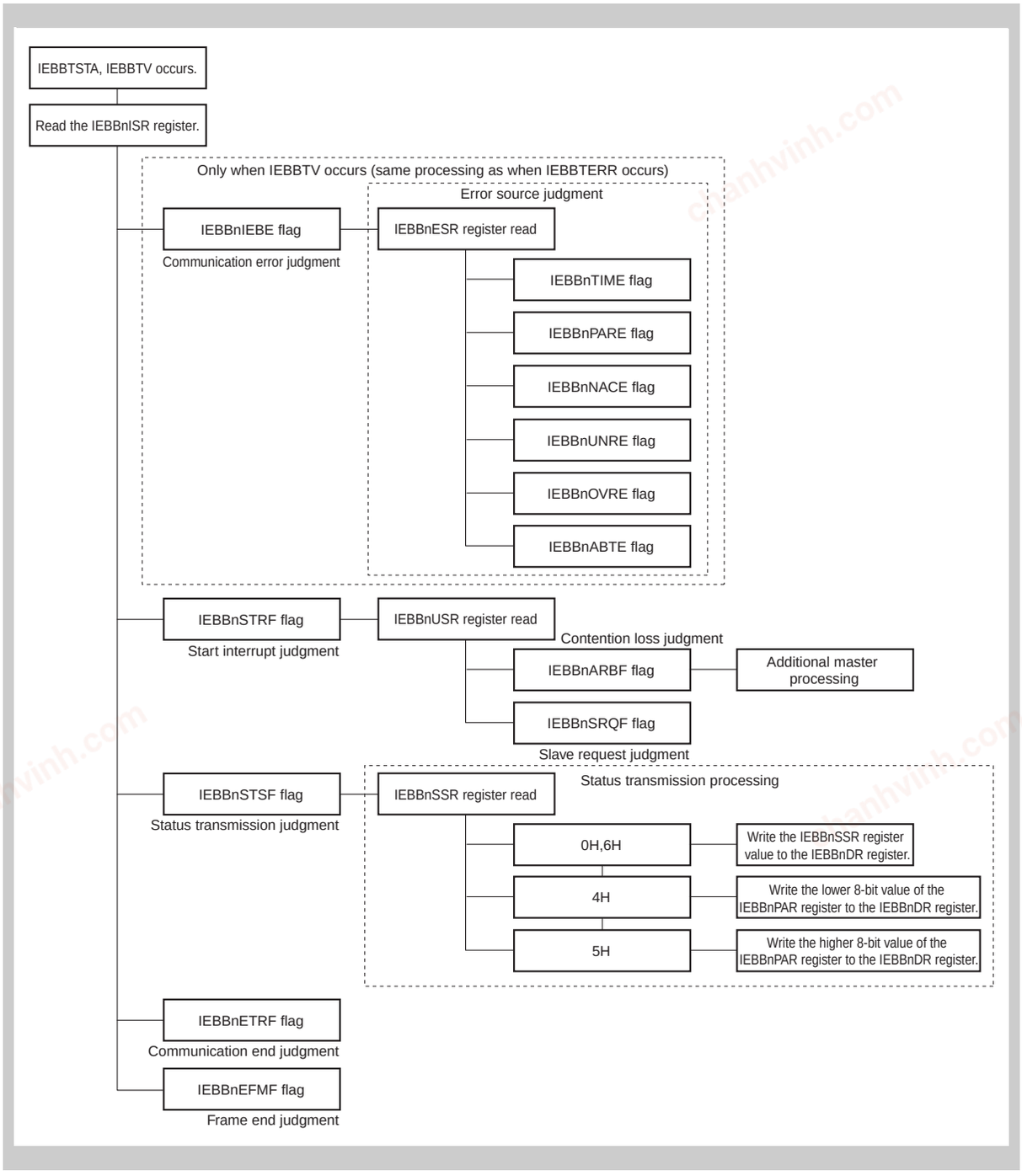


Figure 22-11 IEBBTSTA and IEBBTV interrupt judgment examples

chanhvinh.com

22.5 Operation

22.5.1 FIFO

(1) Transmission FIFO buffer

When the IEBBnTMS.IEBBnFMDE bit = 1, data can be stored in the FIFO buffer while automatically incrementing the FIFO buffer pointer for writing by continuously writing transmission data to the IEBBnDR register. The FIFO buffer size is 8 bits x 32 stages.

When the transfer is started, the data indicated by the load pointer is transferred. After the transfer finishes, the load pointer is incremented. The initial value for the write pointer and load pointer is 0.

The IEBBnBSR.IEBBnTFLF bit is set (to 1) when there are 32 bytes of data in the FIFO buffer, and the transmission FIFO buffer overwrite flag (IEBBnBSR.IEBBnFOVW) is set (to 1) when a 33rd byte of data is written while the IEBBnTFLF bit = 1. At this time, the write data is ignored and the write pointer is not changed.

The data below can be written when one byte is transferred while the IEBBnTFLF bit = 1 and then the bit is cleared to 0.

If the write is not in time for data loading, an underrun error occurs.

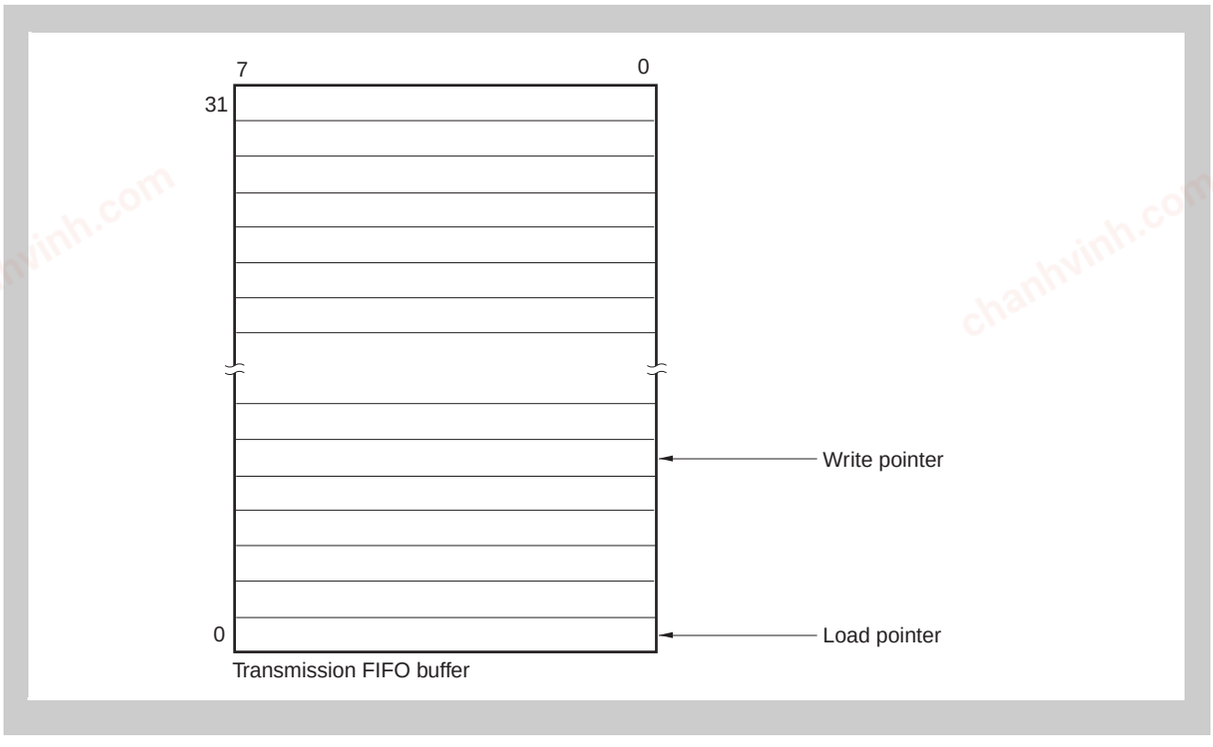


Figure 22-12 Transmission FIFO buffer

chanhvinh.com

(2) Reception FIFO buffer

When the IEBBnTMS.IEBBnFMDE bit = 1, the received data is stored at the address indicated by the storage pointer. The storage pointer is incremented after storing the data. Data is stored in the FIFO buffer at the end of the data field parity period if the parity value is normal. The FIFO buffer size is 8 bits x 32 stages.

By reading the IEBBnDR register, the data in the FIFO buffer can be read while automatically incrementing the read pointer. The initial value for the read pointer and storage pointer is 0.

If there are 32 bytes of unread data in the FIFO buffer, the operation when the next data is received is as follows.

- During individual communication: No data is stored, a NACK signal is returned, and data retransmission is requested.
- During broadcast communication: No data is stored and an overrun error occurs.

  If the IEBBnDR register is read again after reading the received data that has been stored finishes (when the read pointer = the storage pointer), the read pointer is not changed, and the over-read flag (IEBBnBSR.IEBBnFOVR bit) is set (to 1).

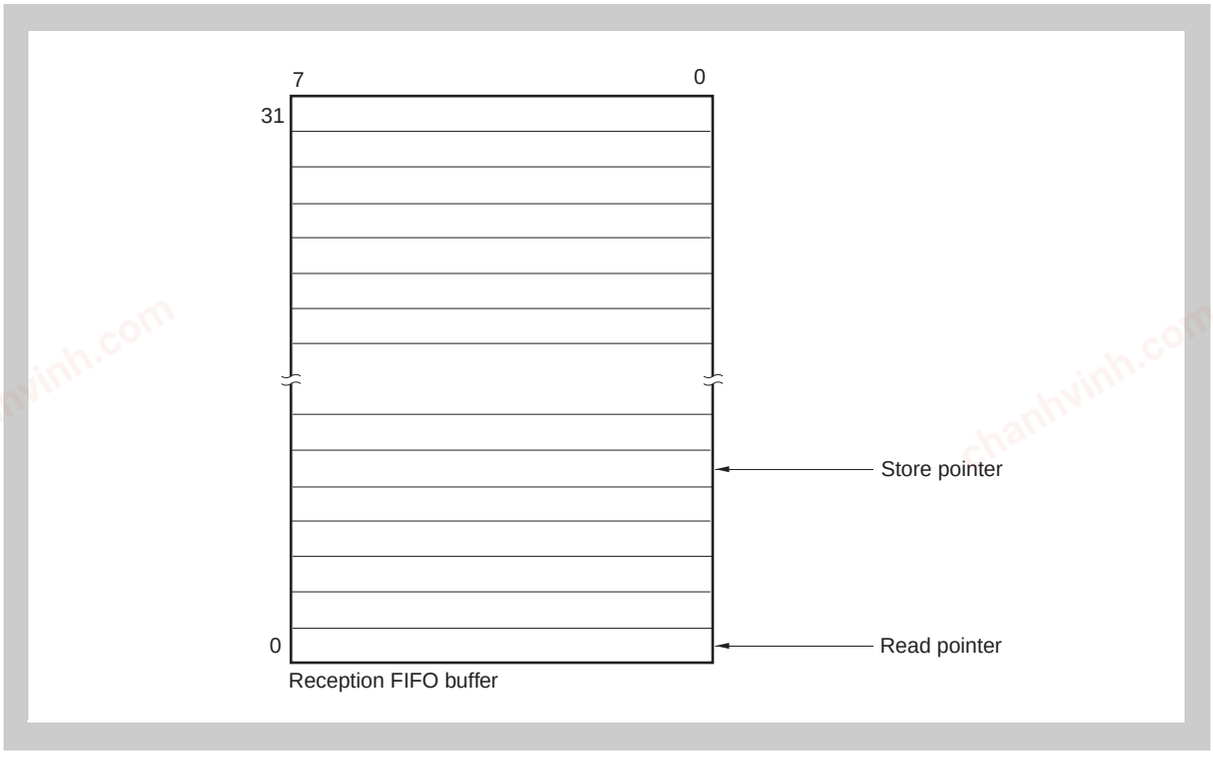


Figure 22-13 Reception FIFO buffer

chanhvinh.com

22.5.2 Initial settings

After setting the IEBBnBCR.IEBBnPW bit to 1, set up the registers below, and then start communication processing.

Table 22-41 Initial setup

Register name	Function	Example
IEBBnPSR	Operation clock and communication mode settings	80H
IEBBnUAR	Set a unit address.	101H
IEBBnCKS	Clock Selection	15H
IEBBnTMS	Communication control	01H[a]

a) For use in the single mode, the initial values do no have to be changed.

chanhvinh.com

22.5.3 Master transmission (single mode)

The unit transmits data and commands to the slave unit as the master.

Data interrupts are used to write transmission data to the IEBBnDR register for each one-byte transfer.

(1) Register settings

After specifying the initial settings in 22.5.2 Initial settings, set up the registers below before starting communication.

Table 22-42 Standard initial processing

Register name	Function	Example
IEBBnSAR	Communication partner unit address	102H
IEBBnCDR or IEBBnTCD	Control data (AH, BH, EH, FH)	FH
IEBBnDLR	Message length	02H
IEBBnDR	Data (1st byte of data)	11H

Table 22-43 Communication startup processing

Register name	Function	Example
IEBBnBCR	Communication startup processing	C8H

(2) Interrupt occurrence timing

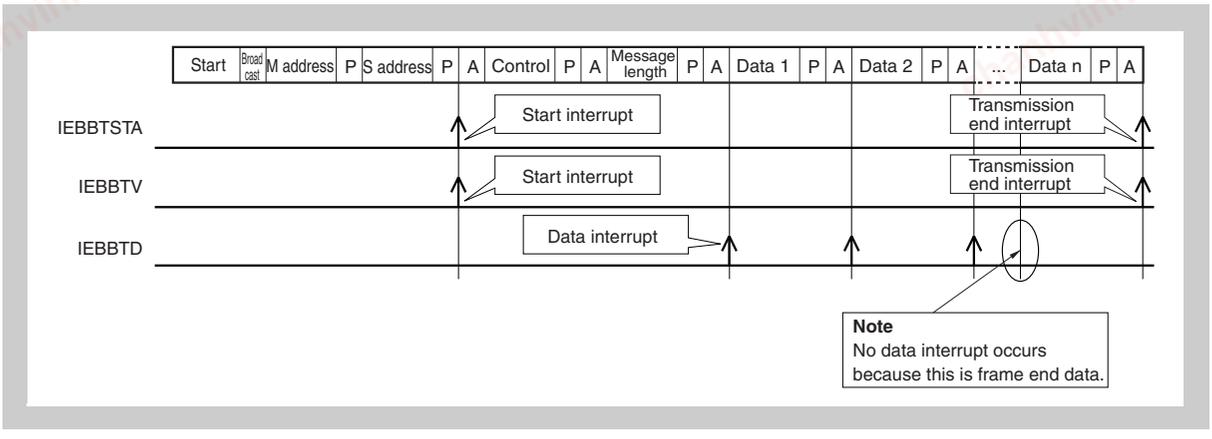


Figure 22-14 Interrupt occurrence timing

chanhvinh.com

(3) Interrupt servicing examples

(a) Start interrupt CPU processing flow example

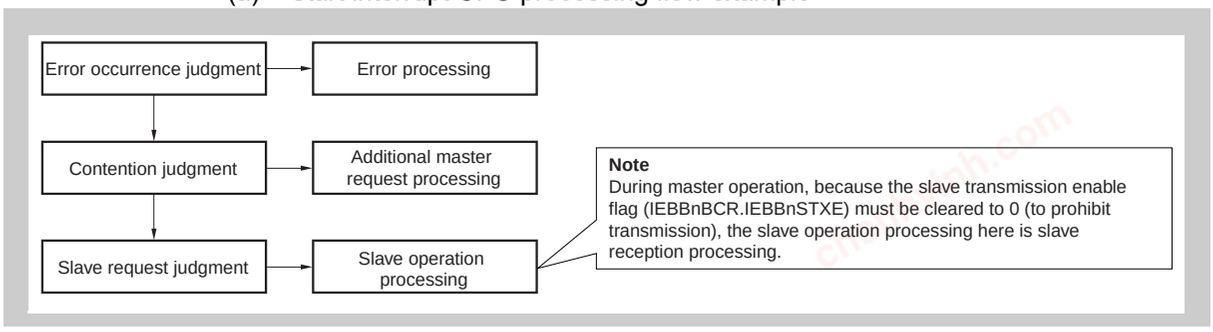


Figure 22-15 Start interrupt CPU processing flow example

(b) Transmission completion interrupt (IEBBTV, IEBBTSTA) CPU processing flow example

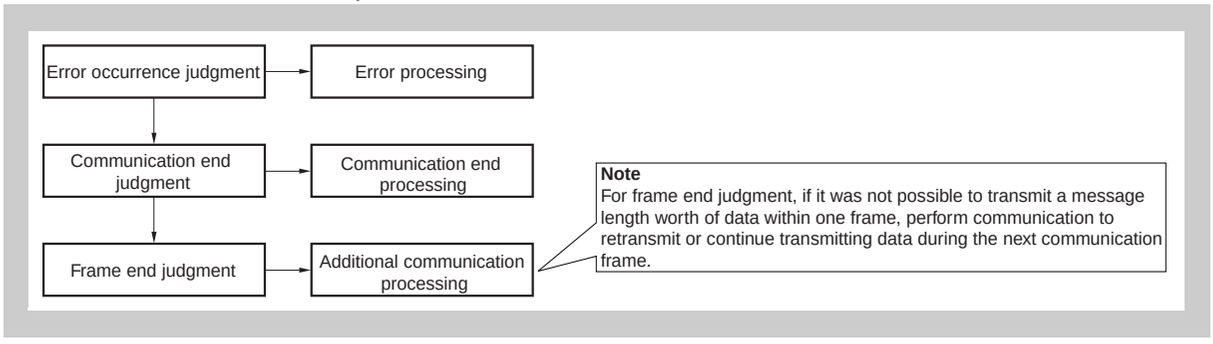


Figure 22-16 Transmission completion interrupt (IEBBTV, IEBBTSTA) CPU processing flow example

(c) Data interrupt CPU processing example

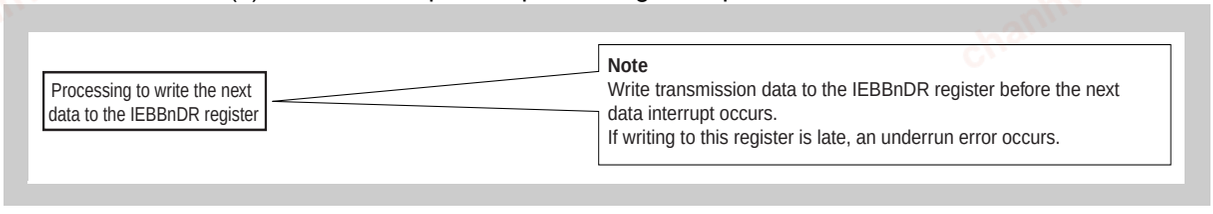


Figure 22-17 Data interrupt CPU processing

chanhvinh.com

22.5.4 Master transmission (FIFO mode)

The unit transmits data and commands to the slave unit as the master.

Transmission data is written into the buffer in advance, and then a master request is issued.

(1) Register settings

After specifying the initial settings in 22.5.2 Initial settings, set up the registers below before starting communication.

Table 22-44 Standard initial processing

Register name	Function	Example
IEBBnSAR	Communication partner unit address	102H
IEBBnCDR or IEBBnTCD	Control data (AH, BH, EH, FH)	FH
IEBBnTDL	Message length	02H
IEBBnDR	Data (all data to be transmitted)	11H, …

Table 22-45 Communication startup processing

Register name	Function	Example
IEBBnBCR	Communication startup processing	C8H

(2) Interrupt occurrence timing

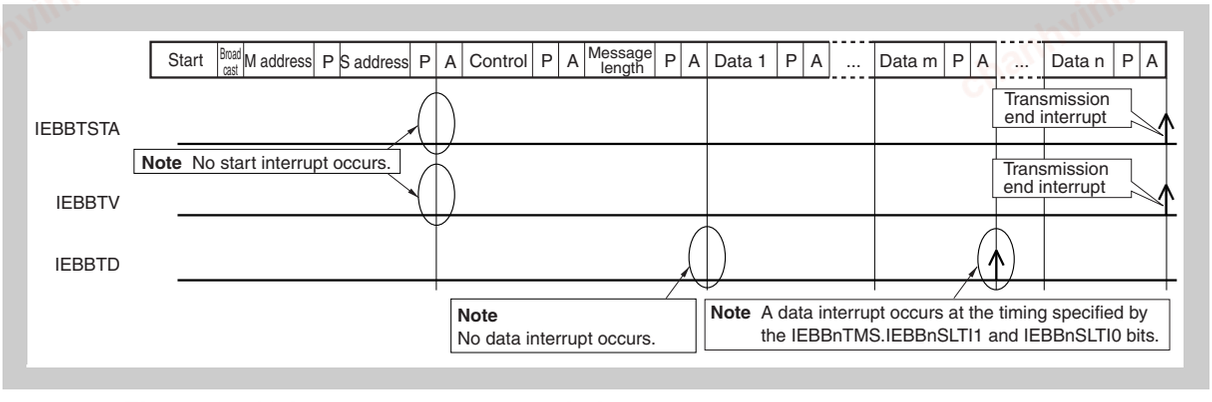


Figure 22-18 Interrupt occurrence timing

chanhvinh.com

(3) Interrupt servicing examples

(a) Transmission completion interrupt (IEBBTV, IEBBTSTA) CPU processing flow example

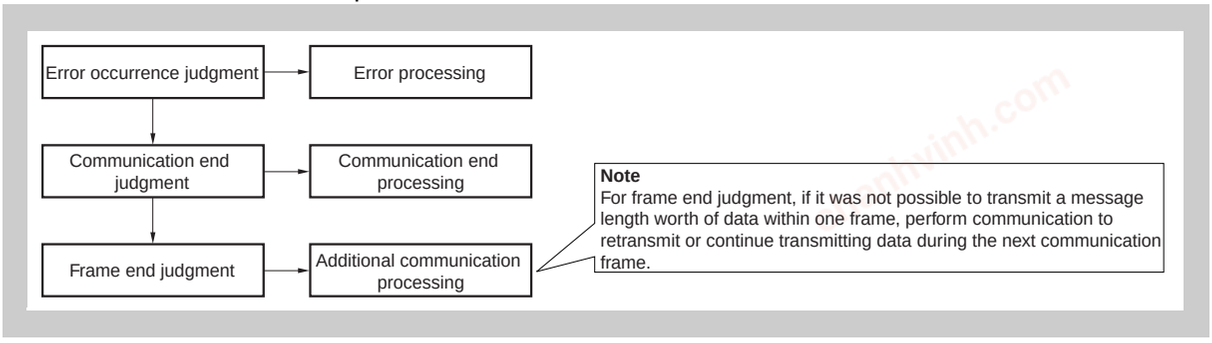


Figure 22-19 Transmission completion interrupt (IEBBTV, IEBBTSTA) CPU processing flow example

chanhvinh.com

22.5.5 Master reception (single mode)

The unit receives data and commands from the slave unit as the master. Because the slave transfers the message length field in the case of master reception, indicate the message length of data to be transmitted to the slave, such as during other communication. Read the received data one byte at a time by using data interrupts.

(1) Register settings

After specifying the initial settings in 22.5.2 Initial settings, set up the registers below before starting communication.

Table 22-46 Standard initial processing

Register name	Function	Example
IEBBnSAR	Communication partner unit address	102H
IEBBnCDR or IEBBnTCD	Control data (0H, 3H, 4H, 5H, 6H, 7H)	7H

Table 22-47 Communication startup processing

Register name	Function	Example
IEBBnBCR	Communication startup processing	C8H

(2) Interrupt occurrence timing

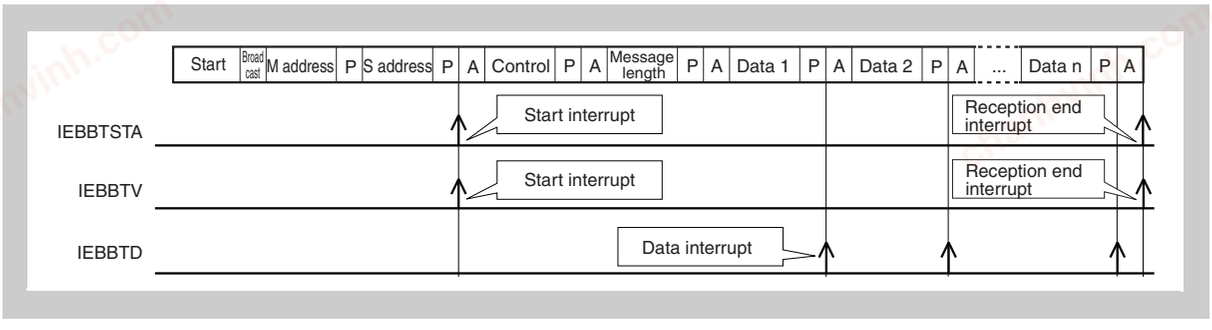


Figure 22-20 Interrupt occurrence timing

chanhvinh.com

(3) Interrupt servicing examples

(a) Start interrupt CPU processing flow example

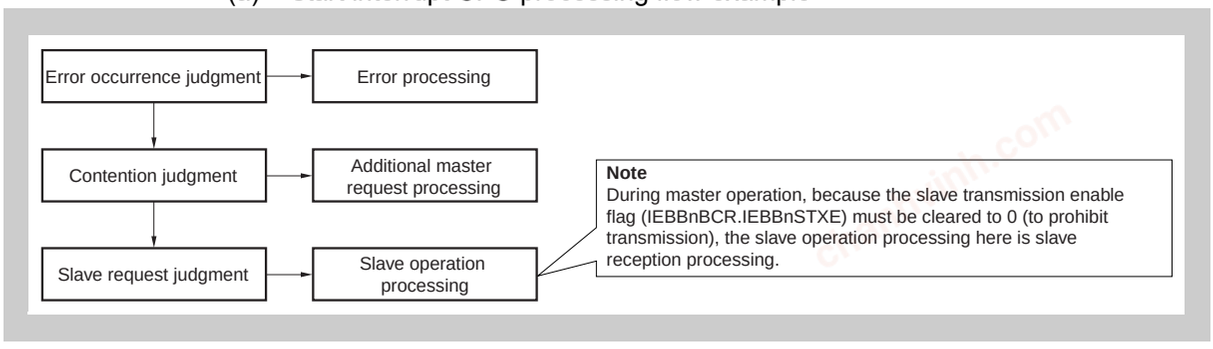


Figure 22-21 Start interrupt CPU processing flow example

(b) Reception completion interrupt (IEBBTV, IEBBTSTA) CPU processing flow example

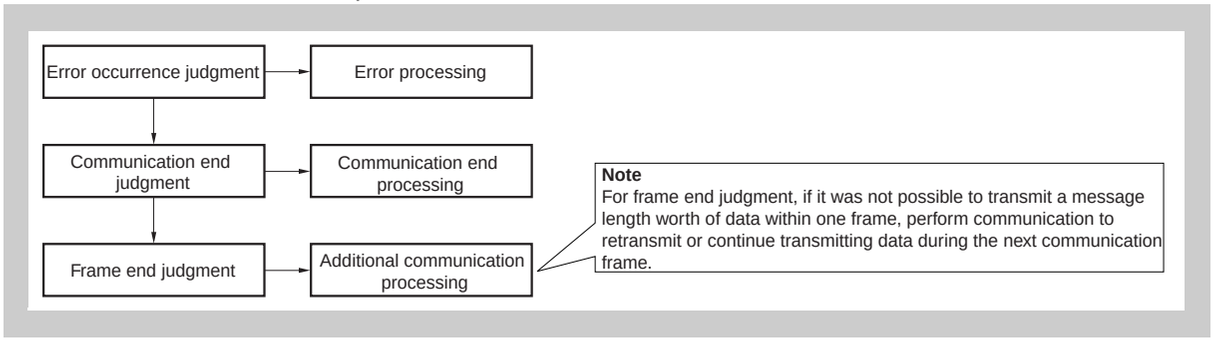


Figure 22-22 Reception completion interrupt (IEBBTV, IEBBTSTA) CPU processing flow example

(c) Data interrupt CPU processing example

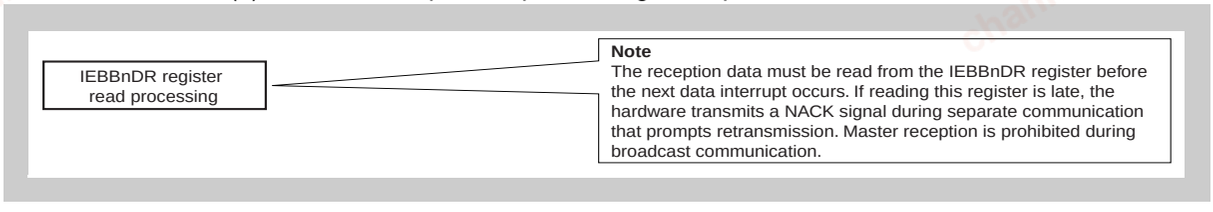


Figure 22-23 Data interrupt CPU processing

chanhvinh.com

22.5.6 Master reception (FIFO mode)

The unit receives data and commands from the slave unit as the master. Because the slave transfers the message length field in the case of master reception, indicate the message length of data to be transmitted to the slave, such as during other communication.

(1) Register settings

After specifying the initial settings in 22.5.2 Initial settings, set up the registers below before starting communication.

Table 22-48 Standard initial processing

Register name	Function	Example
IEBBnSAR	Communication partner unit address	102H
IEBBnCDR or IEBBnTCD	Control data (0H, 3H, 4H, 5H, 6H, 7H)	7H

Table 22-49 Communication startup processing

Register name	Function	Example
IEBBnBCR	Communication startup processing	C8H

(2) Interrupt occurrence timing

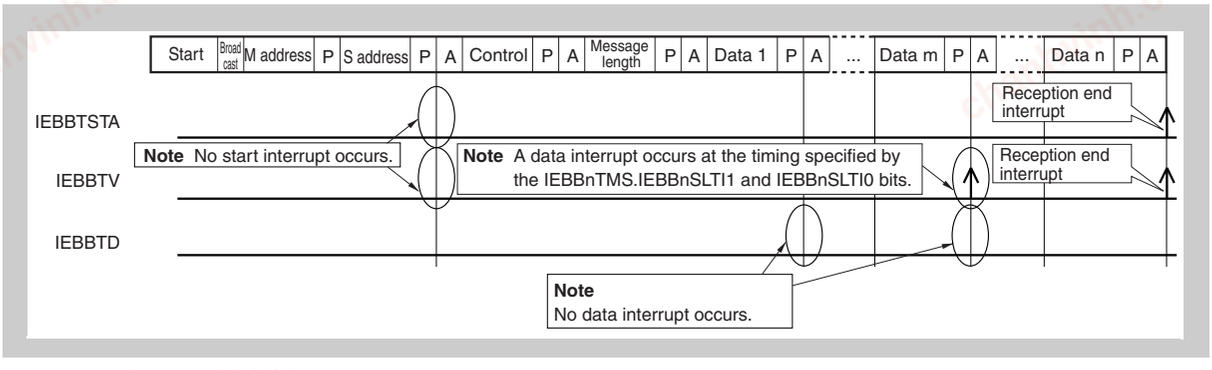


Figure 22-24 Interrupt occurrence timing

chanhvinh.com

(3) Interrupt servicing examples

(a) Reception completion interrupt (IEBBTV, IEBBTSTA) CPU processing flow example

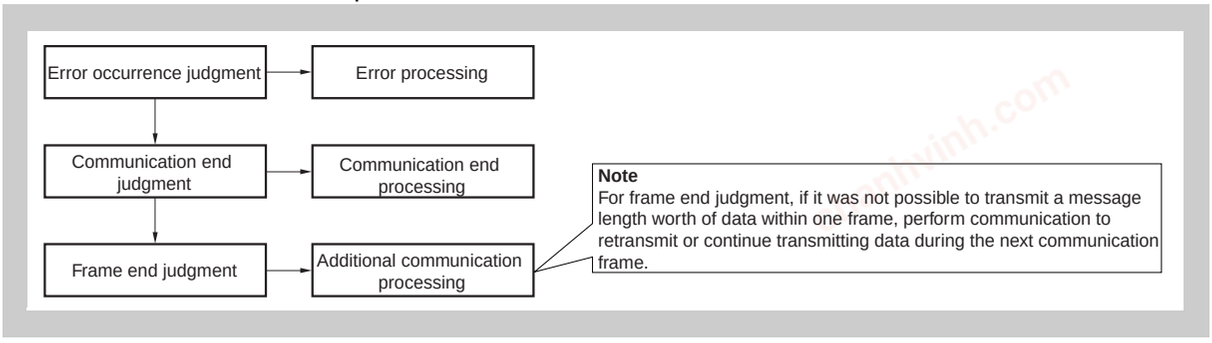


Figure 22-25 Reception completion interrupt (IEBBTV, IEBBTSTA) CPU processing flow example

chanhvinh.com

22.5.7 Slave transmission (single mode)

The unit transfers data and commands to the master unit as a slave.

Data interrupts are used to write transmission data to the IEBBnDR register for each one-byte transfer.

(1) Register settings

After specifying the initial settings in 22.5.2 Initial settings, set up the registers below before starting communication.

Table 22-50 Standard initial processing

Register name	Function	Example
IEBBnDLR	Message length (other than during slave status transmission)	02H
IEBBnDR	Data (1st byte of data)	11H

Caution When starting slave transmission, information such as the value to be set to the message length register (IEBBnDLR) and which data is to be returned (the value to be set to the IEBBnDR register) must be assigned in advance by the master, such as during separate communication.

Table 22-51 Communication startup processing

Register name	Function	Example
IEBBnBCR	Communication startup processing	90H

(2) Interrupt occurrence timing

(a) When the control bit 3H or 7H, which is addressed to the unit, is received

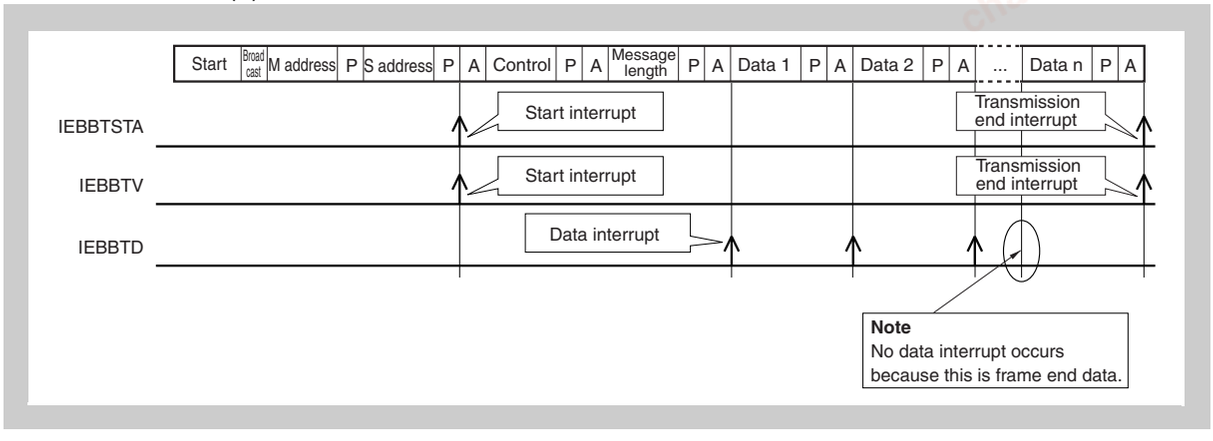


Figure 22-26 When the control bit 3H or 7H, which is addressed to the unit, is received

chanhvinh.com

(b) When the control bit 0H or 6H, which is addressed to the unit, is received (or when 4H or 5H is received from the locked master while the unit is locked)

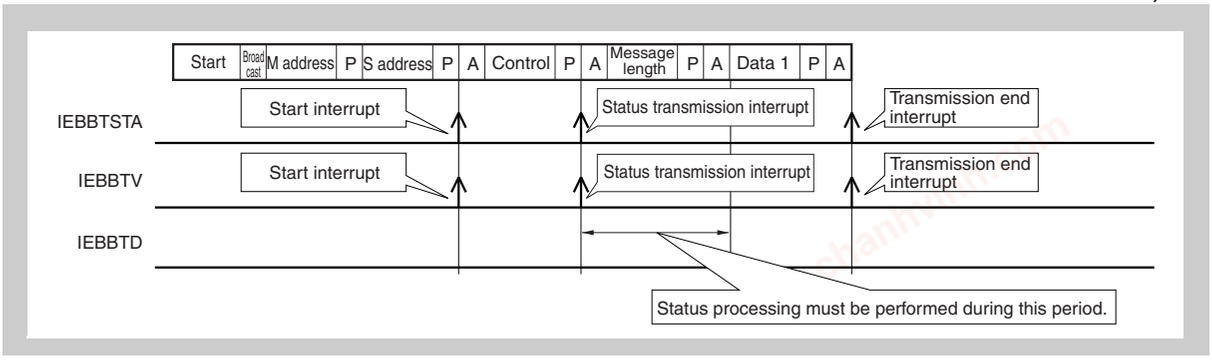


Figure 22-27 When the control bit 0H or 6H, which is addressed to the unit, is received (or when 4H or 5H is received from the locked master while the unit is locked)

(c) When the control bit 0H, 4H, or 5H, which is addressed to the unit, is received from a unit other than the locked master while the unit is locked

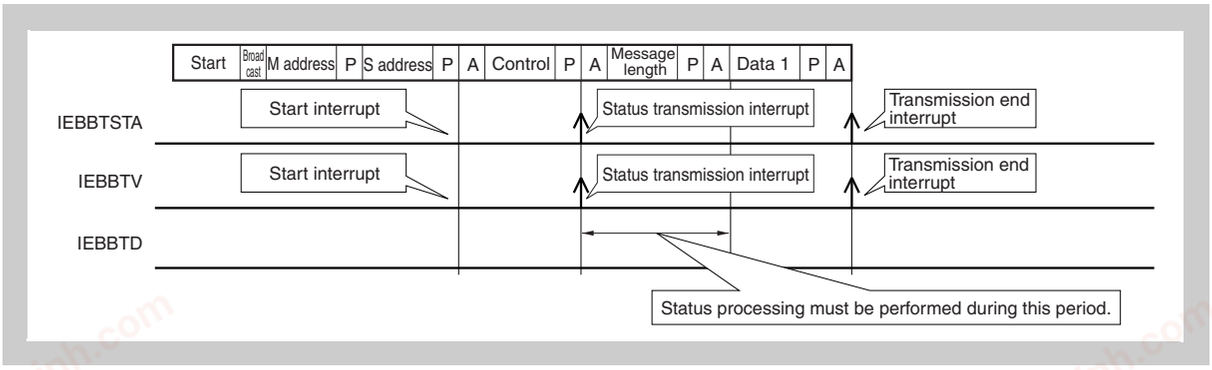


Figure 22-28 When the control bit 0H, 4H, or 5H, which is addressed to the unit, is received from a unit other than the locked master while the unit is locked

chanhvinh.com

(3) Interrupt servicing examples

(a) Start interrupt CPU processing flow example

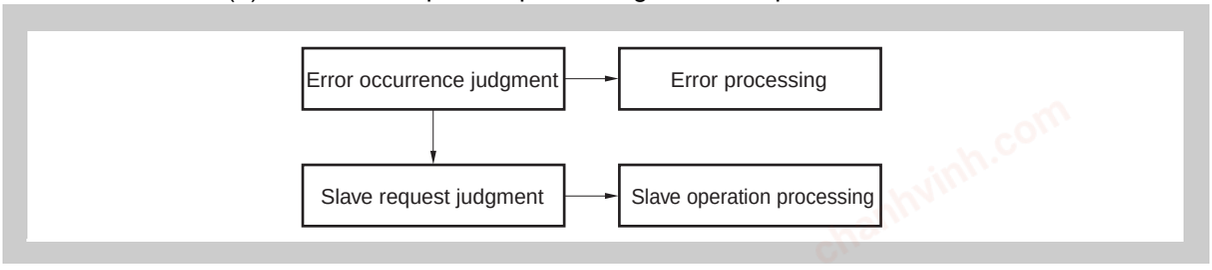


Figure 22-29 Start interrupt CPU processing flow example

(b) Transmission completion interrupt (IEBBTV, IEBBTSTA) CPU processing flow example

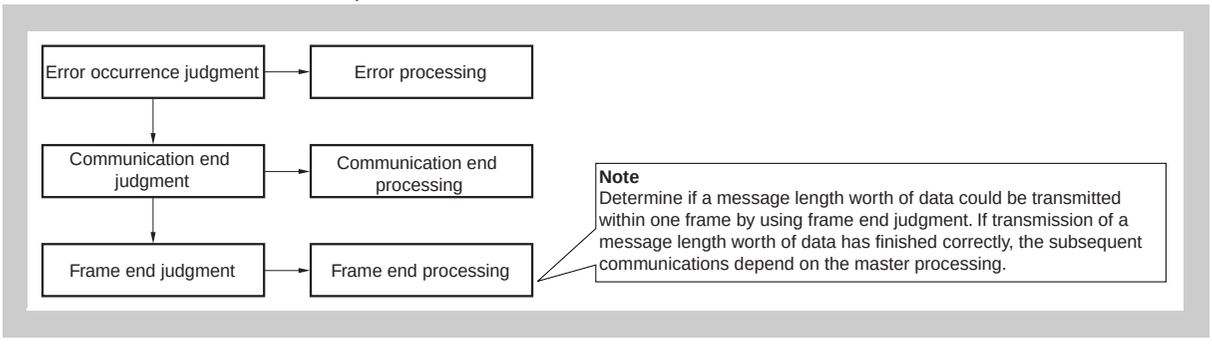


Figure 22-30 Transmission completion interrupt (IEBBTV, IEBBTSTA) CPU processing flow example

(c) Slave status transmission request processing flow example

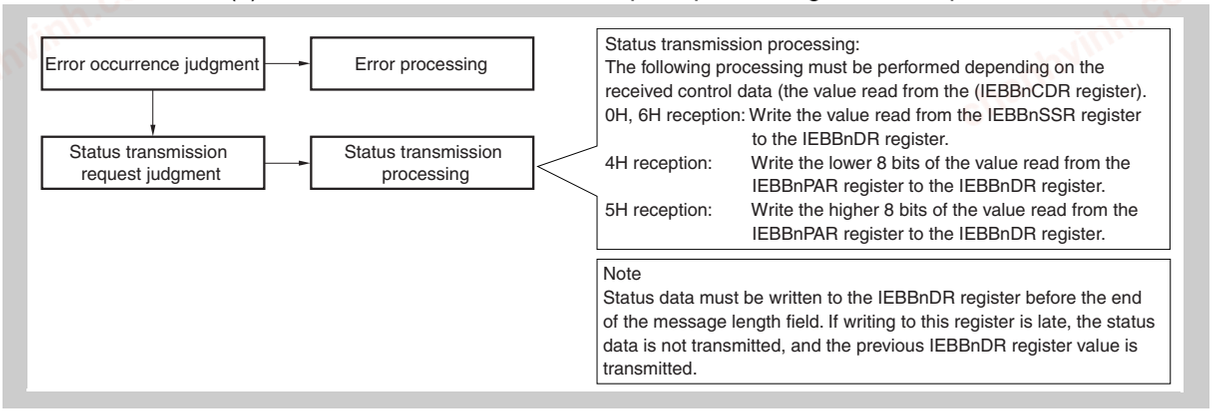


Figure 22-31 Slave status transmission request processing flow

(d) Data interrupt CPU processing example

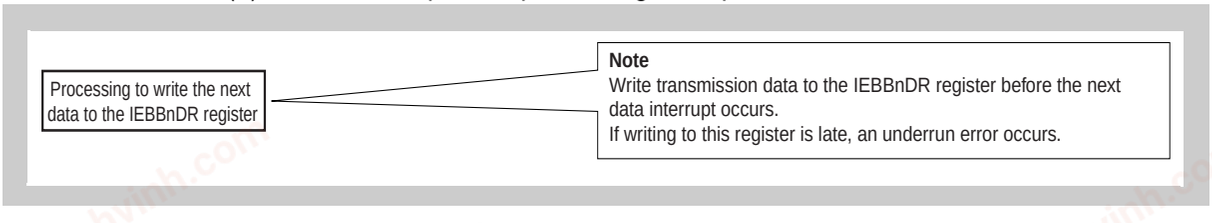


Figure 22-32 Data interrupt CPU processing

chanhvinh.com

22.5.8 Slave transmission (FIFO mode)

The unit transfers data and commands to the master unit as a slave.

Write the transmission data to the buffer in advance.

(1) Register settings

After specifying the initial settings in 22.5.2 Initial settings, set up the registers below before starting communication.

Table 22-52 Standard initial processing

Register name	Function	Example
IEBBnTDL	Message length (other than during slave status transmission)	02H
IEBBnDR	Data (all data to be transmitted)	11H, …

Caution When starting slave transmission, information such as the value to be set to the message length register (IEBBnTDL) and which data is to be returned (the value to be set to the IEBBnDR register) must be assigned in advance by the master, such as during separate communication.

Table 22-53 Communication startup processing

Register name	Function	Example
IEBBnBCR	Communication startup processing	90H

(2) Interrupt occurrence timing

(a) When the control bit 3H or 7H, which is addressed to the unit, is received

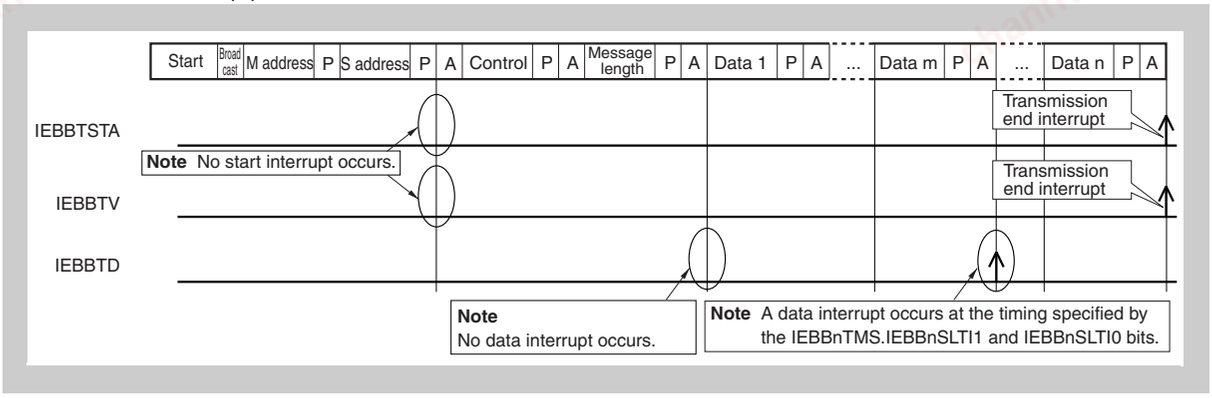


Figure 22-33 When the control bit 3H or 7H, which is addressed to the unit, is received

chanhvinh.com

(b) When the control bit 0H or 6H, which is addressed to the unit, is received (or when 4H or 5H is received from the locked master while the unit is locked)

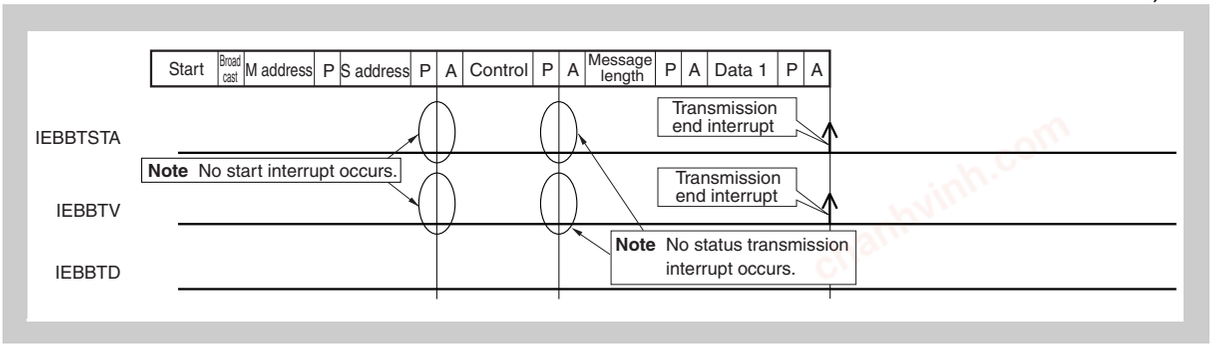


Figure 22-34 When the control bit 0H or 6H, which is addressed to the unit, is received (or when 4H or 5H is received from the locked master while the unit is locked)

(c) When the control bit 0H, 4H, or 5H, which is addressed to the unit, is received from a unit other than the locked master while the unit is locked

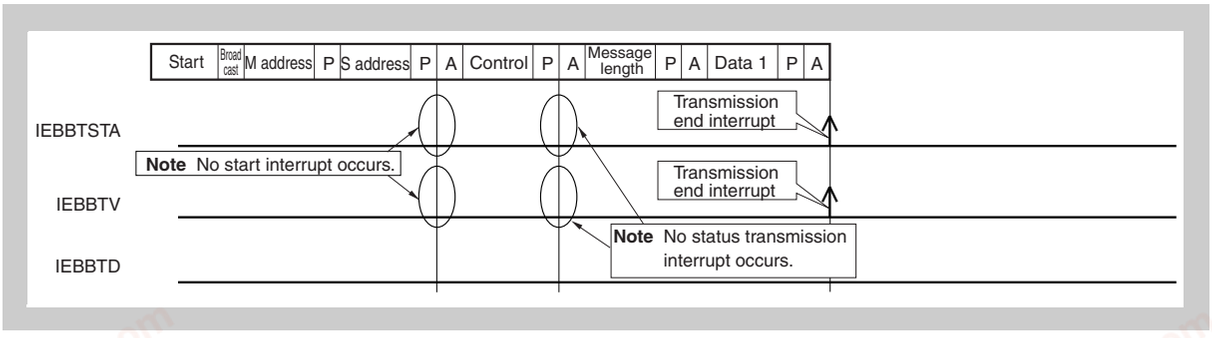


Figure 22-35 When the control bit 0H, 4H, or 5H, which is addressed to the unit, is received from a unit other than the locked master while the unit is locked

chanhvinh.com

(3) Interrupt servicing examples

(a) Transmission completion interrupt (IEBBTV, IEBBTSTA) CPU processing flow example

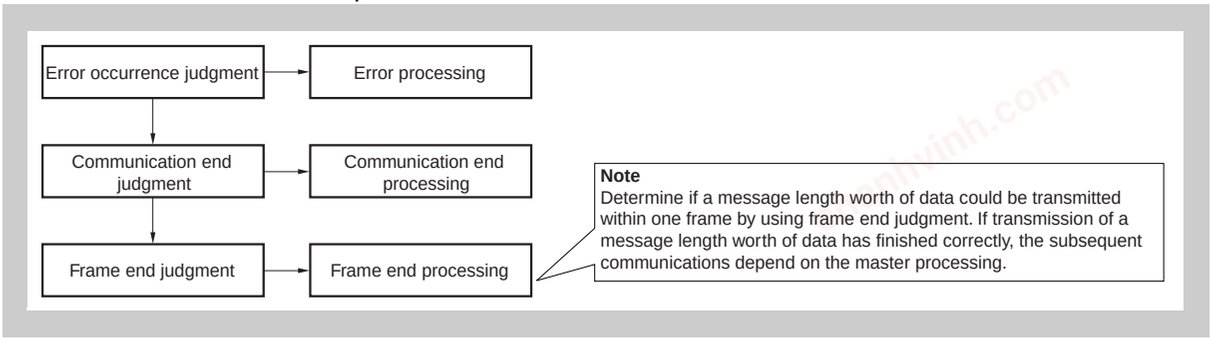


Figure 22-36 Transmission completion interrupt (IEBBTV, IEBBTSTA) CPU processing flow example

(b) Slave status transmission request processing flow example

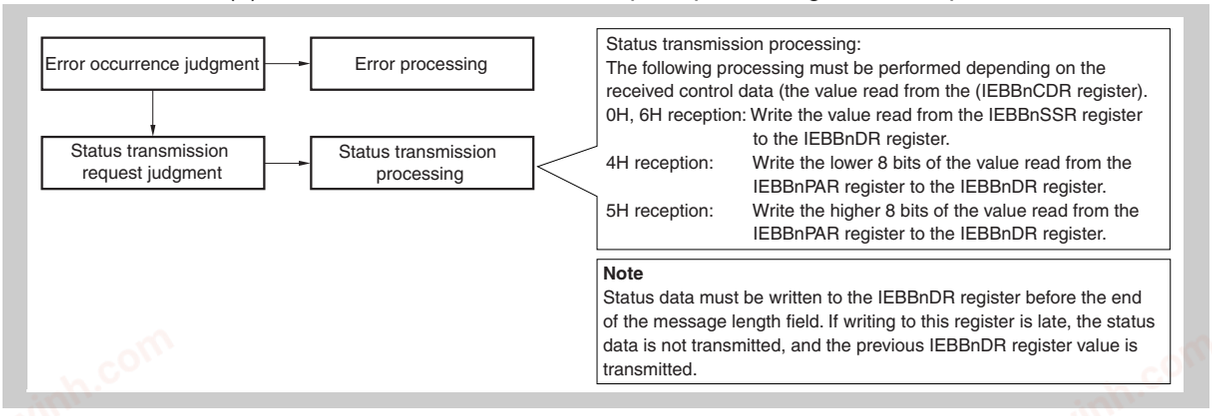


Figure 22-37 Slave status transmission request processing flow

In the FIFO mode, if the unit loses a conflict between a unit master transmission request and a slave transmission request addressed to the unit, because the slave transmission enable flag (IEBBnBCR.IEBBnSTXE) is not set (to 1) for the unit, a NACK signal is transmitted using the control data field and communication ends. Next, specify the slave transmission data for the FIFO buffer, set the slave transmission enable flag (to 1), and prepare to receive another slave transmission request from the master.

chanhvinh.com

22.5.9 Slave reception (single mode)

The unit receives data and commands from the master unit as a slave.

Read the received data one byte at a time by using data interrupts.

(1) Register settings

After specifying the initial settings in 22.5.2 Initial settings, set up the registers below before starting communication.

Table 22-54 Communication startup processing

Register name	Function	Example
IEBBnBCR	Communication startup processing	88H

(2) Interrupt occurrence timing

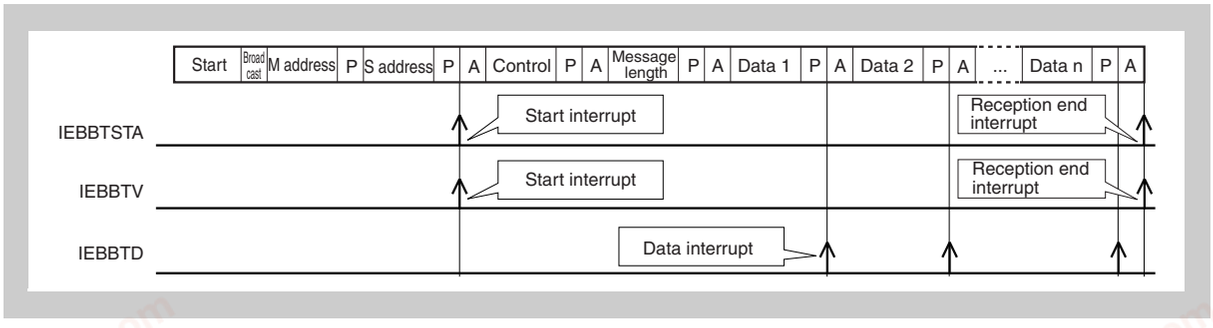


Figure 22-38 Interrupt occurrence timing

chanhvinh.com

(3) Interrupt servicing examples

(a) Start interrupt CPU processing flow example

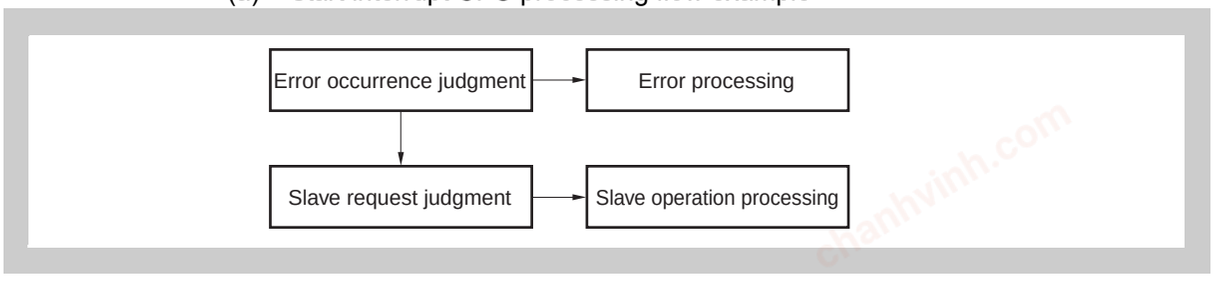


Figure 22-39 Start interrupt CPU processing flow example

(b) Reception completion interrupt (IEBBTV, IEBBTSTA) CPU processing flow example

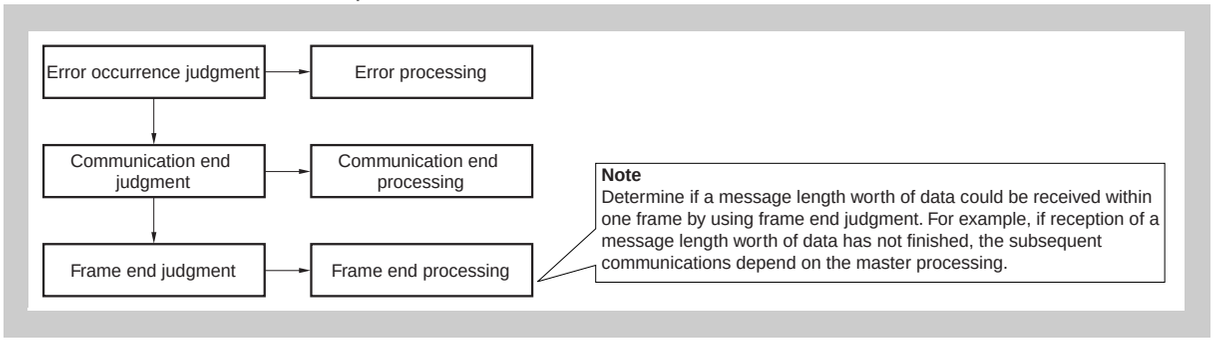


Figure 22-40 Reception completion interrupt (IEBBTV, IEBBTSTA) CPU processing flow example

(c) Data interrupt CPU processing example

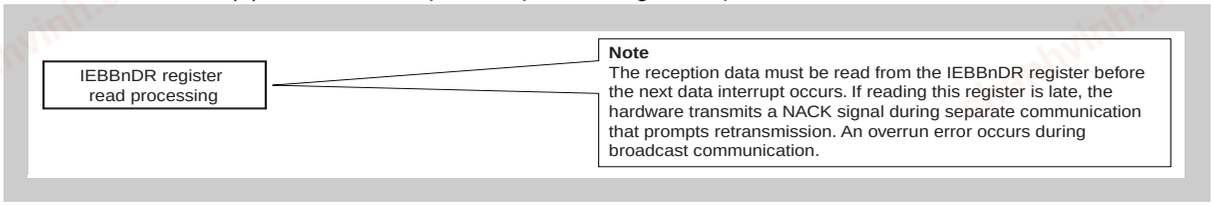


Figure 22-41 Data interrupt CPU processing

chanhvinh.com

22.5.10 Slave reception (FIFO mode)

The unit receives data and commands from the master unit as a slave.

(1) Register settings

After specifying the initial settings in 22.5.2 Initial settings, set up the registers below before starting communication.

Table 22-55 Communication startup processing

Register name	Function	Example
IEBBnBCR	Communication startup processing	88H

(2) Interrupt occurrence timing

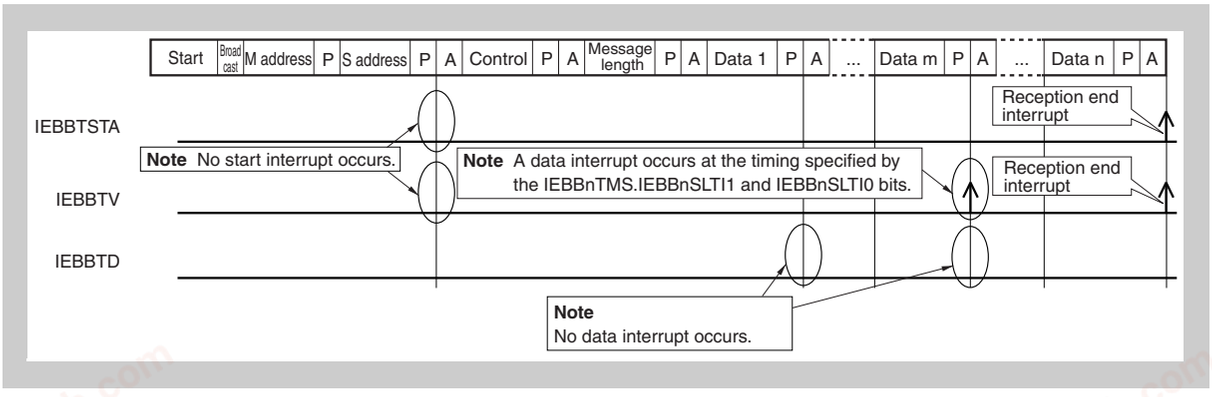


Figure 22-42 Interrupt occurrence timing

(3) Interrupt servicing examples

(a) Reception completion interrupt (IEBBTV, IEBBTSTA) CPU processing flow example

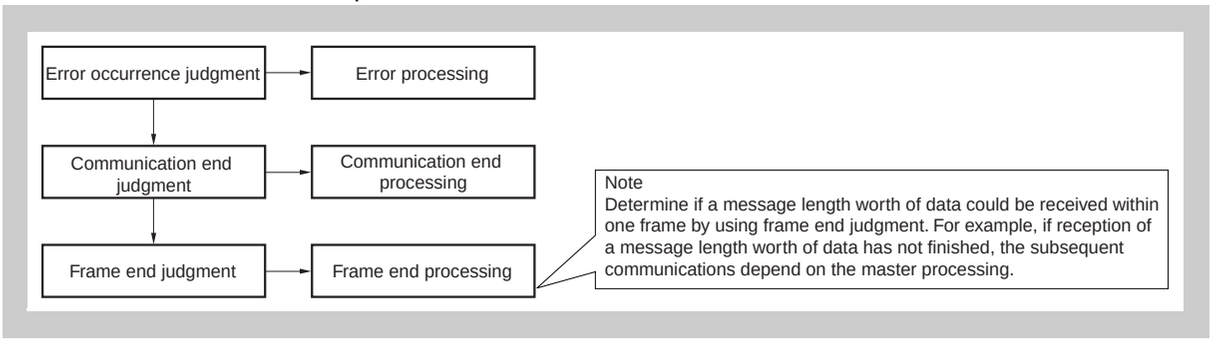


Figure 22-43 Reception completion interrupt (IEBBTV, IEBBTSTA) CPU processing flow example

chanhvinh.com

22.6 Setup Procedures

22.6.1 Master transmission (single mode)

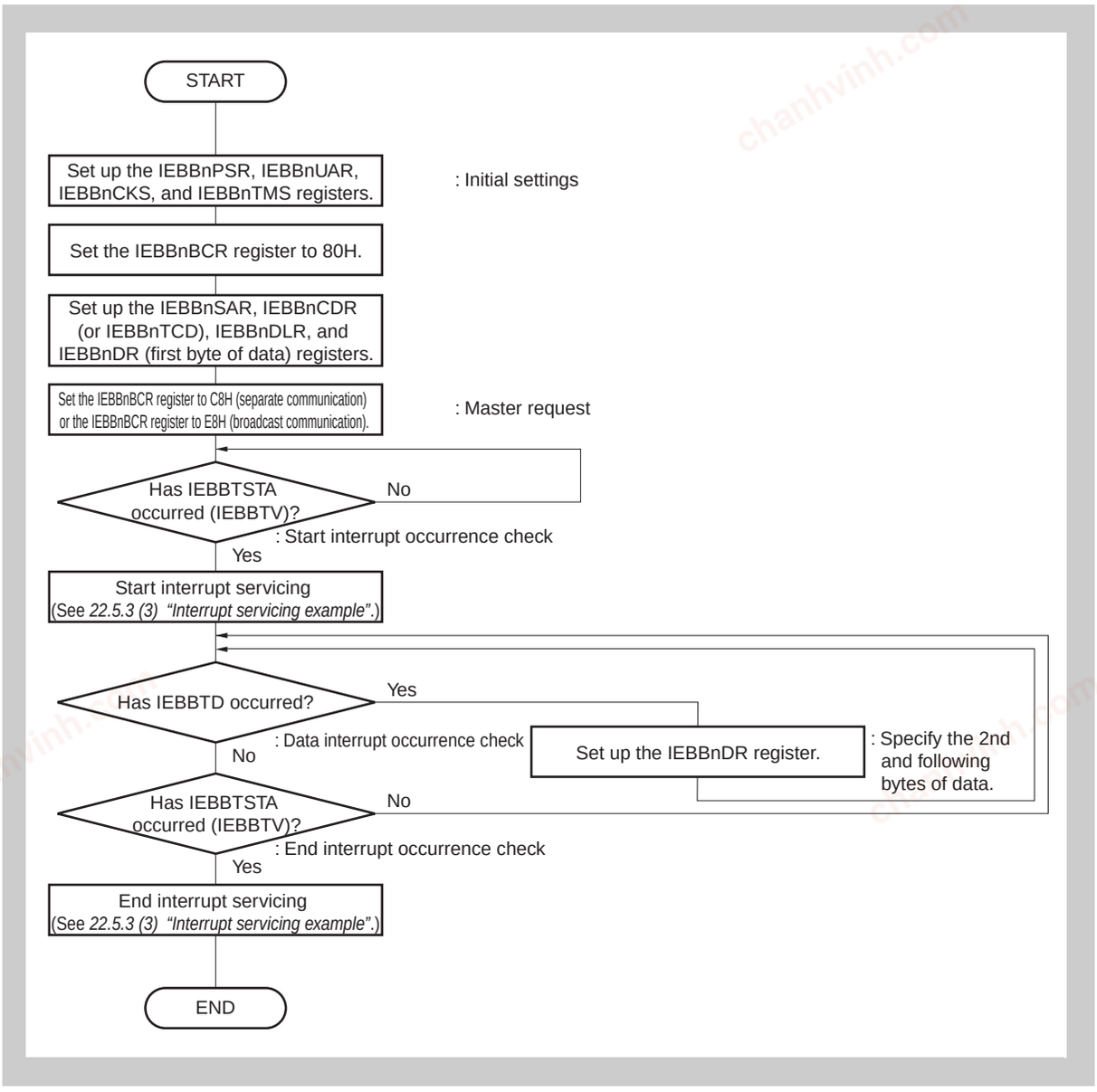


Figure 22-44 Master transmission (single mode)

chanhvinh.com

22.6.2 Master transmission (FIFO mode)

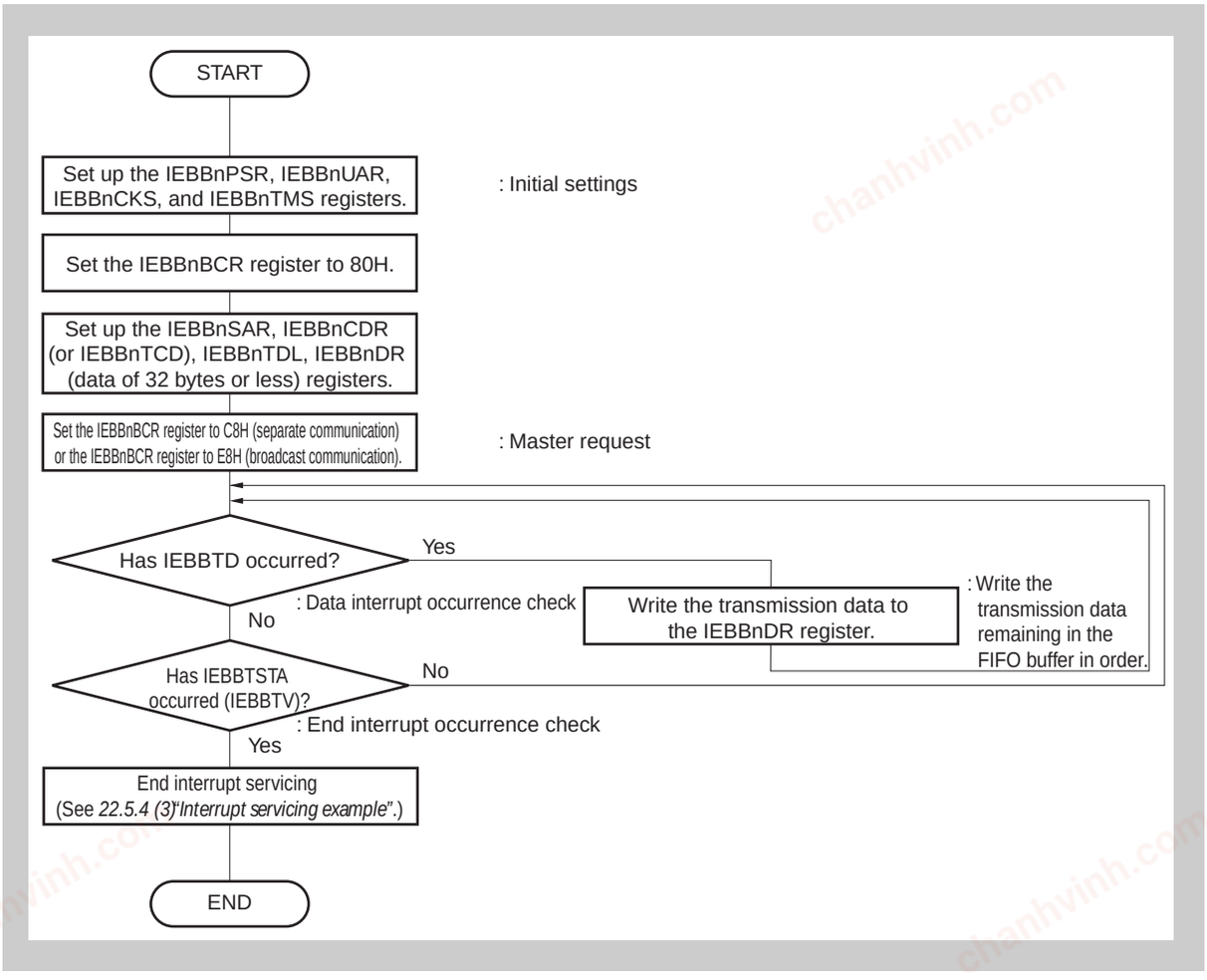


Figure 22-45 Master transmission (FIFO mode)

chanhvinh.com

22.6.3 Master reception (single mode)

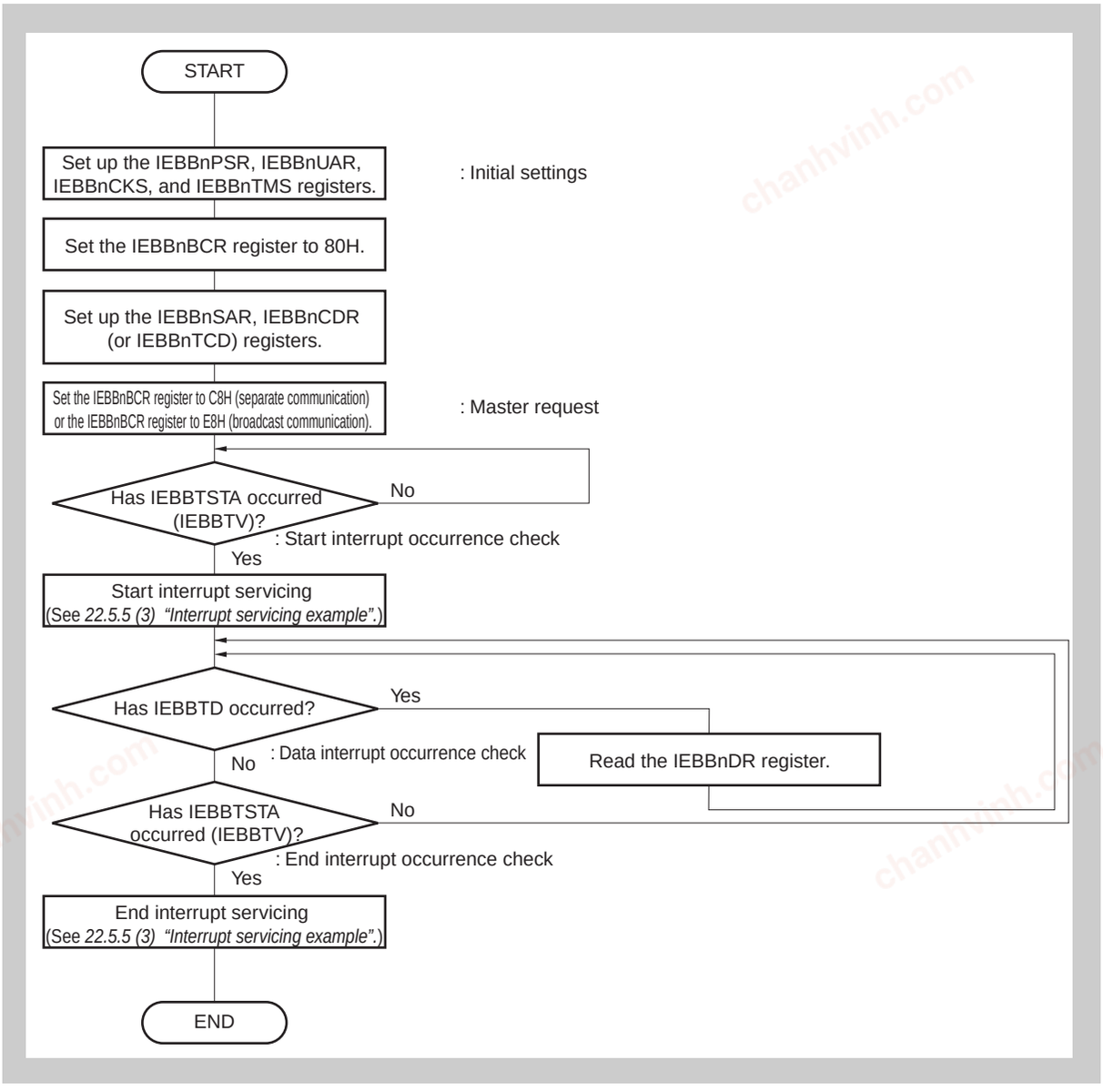


Figure 22-46 Master reception (single mode)

chanhvinh.com

22.6.4 Master reception (FIFO mode)

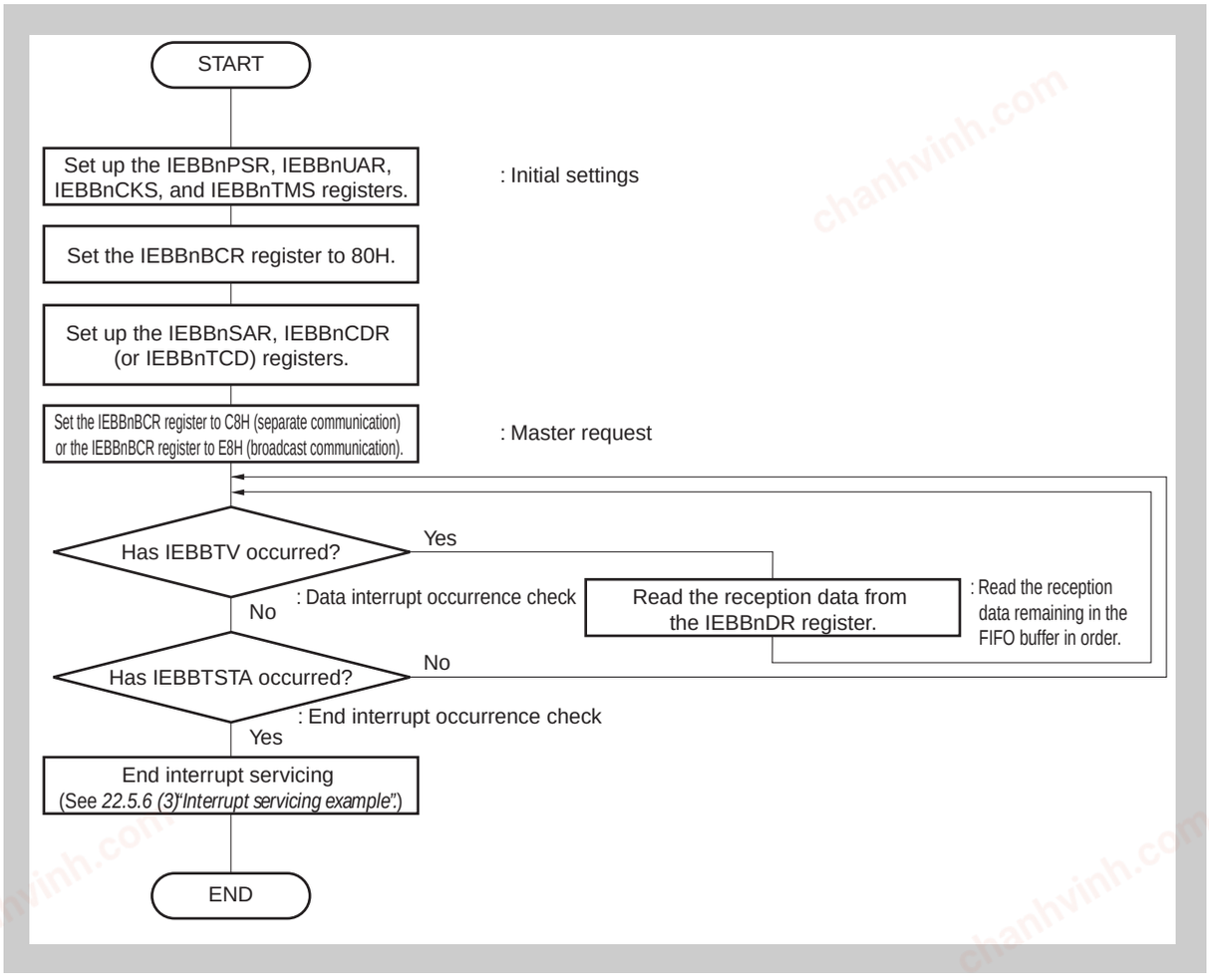


Figure 22-47 Master reception (FIFO mode)

chanhvinh.com

22.6.5 Slave transmission (single mode)

(1) When the control bit 3H or 7H is received

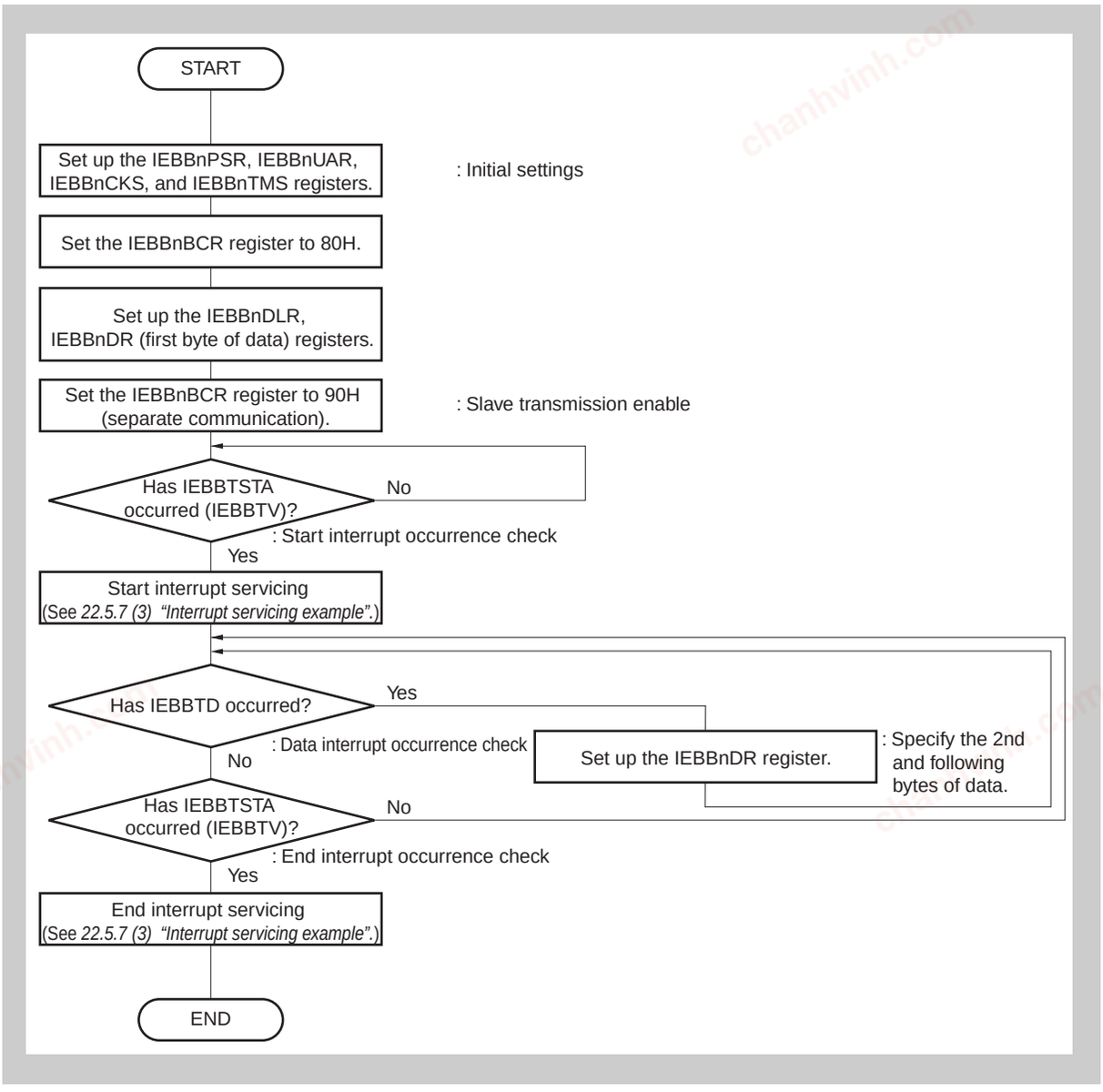


Figure 22-48 Slave transmission (single mode): When the control bit 3H or 7H is received

chanhvinh.com

(2) When the control bit 0H or 6H is received (or when 4H or 5H is received from the locked master while the unit is locked)

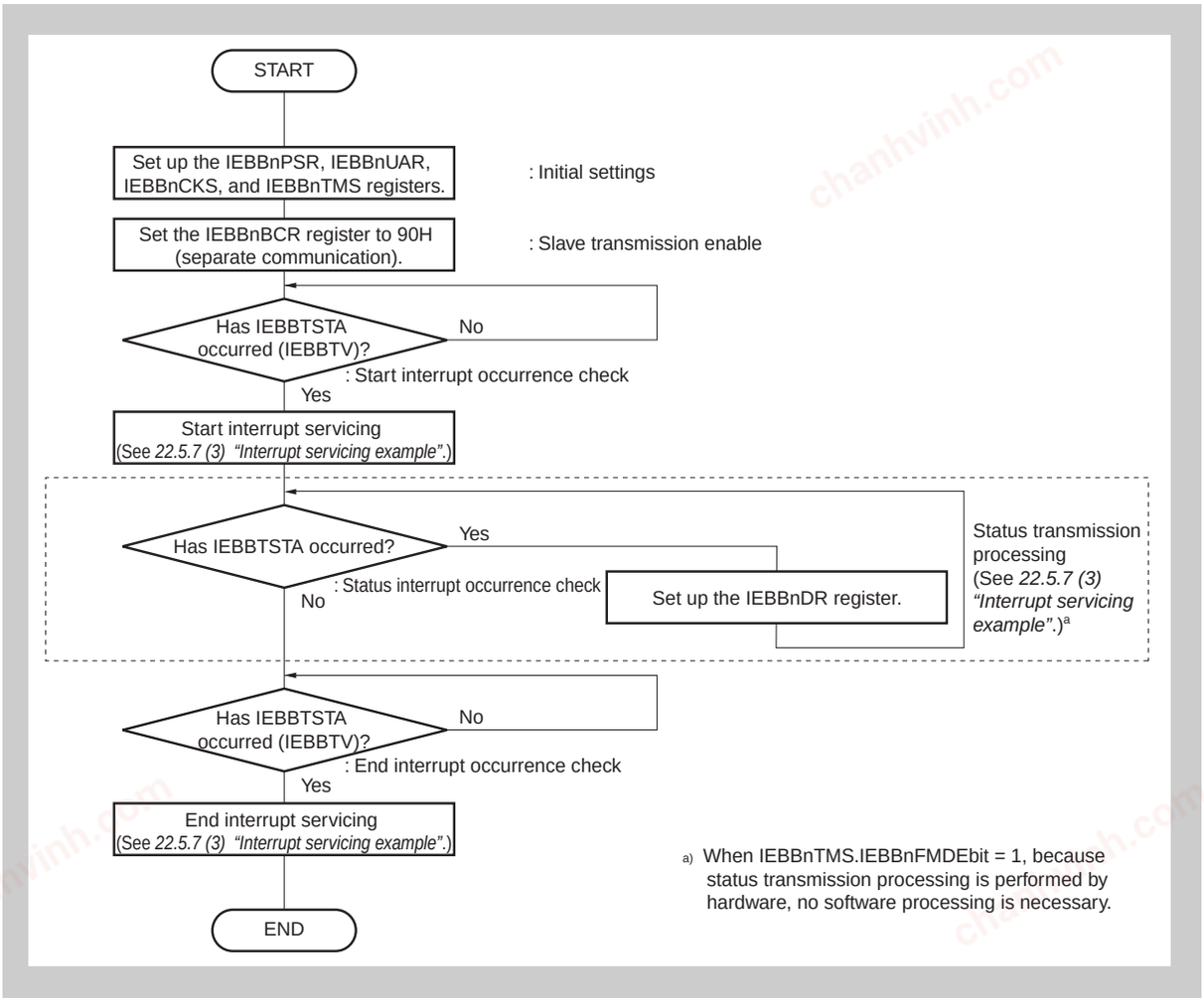


Figure 22-49 Slave transmission (single mode): When the control bit 0H or 6H is received (or when 4H or 5H is received from the locked master while the unit is locked)

chanhvinh.com

(3) When the control bit 0H, 4H, or 5H, which is addressed to the unit, is received from a unit other than the locked master while the unit is locked

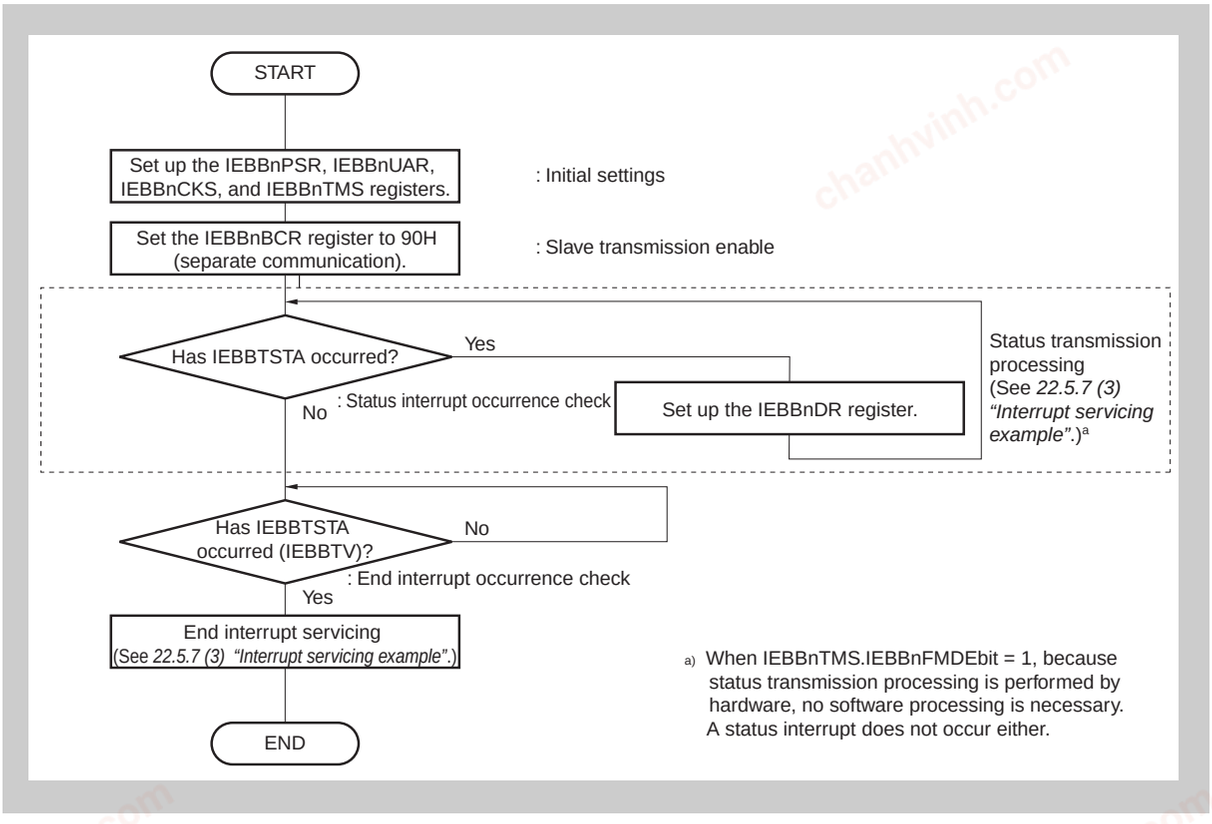


Figure 22-50 Slave transmission (single mode): When the control bit 0H, 4H, or 5H, which is addressed to the unit, is received from a unit other than the locked master while the unit is locked

chanhvinh.com

22.6.6 Slave transmission (FIFO mode)

(1) When the control bit 3H or 7H is received

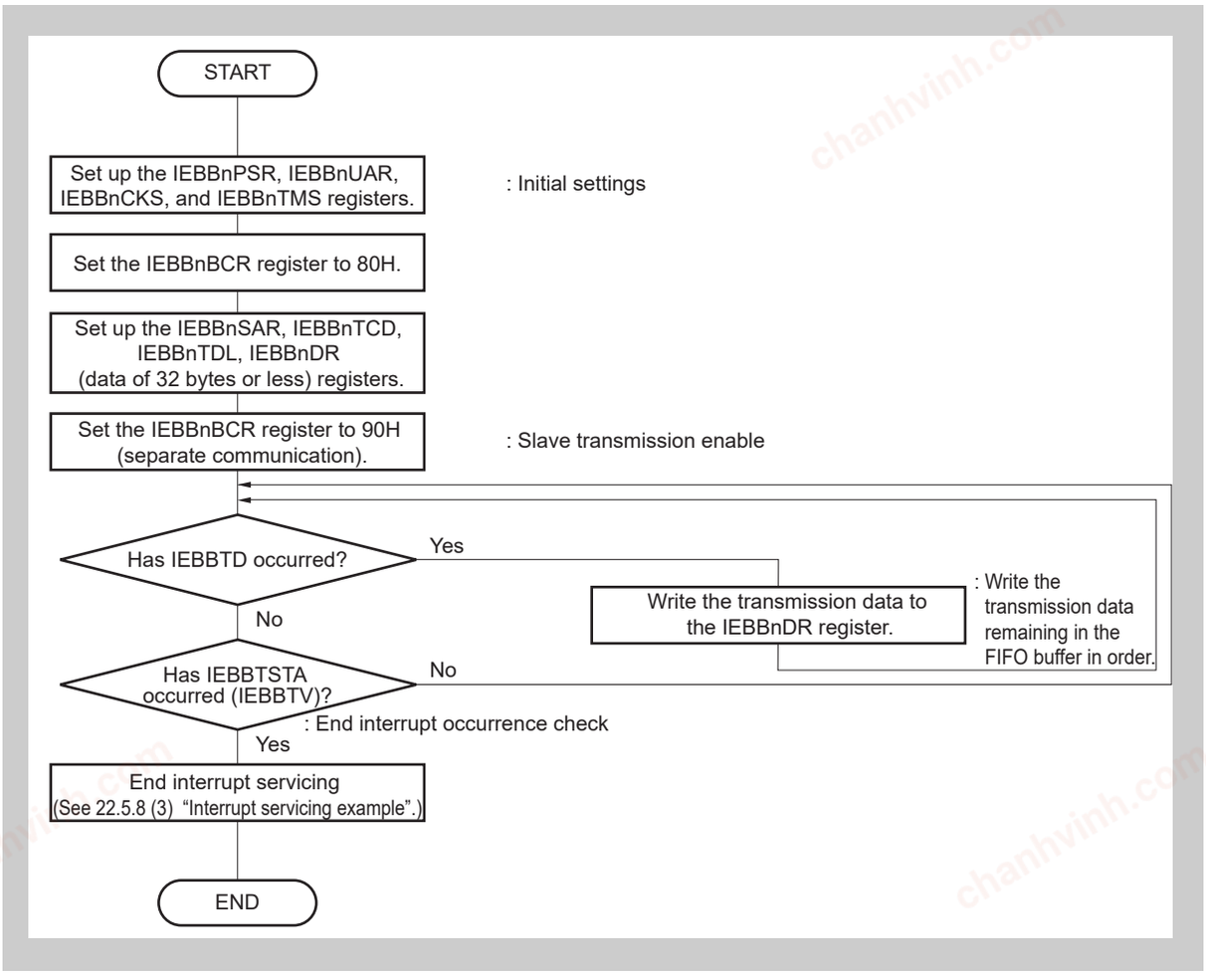


Figure 22-51 Slave transmission (FIFO mode): When the control bit 3H or 7H is received

chanhvinh.com

(2) When the control bit 0H or 6H is received (or when 4H or 5H is received from the locked master while the unit is locked)

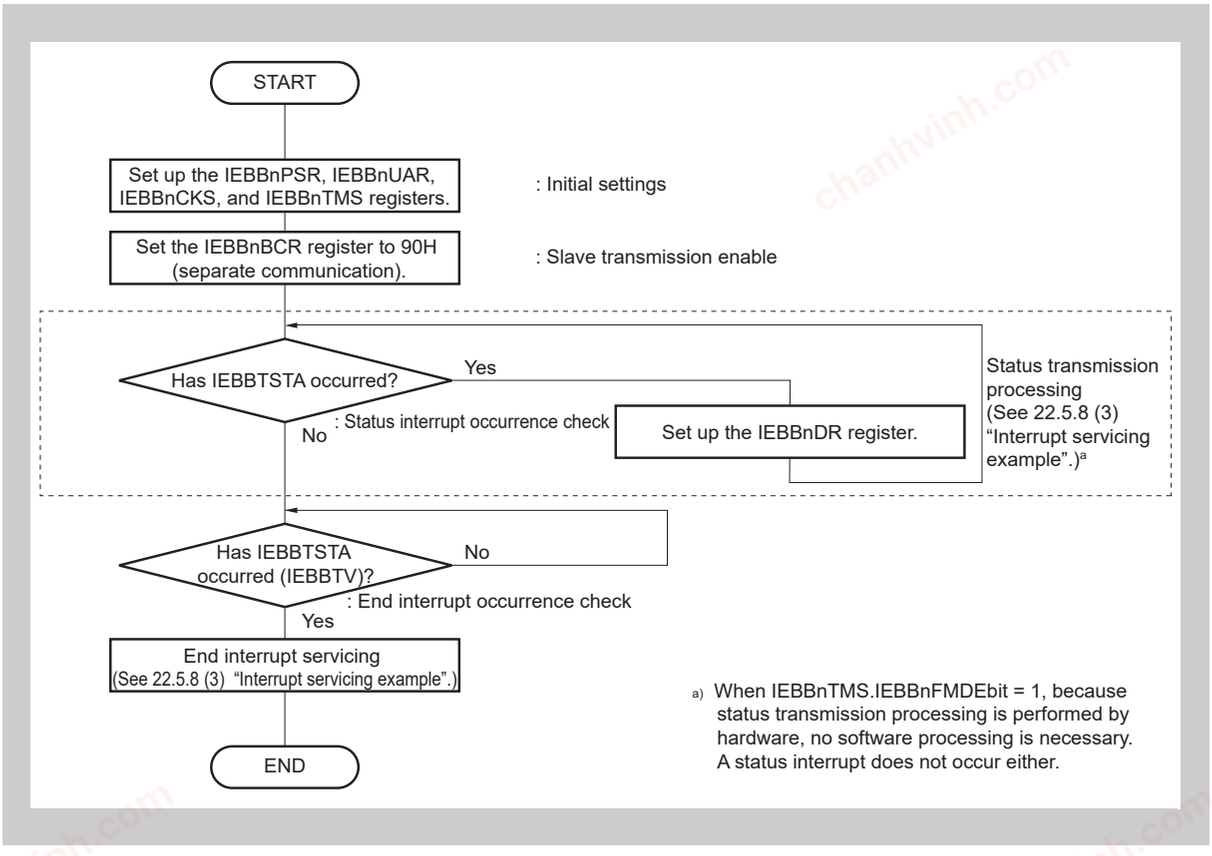


Figure 22-52 Slave transmission (FIFO mode): When the control bit 0H or 6H is received (or when 4H or 5H is received from the locked master while the unit is locked)

chanhvinh.com

(3) When the control bit 0H, 4H, or 5H, which is addressed to the unit, is received from a unit other than the locked master while the unit is locked

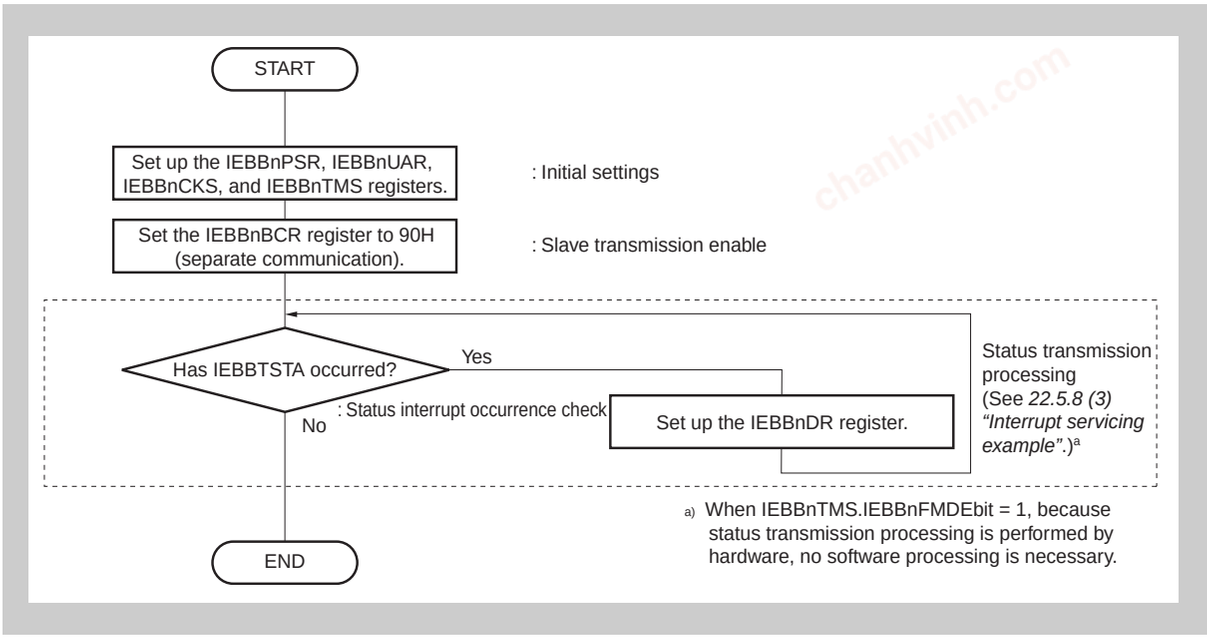


Figure 22-53 Slave transmission (FIFO mode): When the control bit 0H, 4H, or 5H, which is addressed to the unit, is received from a unit other than the locked master while the unit is locked

chanhvinh.com

22.6.7 Slave reception (single mode)

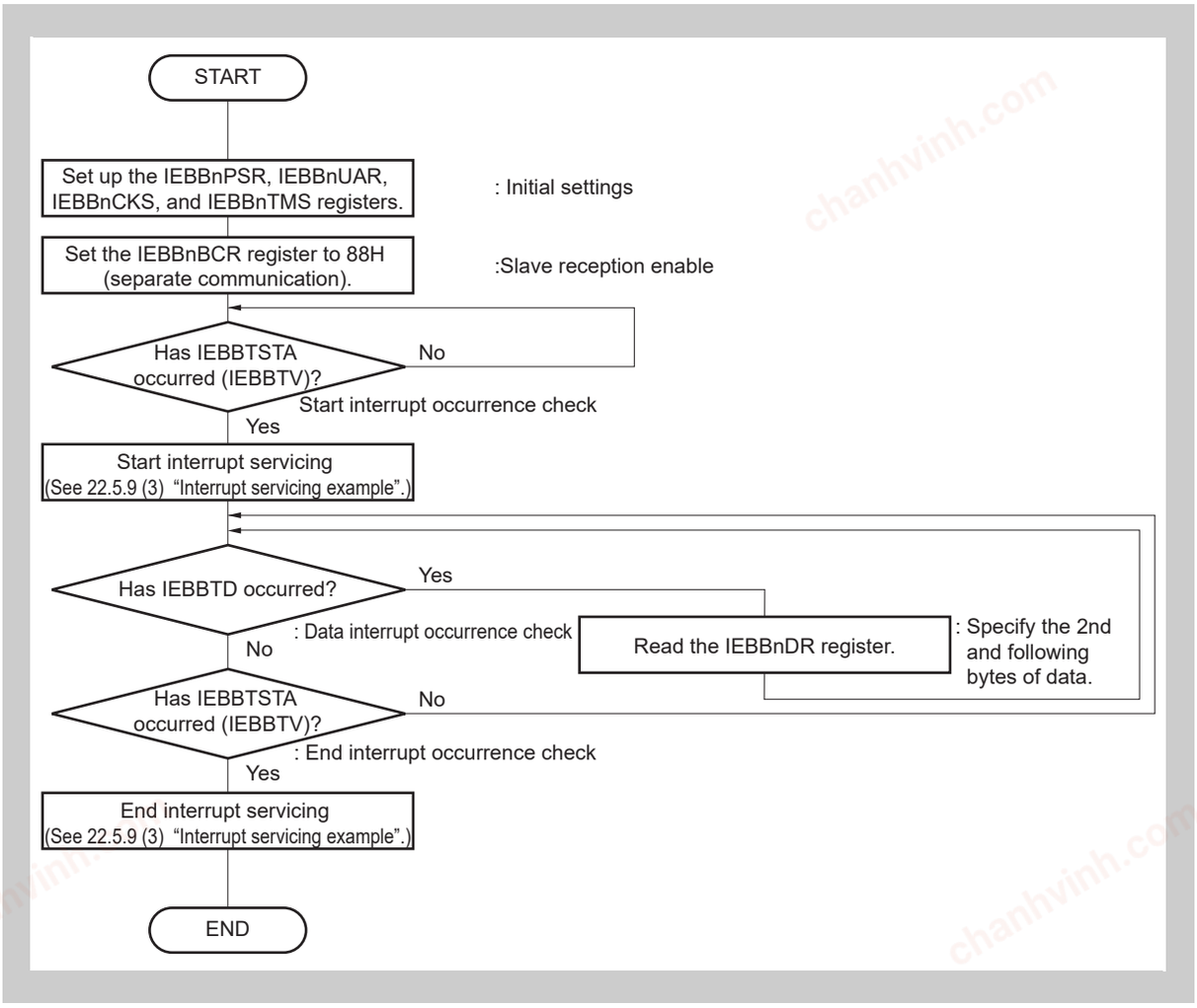


Figure 22-54 Slave reception (single mode)

chanhvinh.com

22.6.8 Slave reception (FIFO mode)

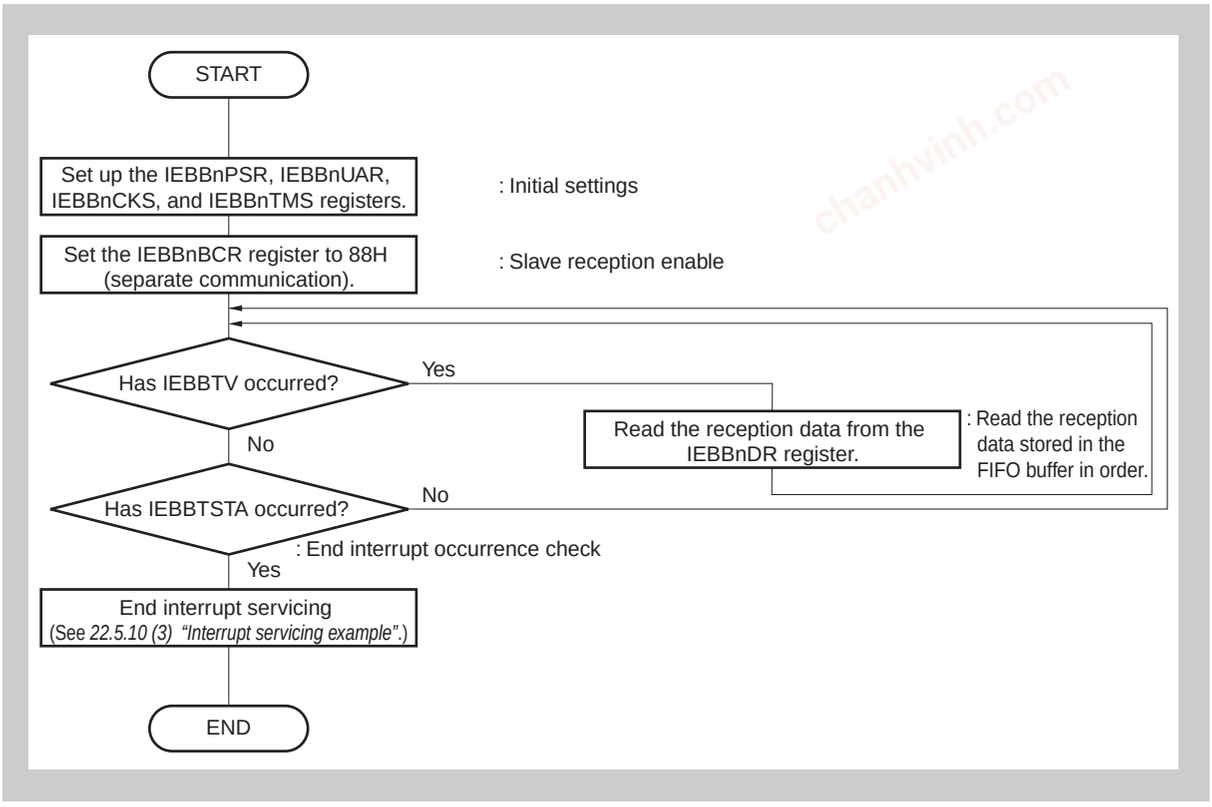


Figure 22-55 Slave reception (FIFO mode)

chanhvinh.com

22.7 Functions

22.7.1 IEBus communication protocol

The communication protocol of the IEBus is as follows.

(1) Multi-task mode

All the units connected to the IEBus can transfer data to the other units.

(2) Broadcast communication

Communication between one unit and multiple units can be performed as follows.

- Group broadcast communication: Broadcast communication to group units
- All-unit broadcast communication: Broadcast communication to all units

(3) Effective transmission speed

The effective transfer rate is in communication mode 1 or communication mode 2. (This product does not support mode 0 for the effective transfer rate.)

- Communication mode 1: Approx. 18 kbps
- Communication mode 2: Approx. 27 kbps

Caution Different modes (communication mode 1, communication mode 2) must not be mixed on one IEBus.

(4) Communication mode

Data transfer is executed in half-duplex asynchronous communication mode.

(5) Access control: CSMA/CD (Carrier Sense Multiple Access with Collision Detection)

The priority of the IEBus is as follows:

1. Broadcast communication takes precedence over individual communication (communication from one unit to another).
2. The lower master address takes precedence.

(6) Communication scale

The communication scale of IEBus is as follows:

- Number of units: 50 maximum
- Cable length: 150 m maximum (when twisted pair cable is used)

Caution The communication scale in an actual system differs depending on the characteristics of the cables, etc., constituting the IEBus driver/receiver and IEBus.

chanhvinh.com

22.7.2 Determination of bus mastership (arbitration)

An operation to occupy the bus is performed when a unit connected to the IEBus controls the other units. This operation is called arbitration.

When multiple units simultaneously start transmission, arbitration is used to grant one of the units permission to occupy the bus.

Because only one unit is granted bus mastership as a result of arbitration, the priority conditions of the bus are predetermined as follows.

Caution Bus mastership is canceled if communication is aborted.

(1) Priority by communication type

Broadcast communication (communication from one unit to multiple units) takes precedence over normal communication (communication from one unit to another).

(2) Priority by master address

If the communication type is the same, communication with the lower master address takes precedence.

A master address consists of 12 bits, with unit 000H having the highest priority and unit FFFH having the lowest priority.

22.7.3 Communication mode

The IEBus has three communication modes, each of which has a different transfer rate. This module supports communication modes 1 and 2. The transfer rate and the maximum number of transfer bytes per communication frame in communication modes 1 and 2 are shown below.

Table 22-56 Transfer rate and maximum number of transfer bytes in each communication mode

Communication mode	Maximum number of transfer bytes (bytes/frame)	Effective transfer rate[a]
1	32 bytes/frame	Approx. 18 kbps
2	128 bytes/frame	Approx. 27 kbps

a) Effective transfer rate when the maximum number of transfer bytes is transmitted

Select the communication mode for each unit connected to the IEBus before starting communication. If the communication mode of the master unit and that of the partner unit (slave unit) are not the same, communication is not correctly executed.

chanhvinh.com

22.7.4 Communication address

For the IEBus, each unit is assigned a specific 12-bit address. This communication address consists of the following identification numbers.

- Higher 4 bits: Group number (number to identify the group to which each unit belongs)
- Lower 8 bits: Unit number (number to identify each unit in a group)

22.7.5 Broadcast communication

Normally, transmission or reception is performed between the master unit and its partner slave unit on a one-to-one basis. During broadcast communication, however, multiple slave units exist and the master unit executes transmission to these slave units. Because multiple slave units exist, the NACK signal is returned by the communicating slave unit as an acknowledge bit.

Whether broadcast communication or normal communication is to be executed is selected by the broadcast bit. (For details about this bit, see 22.7.6 (2) Broadcast bit.)

Broadcast communication is classified into two types: group-unit broadcast communication and all-unit broadcast communication. Group-unit broadcast and all-unit broadcast are identified by the value of the slave address. (For the slave address, see 22.7.6 (4) Slave address field.)

(1) Group-unit broadcast communication

Broadcast communication is performed to the units in a group identified by the group number indicated by the higher 4 bits of the communication address.

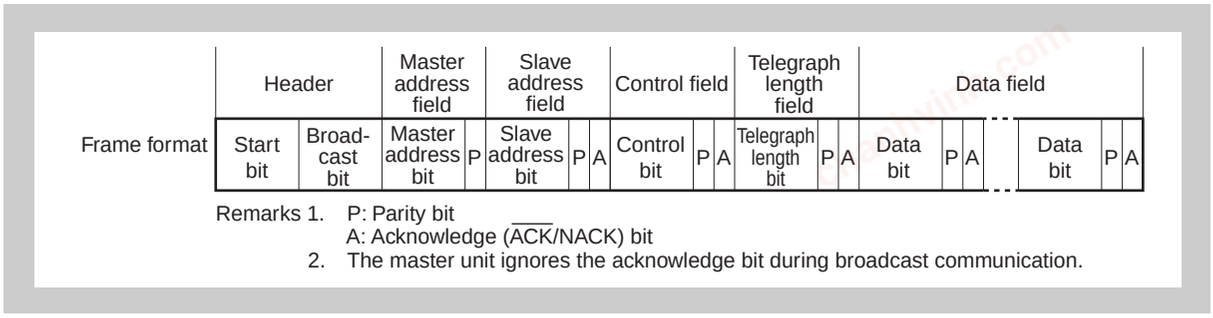
(2) All-unit broadcast communication

Broadcast communication is performed to all the units, regardless of the value of the group number.

chanhvinh.com

22.7.6 IEBus transfer format

The IEBus transfer signal format is shown in Figure 22-56.



Remarks 1. P: Parity bit
A: Acknowledge ($\overline{ACK}$/NACK) bit
2. The master unit ignores the acknowledge bit during broadcast communication.

Figure 22-56 IEBus transfer signal format

(1) Start bit

The start bit is a signal that informs the other units of the start of a data transfer.

The unit that is to start a data transfer outputs a low-level signal (start bit) for a specific time, and then starts outputting the broadcast bit.

If another unit has already output its start bit when one unit is to output the start bit, this unit does not output the start bit and instead waits for completion of output of the start bit by the other unit. When the output of the start bit by the other unit is complete, the unit starts outputting the broadcast bit in synchronization with the completion of the start bit output by the other unit.

The units other than the one that started communication detect this start bit, and enter the reception status.

(2) Broadcast bit

This bit indicates whether the master selects one slave (individual communication) or multiple slaves (broadcast communication) as the other party of communication.

When the broadcast bit is 0, it indicates broadcast communication. When it is 1, individual communication is indicated. Broadcast communication is classified into two types: group-unit communication and all-unit communication. These communication types are identified by the value of the slave address. (For the slave address, see 22.7.6 (4) Slave address field.)

Because multiple slave units exist as a partner slave unit of communication in the case of broadcast communication, the NACK signal is returned as an acknowledge bit in each field subsequent to the master address field.

If multiple units start transmitting a communication frame at the same time, broadcast communication takes precedence over individual communication, and wins in arbitration.

If one unit occupies the bus as the master, the value set to the broadcast request flag (the IEBBnBCR.IEBBnALRQ bit) is output.

chanhvinh.com

(3) Master address field

The master address field is output by the master to inform a slave of the master's address.

The configuration of the master address field is shown in Figure 22-57.

If multiple units start transmitting the broadcast bit at the same time, the master address field makes a judgment of arbitration.

The master address field compares the data it outputs with the data on the bus each time it has output one bit. If the master address output by the master address field is found to differ from the data on the bus as a result of comparison, it is assumed that the master has lost arbitration.

As a result, the master stops transmission and enters the reception status. Because the IEBus is configured of wired AND, the unit having the smallest master address of the units participating in arbitration (arbitration masters) wins arbitration.

After a 12-bit master address has been output, only one unit remains in the transmission status as one master unit.

Next, this master unit outputs a parity bit, determines the master address of other unit, and starts outputting a slave address field.

If one unit occupies the bus as the master, the address specified by the IEBBnUAR register is output.

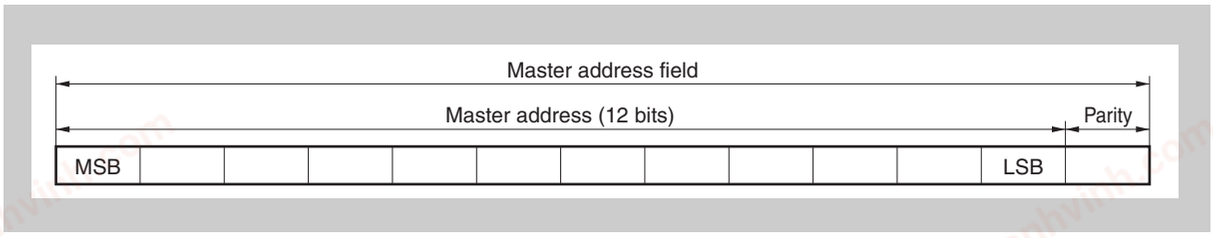


Figure 22-57 Master address field

chanhvinh.com

(4) Slave address field

The master outputs the address of the unit with which it is to communicate.

The configuration of the slave address field is shown in Figure 22-58.

A parity bit is output after a 12-bit slave address has been transmitted to prevent the wrong slave address from being received by mistake. Next, the master unit detects an $\overline{\text{ACK}}$ signal from the slave unit to confirm that the slave unit exists on the bus. The master unit starts outputting the control field after detecting the $\overline{\text{ACK}}$ signal. During broadcast communication, however, the master does not confirm the acknowledge bit and instead starts outputting the control field.

The slave unit outputs the $\overline{\text{ACK}}$ signal if its slave address matches and if the slave detects that the parities of both the master address and slave address are even. The slave unit judges that the master address or slave address has not been correctly received and outputs the NACK signal if the parities are odd. At this time, the master unit is in the standby (monitor) status, and communication ends.

During broadcast communication, the slave address is used to identify group-unit broadcast or all-unit broadcast, as follows:

If the slave address is FFFH: All-unit broadcast communication

If the slave address is not FFFH: Group-unit broadcast communication

Note The group No. during group-unit broadcasting communication is the value of the higher 4 bits of the slave address.

If one unit occupies the bus as the master, the address specified by the IEBBnSAR register is output.

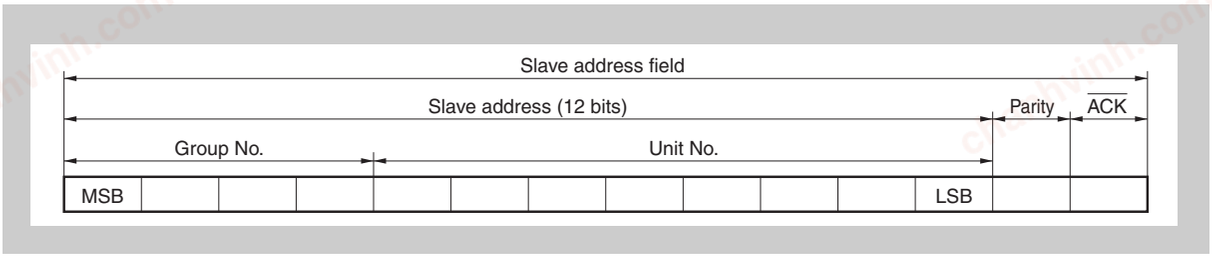


Figure 22-58 Slave address field

chanhvinh.com

(5) Control data field

The master uses this field to output the operation it requires the slave to perform.

The configuration of the control field is shown in Figure 22-59.

If the parity following the control bit is even and the slave unit can execute the function required by the master unit, the slave unit outputs an $\overline{\text{ACK}}$ signal and starts outputting the message length field. If the slave unit cannot execute the function required by the master unit even if the parity is even, or if the parity is odd, the slave unit outputs the NACK signal, and returns to the standby (monitor) status.

The master unit starts outputting the message length field after detecting the $\overline{\text{ACK}}$ signal.

If the master detects the NACK signal, the master unit enters the standby status, and communication ends. During broadcast communication, however, the master unit does not confirm the acknowledge bit and starts outputting the message length field.

If one unit occupies the bus as the master, the value set to the IEBBnTCD register is output.

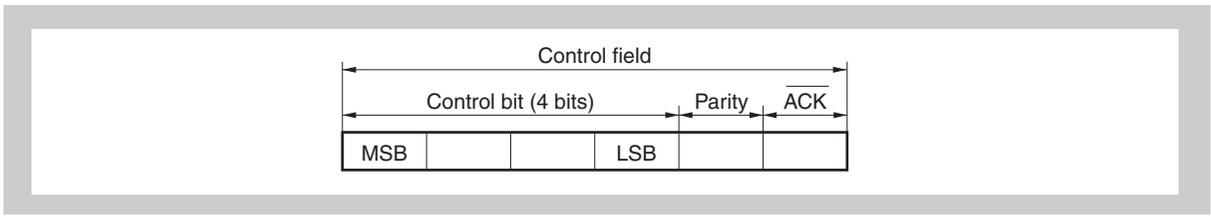


Figure 22-59 Control field

The contents of the control bits are shown below.

Table 22-57 Control bit contents

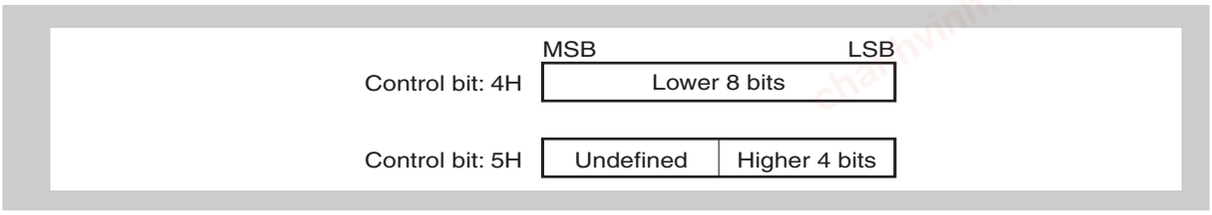
Bit 3[a]	Bit 2	Bit 1	Bit 0	Function
0	0	0	0	Read slave status
0	0	0	1	Undefined
0	0	1	0	Undefined
0	0	1	1	Read data and lock[b]
0	1	0	0	Read lock address (lower 8 bits)[c]
0	1	0	1	Lock address reading (higher 4 bits)[c]
0	1	1	0	Slave status reading and unlocking[b]
0	1	1	1	Read data
1	0	0	0	Undefined
1	0	0	1	Undefined
1	0	1	0	Command writing and locking[b]
1	0	1	1	Data writing and locking[b]
1	1	0	0	Undefined
1	1	0	1	Undefined
1	1	1	0	Write command
1	1	1	1	Write data

chanhvinh.com

a) The message length bit of the message length field and data transfer direction of the data field change as follows depending on the value of bit 3 (MSB).
If bit 3 is 1: Transfer from master unit to slave unit
If bit 3 is 0: Transfer from slave unit to master unit

b) This is a control bit that specifies locking or unlocking. (For details, see 22.7.7 (4) Locking and unlocking.)

c) The lock address is transferred in 1-byte (8-bit) units and is configured as follows:



If the control bit received from the master unit is not as shown in Table 22-58, the unit locked by the master unit rejects acknowledging the control bit, and outputs the NACK signal.

Table 22-58 Control field for locked slave unit

Bit 3	Bit 2	Bit 1	Bit 0	Function
0	0	0	0	Read slave status
0	1	0	0	Lock address reading (lower 8 bits)
0	1	0	1	Lock address reading (higher 4 bits)

In addition, units for which locking is not set up by the master unit reject acknowledgment and output a NACK signal when the control data shown in Table 22-59 is acknowledged.

Table 22-59 Control field for unlocked slave unit

Bit 3	Bit 2	Bit 1	Bit 0	Function
0	1	0	0	Lock address reading (lower 8 bits)
0	1	0	1	Lock address reading (higher 4 bits)

chanhvinh.com

Table 22-60 Control field $\overline{\text{ACK}}$ signal response conditions (when the received control data is 0H, 3H, 4H, 5H, 6H, or 7H)

Communication target (IEBBnUSR. IEBBnSRQF bit) Slave specification = 1 No specification = 0	Lock status (IEBBnUSR. IEBBnLCKF bit) Lock = 1 No lock = 0	Master unit judgment (IEBBnPAR register match) Lock request unit = 1 Other = 0	Slave transmission enabled (IEBBnBCR. IEBBnSTXE bit)	Slave reception enabled (IEBBnBCR. IEBBnSRXE bit)	Received control data					
					0H	3H	4H	5H	6H	7H
1	0	don't care	0	don't care	A	N	N	N	A	N
			1		A	A	N	N	A	A
	1	0	don't care		A	N	A	A	N	N
		1	0		A	N	A	A	A	N
			1		A	A	A	A	A	A
Other than the above					N					

Note A: Slave transmission is performed. (The $\overline{\text{ACK}}$ signal is returned.)

N: Slave transmission is not performed. (The NACK signal is returned.)

Caution If the received control data is other than the data shown in the above table, N is unconditionally assumed. (Slave transmission is not performed (and the NACK signal is returned).)

Table 22-61 Control field $\overline{\text{ACK}}$ signal response conditions (when the received control data is AH, BH, EH, or FH)

Communication target (IEBBnUSR. IEBBnSRQF bit) Slave specification = 1 No specification = 0	Lock status (IEBBnUSR. IEBBnLCKF bit) Lock = 1 No lock = 0	Master unit judgment (IEBBnPAR register match) Lock request unit = 1 Other = 0	Slave transmission enabled (IEBBnBCR. IEBBnSTXE bit)	Slave reception enabled (IEBBnBCR. IEBBnSRXE bit)	Received control data			
					AH	BH	EH	FH
1	0	don't care	don't care	1	A			
	1	1						
Other than the above					N			

Note A: Slave transmission is performed. (The $\overline{\text{ACK}}$ signal is returned.)

N: Slave transmission is not performed. (The NACK signal is returned.)

Caution If the received control data is other than the data shown in the above table, N is unconditionally assumed. (Slave transmission is not performed (and the NACK signal is returned).)

chanhvinh.com

(6) Message length field

This field is output by the transmission side to inform the reception side of the number of bytes of the transmit data.

The configuration of the message length field is shown in Figure 22-60.

Table 22-62 shows the relationship between the message length bit and the number of transmission data bytes.

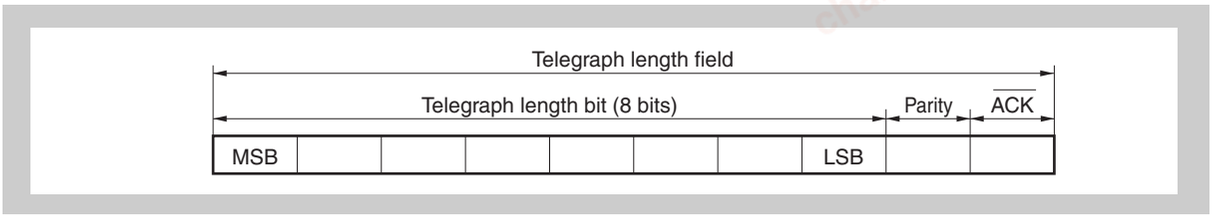


Figure 22-60 Message length field

Table 22-62 Contents of the message length bit

Message length bit (hexadecimal)	Number of transmission data bytes
01H	1 byte
02H	2 bytes
...	...
FFH	255 bytes
00H	256 bytes

The operation of the message length field differs depending on whether the master transmits data (when control bit 3 is 1) or receives data (when control bit 3 is 0).

(a) During master transmission

The message length bit and parity bit are output by the master unit and the synchronization signals of bits are output by the master unit. When the slave unit detects that the parity is even, it outputs the $\overline{\text{ACK}}$ signal, and starts outputting the data field. During broadcast communication, however, the slave unit outputs the NACK signal.

If the parity is odd, the slave unit judges that the message length bit has not been correctly received, outputs the NACK signal, and returns to the standby (monitor) status. At this time, the master unit also returns to the standby status, and communication ends.

(b) Master reception

The message length bit and parity bit are output by the slave unit and the synchronization signals of bits are output by the master unit. If the master unit detects that the parity bit is even, it outputs the $\overline{\text{ACK}}$ signal.

If the parity bit is odd, the master unit judges that the message length bit has not been correctly received, outputs the NACK signal, and returns to the standby status. At this time, the slave unit also returns to the standby status, and communication ends.

chanhvinh.com

(7) Data field

This is data output by the transmission side.

The master unit transmits or receives data to or from a slave unit by using the data field.

The configuration of the data field is shown below.

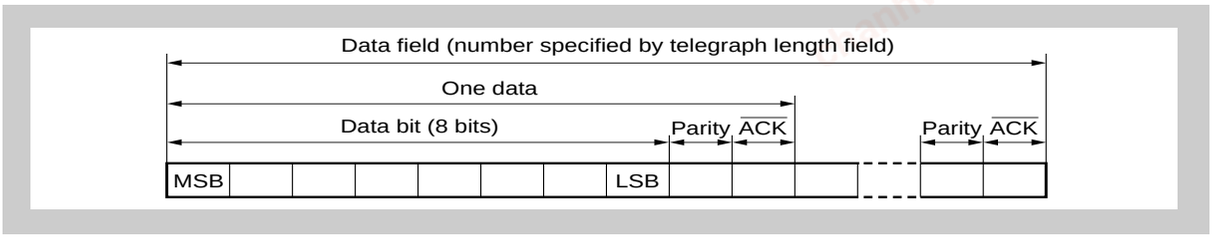


Figure 22-61 Data field

Following the data bit, the parity bit and acknowledge bit are output by the master unit and slave unit, respectively.

Use broadcast communication only when the master unit transmits data. At this time, the acknowledge bit is ignored.

The operation differs as follows depending on whether the master transmits or receives data.

(a) During master transmission

When the master units writes data to a slave unit, the master unit transmits the data bit and parity bit to the slave unit. If the parity is even and the received data is not stored in the IEBBnDR register when the slave unit has received the data bit and parity bit, the slave unit outputs an $\overline{\text{ACK}}$ signal. If the parity is odd or the received data is stored in the IEBBnDR register, the slave unit rejects receiving the data, and outputs the NACK signal.

If the slave unit outputs the NACK signal, the master unit transmits the same data again. This operation continues until the master detects the $\overline{\text{ACK}}$ signal from the slave unit, or the data exceeds the maximum number of transmit bytes.

If there is more data and the maximum number of transmission bytes is not exceeded when the parity is even and when the slave unit outputs the $\overline{\text{ACK}}$ signal, the master unit transmits the next data.

During broadcast communication, the slave unit outputs the NACK signal, and the master unit transfers 1 byte of data at a time. If the parity is odd or the IEBBnDR register is storing received data after the slave unit receives the data bit and parity bit during broadcast communication, the slave unit judges that reception has not been performed correctly, and stops reception.

chanhvinh.com

(b) Master reception

When the master unit reads data from a slave unit, the master unit outputs a sync signal corresponding to all the read bits.

The slave unit outputs the contents of the data and parity bits to the bus in response to the sync signal from the master unit.

The master unit reads the data and parity bits output by the slave unit, and checks the parity.

If the parity is odd or the IEBBnDR register is storing received data, the master unit rejects accepting the data, and outputs the NACK signal. If the maximum number of transmission bytes is within the value that can be transmitted in one communication frame, the master unit rereads the same data.

If the parity is even and the IEBBnDR register is not storing received data, the master unit accepts the data and outputs the $\overline{\text{ACK}}$ signal. If the maximum number of transmission bytes is within the value that can be transmitted in one frame, the master unit reads the next data.

Caution During broadcast communication, do not perform master reception. If you do this, the slave unit cannot be defined and data transfers cannot be performed correctly.

Note that, due to the IEBBn specifications, overrun errors can occur. Therefore, even if reading the IEBBnDR register is late during individual communication and the system has reached the timing for receiving the next data (the overrun status), data can be retransmitted from the master unit by returning a NACK signal, which makes it possible to buy time for reading the IEBBnDR register. However, during broadcast communication, because no $\overline{\text{ACK}}$ signal is output from the slave unit and the master unit ignores $\overline{\text{ACK}}$ signals, even if reading the IEBBnDR register is late, no data is retransmitted from the master. Therefore, for IEBBn, if an overrun occurs during broadcast communication, normal reception is not possible, an overrun error occurs, and an interrupt request (for a communication error) is output.

(8) Parity bit

The parity bit is used to make sure that the transmission data has no error.

The parity bit is appended to each data of the master address, slave address, control, message length, and data bits.

The parity is an even parity. If the number of data bits that are ‘1’ is odd, the parity bit is ‘1’. If the number of data bits that are ‘1’ is even, the parity bit is ‘0’.

chanhvinh.com

(9) Acknowledge bit

During normal communication (communication from one unit to another), an acknowledge bit is appended to the following locations to check whether the data has been correctly received.

- End of slave address field
- End of control field
- End of message length field
- End of data field

The definition of the acknowledge bit is as follows.

0: The transmission data is recognized. ($\overline{\text{ACK}}$ signal)

1: The transmission data is not recognized. (NACK signal)

During broadcast communication, however, the contents of the acknowledge bit are ignored.

(a) Last acknowledge bit of the slave address field

The last acknowledge bit of the slave address field serves as a NACK signal in any of the following cases, and transmission is stopped.

- If the parity of the master address bit or slave address bit is incorrect
- If a timing error (an error in the bit format) occurs
- If a slave unit does not exist

(b) Last acknowledge bit of the control field

The last acknowledge bit of the control field serves as a NACK signal in any of the following cases, and transmission is stopped.

- If the parity of the control bit is incorrect
- If control bit 3 is 1 (write operation) when the slave reception enable flag (the IEBBnBCR.IEBBnSRXE bit) is not set (to 1)

  (For details, see 22.3.2 (1) IEBBnBCR - IEBBn bus control register.)
- If control bit data is read (3H, 7H) when the slave transmission enable flag (the IEBBnBCR.IEBBnSTXE bit) is not set (to 1)

  (For details, see 22.3.2 (1) IEBBnBCR - IEBBn bus control register.)
- If a unit other than one that has set locking requests 3H, 6H, 7H, AH, BH, EH, or FH of the control bit when locking is set
- If the control bit indicates reading of lock addresses (4H, 5H) even when locking is not set
- If a timing error occurs
- If the control bit is undefined

chanhvinh.com

Cautions
1. The $\overline{\text{ACK}}$ signal is always returned when the control data of the slave status request is received, if the IEBBnSTXE bit = 0.
2. The NACK signal is returned by the acknowledge bit in the control field when the control data for data/command writing is received, even if the IEBBnSRXE bit = 0.
   Slave reception can be disabled (communication stopped) by the IEBBnSRXE bit only in the case of individual communication. In the case of broadcast communication, communication is maintained and the data interrupt (IEBBTD) or completion interrupt (IEBBTSTA) is generated.

(c) Last acknowledge bit of message length field

The last acknowledge bit of the message length field serves as a NACK signal in any of the following cases, and transmission is stopped.

- If the parity of the message length bit is incorrect
- If a timing error occurs

(d) Last acknowledge bit of the data field

The last acknowledge bit of the data field serves as a NACK signal in any of the following cases, and transmission is stopped.

- If the parity of the data bit is incorrect[a]
- If a timing error occurs after the preceding acknowledge bit has been transmitted
- If the received data is stored in the IEBBnDR register and no more data can be received[a]

a) In this case, when the communication executed is individual communication, if the maximum number of transmit bytes is within the value that can be transmitted in one frame, the transmission side executes transmission of that data field again. For broadcast communication, the transmission side does not execute transmission again, a communication error occurs on the reception side and reception stops.

22.7.7 Transfer data

(1) Slave status

The master unit can learn why the slave unit did not return the $\overline{\text{ACK}}$ signal by reading the slave status.

The slave status is determined according to the result of the last communication the slave unit has executed.

All the slave units can supply information on the slave status.

The configuration of the slave status is shown below.

chanhvinh.com

MSB LSB

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit 0[a]	Function
0	Transmit data is not written in IEBBnDR register
1	Transmit data is written in IEBBnDR register

Bit 1	Function
0	Receive data is not stored in IEBBnDR register
1	Receive data is stored in IEBBnDR register

Bit 2	Function
0	Unit is not locked
1	Unit is locked

Bit 3	Function
0	Fixed to 0

Bit 4[b]	Function
0	Slave transmission is stopped
1	Slave transmission is ready

Bit 5	Function
0	Fixed to 0

Bit 7	Bit 6	Function	
0	0	Communication mode 0	Indicates the highest communication mode supported by the unit[c].
0	1	Communication mode 1	
1	0	Communication mode 2	
1	1	Not used	

a) After reset: Bit 0 is set to 1.

b) When this module serves as a slave unit, this bit corresponds to the status indicated by IEBBnBCR.IEBBnSTXE bit.

c) Bits 7 and 6 are fixed to “10” because this module can support communication modes 1 and 2.

Figure 22-62 Slave status bit configuration

chanhvinh.com

(2) Lock address

When the lock address is read (control bit: 4H or 5H), the address (12 bits) of the master unit that has issued the lock instruction is configured in 1-byte units as shown below and read.

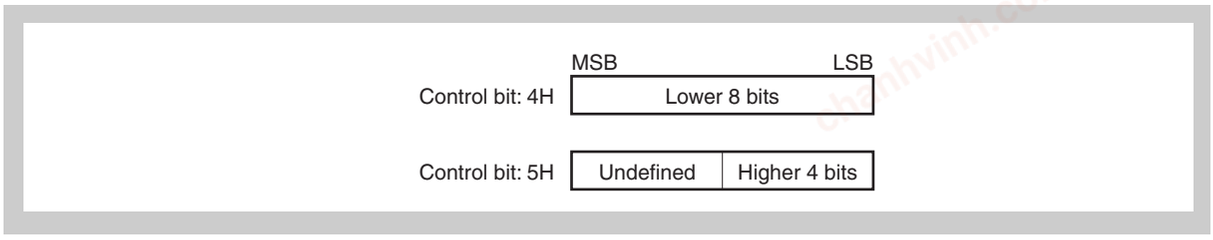


Figure 22-63 Lock address configuration

(3) Data

If the control bit indicates reading of data (3H or 7H), the data in the data buffer of the slave unit is read by the master unit.

If the control bit indicates writing of data (BH or FH), the data received by the slave unit is processed according to the operation rule of that slave unit.

(4) Locking and unlocking

The lock function is used when a message is transferred in two or more communication frames.

The unit that is locked does not receive data from units other than the one that has locked the unit (does not receive broadcast communication).

A unit is locked or unlocked as follows.

(a) Lock setting

If the communication frame is completed without succeeding to transmit or receive data of the number of bytes specified by the message length bit after the message length field has been transmitted or received ($\overline{ACK} = 0$) by the control bit that specifies locking (3H, AH, or BH), the slave unit is locked by the master unit. At this time, the bit (bit 2) in the byte indicating the slave status is set to ‘1’.

(b) Unlocked

After transmitting or receiving data of the number of data bytes specified by the message length bit in one communication frame by the control bit that has specified locking (3H, AH, or BH), or the control bit that has specified unlocking (6H), the slave unit is unlocked by the master unit. At this time, the bit related to locking (bit 2) in the byte indicating the slave status is reset to ‘0’.

Locking or unlocking is not performed during broadcast communication.

chanhvinh.com

Locking and unlocking conditions are shown below.

Table 22-63 Setting conditions:

Control data	Broadcast communication		Individual communication	
	End of communication	End of frame	End of communication	End of frame
3H, 6H[a]	–	–	Cannot be locked	Lock set
AH, BH	Cannot be locked	Cannot be locked	Cannot be locked	Lock set
0H, 4H, 5H, EH, FH	Cannot be locked	Cannot be locked	Cannot be locked	Cannot be locked

a) The frame end of control data 6H (slave status read/unlock) occurs when the parity in the data field is odd, and when the NACK signal from the IEBus unit is repeated with up to the maximum number of transfer bytes being output.

Table 22-64 Unlocking conditions (while locked)

Control data	Broadcast communication from the lock request unit		Individual communication from the lock request unit	
	End of communication	End of frame	End of communication	End of frame
3H, 6H[a]	–	–	Unlocked	Remains locked
AH, BH	Unlocked	Unlocked	Unlocked	Remains locked
0H, 4H, 5H, EH, FH	Remains locked	Remains locked	Remains locked	Remains locked

a) The frame end of control data 6H (slave status read/unlock) occurs when the parity in the data field is odd, and when the NACK signal from the IEBus unit is repeated with up to the maximum number of transfer bytes being output.

chanhvinh.com

22.7.8 Bit format

The format of the bits constituting the communication frame of the IEBus is shown below.

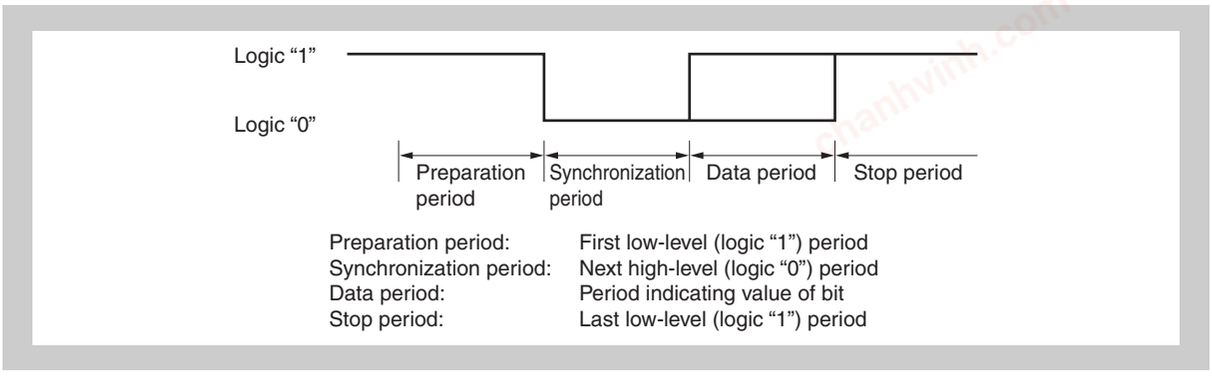


Figure 22-64 IEBus bit format

The synchronization period and data period are almost equal to each other in length.

The IEBus synchronizes each bit. The specifications on the time of the entire bit and the time related to the period allocated to that bit differ depending on the type of transmit bit, or whether the unit is the master unit or a slave unit. The master and slave units monitor whether each period (preparation period, synchronization period, data period, and stop period) is output for the specified time while they are in communication. If a period is not output for the specified time, the master and slave units report a timing error, immediately terminate communication, and enter the standby status.

chanhvinh.com

23. Renesas SPDIF Interface

23.1 Overview

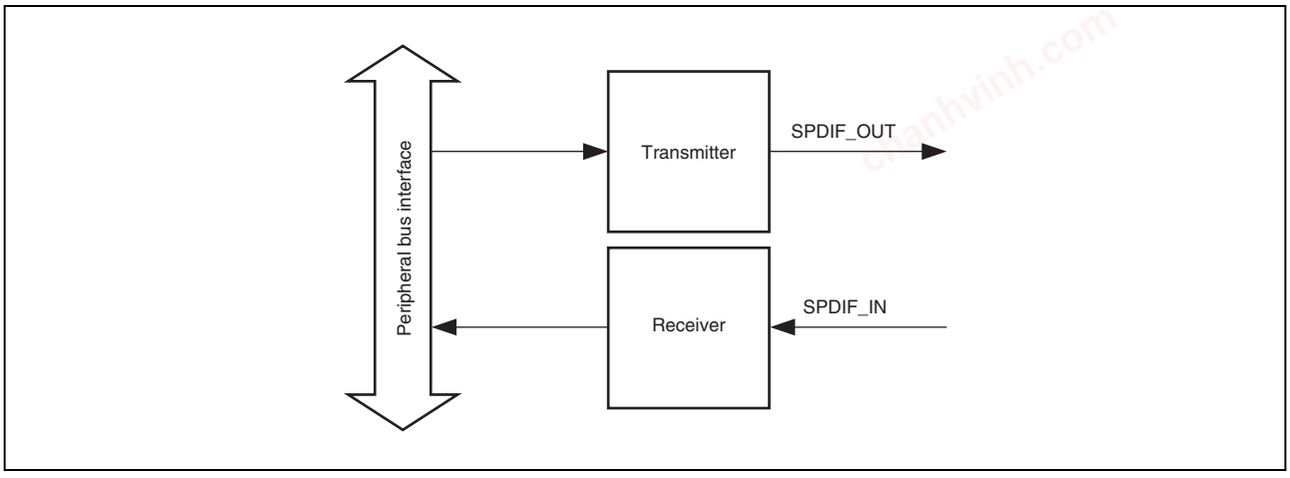


Figure 23.1 Overview Block Diagram

23.2 Features

- Supports the IEC 60958 standard (stereo and consumer use modes only).
- Supports sampling frequencies of 32 kHz, 44.1 kHz, and 48 kHz.
- Supports audio word sizes of 16 to 24 bits per sample.
- Biphase mark encoding.
- Double buffered data.
- Parity encoded serial data.
- Simultaneous transmit and receive
- Receiver autodetects IEC 61937 compressed mode data

chanhvinh.com

23.3 Functional Block Diagram

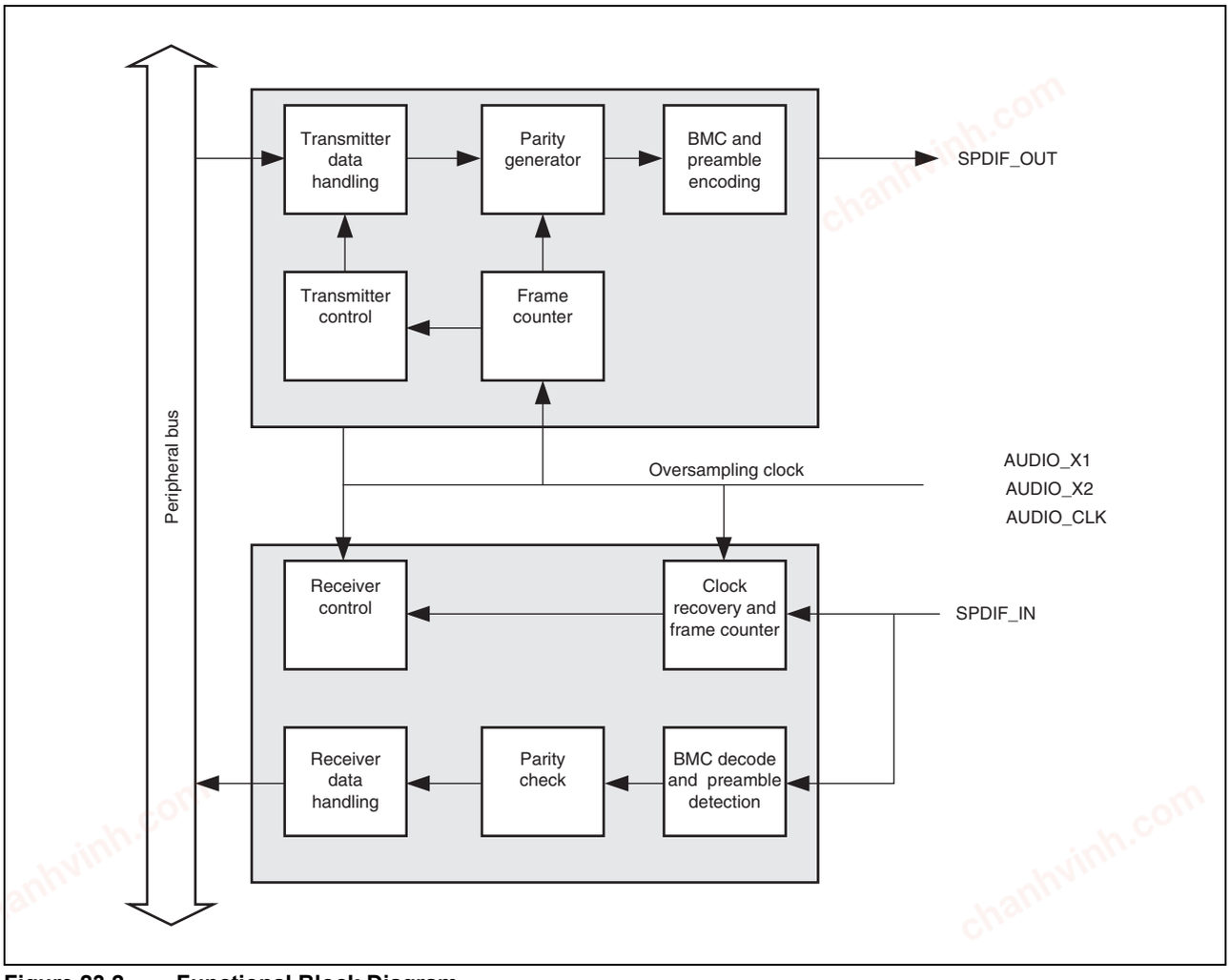


Figure 23.2 Functional Block Diagram

chanhvinh.com

23.4 Input/Output Pins

Table 23.1 shows the pin configuration.

Table 23.1 Pin Configuration

Channel	Pin Name	I/O	Description
0	SPDIF_OUT	Output	Transmitter biphase-mark encoded SPDIF bitstream
1	SPDIF_IN	Input	Receiver biphase-mark encoded SPDIF bitstream
0, 1 (Common)	AUDIO_CLK	Input	External clock for audio
	AUDIO_X1	Input	Crystal resonator/external clock for audio
	AUDIO_X2	Output	

23.5 Renesas SPDIF (IEC60958) Frame Format

The Renesas SPDIF frame consists of two subframes (for channels 1 and 2), each of which contains a 4-bit preamble, audio data of up to 24 bits, a V flag, a user bit, a channel status bit, and an even parity bit. Figure 23.3 shows the subframe format. According to this format, the Renesas SPDIF performs biphase-mark modulation (channel coding) that will make the transmission line's DC component a minimum value.

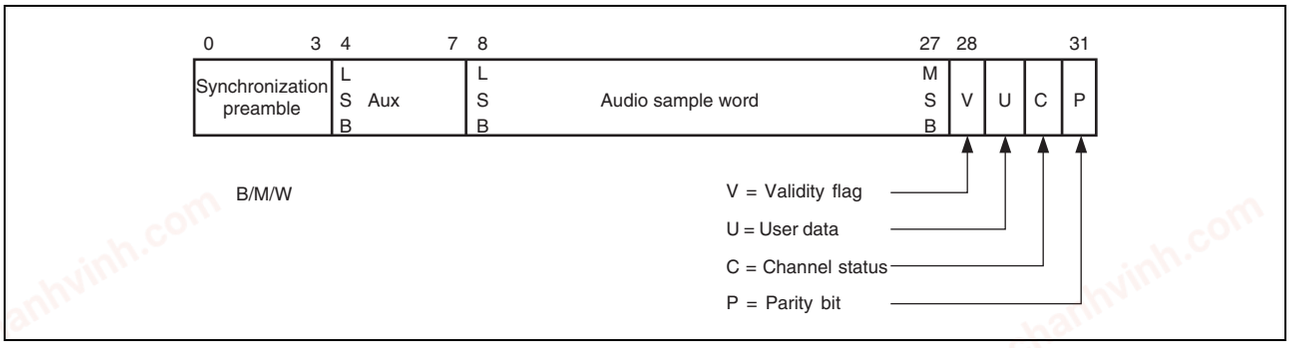


Figure 23.3 Subframe Format

Figure 23.4 shows the block format, which consists of 192 continuous frames. One block begins at the starting frame (preamble B) and ends at the 192nd frame (frame 191), and the preamble is used to identify all subframes. Each block has a total of 384 subframes, which are classified into three categories: subframe 0 indicating the beginning of a new block, subframe 1 (usually the channel 1), and subframe 2 (usually the channel 2). Usually, the music data sent and received by the SPDIF is continuous so that continuous blocks appear.

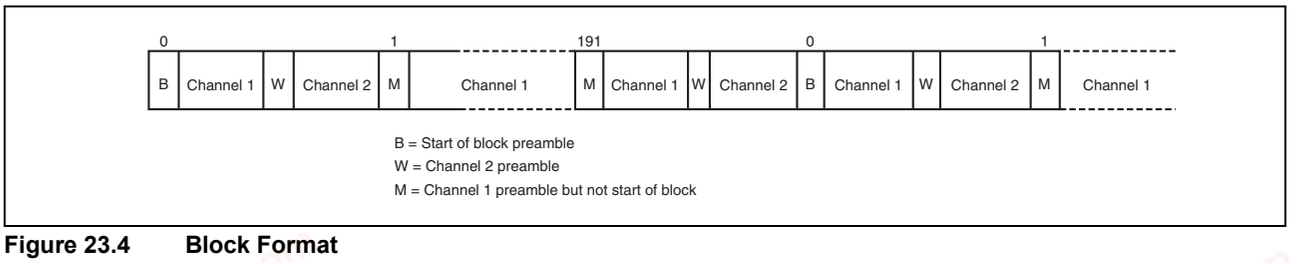


Figure 23.4 Block Format

chanhvinh.com

Table 23.2 shows the binary values of the Renesas SPDIF preambles. The polarity of these preambles differs depending on the status of the preceding symbol (parity bit).

Table 23.2 Binary Preamble Values

Preamble	Preceding Symbol's Status = 0	Preceding Symbol's Status = 1
B	11101000	00010111
M	11100010	00011101
W	11100100	00011011

Note: As shown in Figure 23.3, the even parity bit at time slot 31 of a subframe determines the type of a preamble for one cycle of transmission. Usually, therefore, any one is selected from the set states that are sent through the Renesas SPDIF. However, IEC60958 requires decoding both types in view of connection with the preamble polarity reversed; the Renesas SPDIF has preambles decoded according to Table 23.2.

Channel status information is encoded at the rate of one bit per subframe, making the channel status information per block have a total of 192 bits for each of subframes 1 and 2. For the format of the channel status, refer to the IEC 60958 standard.

chanhvinh.com

23.6 Register

Table 23.3 shows the register configuration.

Table 23.3 Register Configuration

Channel	Register Name	Abbreviation	Address	Access Size
0 (Transmit)	Transmitter channel 1 audio register	TLCA	H'FFFF D800	32
	Transmitter channel 2 audio register	TRCA	H'FFFF D804	32
	Transmitter channel 1 status register	TLCS	H'FFFF D808	32
	Transmitter channel 2 status register	TRCS	H'FFFF D80C	32
	Transmitter user data register	TUI	H'FFFF D810	32
1 (Receive)	Receiver channel 1 audio register	RLCA	H'FFFF D814	32
	Receiver channel 2 audio register	RRCA	H'FFFF D818	32
	Receiver channel 1 status register	RLCS	H'FFFF D81C	32
	Receiver channel 2 status register	RRCS	H'FFFF D820	32
	Receiver user data register	RUI	H'FFFF D824	32
0, 1 (Common)	Control register	CTRL	H'FFFF D828	32
	Status register	STAT	H'FFFF D82C	32
0, 1 (Common)	Transmitter DMA audio data register	TDAD	H'FFFF D830	32
	Receiver DMA audio data register	RDAD	H'FFFF D834	32

Note: All registers are longword registers and must be accessed as such.
A register diagram containing a 0 indicates that the write value should always be 0 (if the register is writeable) and that the read value should always be 0 (if readable).

chanhvinh.com

23.7 Register Descriptions

Legend:

Initial Value: Register value after reset
—: Undefined value
R/W: Readable/writable register. The write value can be read.
R: Read only register. The write value should always be 0.
R/WC0: Readable/writable register. Writing 0 initializes the bit, but writing 1 is ignored.
R/WC1: Readable/writable register. Writing 1 initializes the bit, but writing 0 is ignored.
W: Write only register. Reading is prohibited. If this bit is reserved, the write value should always be 0.
—/W: Write only, Read value undefined

23.7.1 Control Register (CTRL)

Bit:	31	30	29	28	27	26	25	24
	-	-	-	CKS	-	PB	RASS	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R/W	R/W	R/W

Bit:	23	22	21	20	19	18	17	16
	TASS		RDE	TDE	NCSI	AOS	RME	TME
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8
	REIE	TEIE	UBOI	UBUI	CREI	PAEI	PREI	CSEI
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	ABOI	ABUI	RUII	TUII	RCSI	RCBI	TCSI	TCBI
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved
28	CKS	0	R/W	Oversampling clock select Selects oversampling clock supply source. 0: AUDIO_X1 1: AUDIO CLK
27	—	0	R	Reserved
26	PB	0	R/W	Pass Back Passes transmitter SPDIF output into SPDIF receiver in SPDIF module. 0: Pass Back disabled 1: Pass Back enabled
25, 24	RASS	All 0	R/W	Receiver Audio Sample Bit Size These bits Indicate the receiver audio sample bit size (16, 20, or 24 bits), for data alignment purposes. 00: 16-bit sample 01: 20-bit sample 10: 24-bit sample 11: Reserved

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
23, 22	TASS	All 0	R/W	Transmitter Audio Sample Bit Size These bits Indicate the transmitter audio sample bit size (16, 20, or 24 bits), for data alignment purposes. 00: 16-bit sample 01: 20-bit sample 10: 24-bit sample 11: Reserved
21	RDE	0	R/W	Receiver DMA Enable Enables DMA requests for the receiver. 0: Receiver DMA disabled 1: Receiver DMA enabled
20	TDE	0	R/W	Transmitter DMA Enable Enables the DMA requests for the transmitter. 0: Transmitter DMA disabled 1: Transmitter DMA enabled
19	NCSI	0	R/W	New Channel Status Information Set this bit to 1 when new channel status information to be corrected is in the transmitter. 0: New channel status information has not been in transmitter 1: New channel status information has been in transmitter
18	AOS	0	R/W	Audio Only Samples Clear this bit to 0 when audio channel 1 and channel 2 registers contain user information. When this bit is set to 1, all user bits are cleared to 0. 0: User information present 1: User information not present
17	RME	0	R/W	Receiver Module Enable Enables the receiver module. 0: Receiver module disabled 1: Receiver module enabled
16	TME	0	R/W	Transmitter Module Enable Enables the transmitter module. 0: Transmitter module disabled 1: Transmitter module enabled
15	REIE	0	R/W	Receiver Error Interrupt Enable Enables the receiver error interrupts. 0: Receiver error interrupt disabled 1: Receiver error interrupt enabled
14	TEIE	0	R/W	Transmitter Error Interrupt Enable Enables the transmitter error interrupts. 0: Transmitter error interrupt disabled 1: Transmitter error interrupt enabled
13	UBOI	0	R/W	User Buffer Overrun Interrupt Enable Enables the user buffer overrun interrupts. 0: User buffer overrun interrupt disabled 1: User buffer overrun interrupt enabled
12	UBUI	0	R/W	User Buffer Underrun Interrupt Enable Enables the user buffer underrun interrupts. 0: User buffer underrun interrupt disabled 1: User buffer underrun interrupt enabled
11	CREI	0	R/W	Clock Recovery Error Interrupt Enable Enables the clock recovery error interrupts. 0: Clock recovery error interrupt disabled 1: Clock recovery error interrupt enabled
10	PAEI	0	R/W	Parity Error Interrupt Enable Enables the parity check error interrupts. 0: Parity check error interrupt disabled 1: Parity check error interrupt enabled

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
9	PREI	0	R/W	Preamble Error Interrupt Enable Enables the preamble check error interrupts. 0: Preamble error interrupt disabled 1: Preamble error interrupt enabled
8	CSEI	0	R/W	Channel Status Error Interrupt Enable Enables the channel status error interrupts. 0: Channel status error interrupt disabled 1: Channel status error interrupt enabled
7	ABOI	0	R/W	Audio Buffer Overrun Interrupt Enable Enables the receiver audio buffer overrun interrupts. 0: Audio buffer overrun interrupt disabled 1: Audio buffer overrun interrupt enabled
6	ABUI	0	R/W	Audio Buffer Underrun Interrupt Enable Enables the transmitter audio buffer underrun interrupts. 0: Audio buffer underrun interrupt disabled 1: Audio buffer underrun interrupt enabled
5	RUII	0	R/W	Receiver User Information Interrupt Enable Enables the receiver user information register full interrupts. 0: Receiver user information interrupt disabled 1: Receiver user information interrupt enabled
4	TUII	0	R/W	Transmitter User Information Interrupt Enable Enables the transmitter user information register empty interrupts. 0: Transmitter user information interrupt disabled 1: Transmitter user information interrupt enabled
3	RCSI	0	R/W	Receiver Channel Status Interrupt Enable Enables the receiver channel status register full interrupts. 0: Receiver channel status interrupt disabled 1: Receiver channel status interrupt enabled
2	RCBI	0	R/W	Receiver Channel Buffer Interrupt Enable Enables the receiver audio channel buffer full interrupts. 0: Receiver audio channel interrupt disabled 1: Receiver audio channel interrupt enabled
1	TCSI	0	R/W	Transmitter Channel Status Interrupt Enable Enables the transmitter channel status register empty interrupts. 0: Transmitter channel status interrupt disabled 1: Transmitter channel status interrupt enabled
0	TCBI	0	R/W	Transmitter Channel Buffer Interrupt Enable Enables the transmitter audio channel buffer empty interrupts. 0: Transmitter audio channel interrupt disabled 1: Transmitter audio channel interrupt enabled

chanhvinh.com

23.7.2 Status Register (STAT)

Bit:	31	30	29	28	27	26	25	24
	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	CMD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8
	RIS	TIS	UBO	UBU	CE	PARE	PREE	CSE
Initial value:	1	1	0	0	0	0	0	0
R/W:	R	R	R/WC0	R/WC0	R/WC0	R/WC0	R/WC0	R/WC0

Bit:	7	6	5	4	3	2	1	0
	ABO	ABU	RUIR	TUIR	CSRX	CBRX	CSTX	CBTX
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/WC0	R/WC0	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved
16	CMD	0	R	Compressed Mode Data Sets if the data being received is compressed mode data (When bit 1 = 1 in the V flag and channel status). 0: Data is not in compressed mode 1: Data is in compressed mode
15	RIS	1	R	Receiver Idle State Sets if the receiver is in the idle state. 0: Receiver is not in idle state 1: Receiver in idle state
14	TIS	1	R	Transmitter Idle State Sets if the transmitter is in the idle state. 0: Transmitter is not in idle state 1: Transmitter is in idle state
13	UBO	0	R/WC0	User Buffer Overrun* Sets if the receiver user buffer overruns. This bit is cleared by writing 0 to the register. If bit REIE and bit UBOI in the control register are set this causes an interrupt. 0: User buffer has not overrun 1: User buffer has overrun
12	UBU	0	R/WC0	User Buffer Underrun* Sets if the transmitter user buffer underrun. This bit is cleared by writing 0. If bits TEIE and UBUI in the control register are set this causes an interrupt. 0: User buffer has not underrun 1: User buffer has underrun
11	CE	0	R/WC0	Clock Error* Sets when the clock recovery falls out of synchronization. This bit is cleared by writing 0. If bits REIE and CREI in the control register are set this causes an interrupt. 0: Clock recovery stable 1: Clock recovery error
10	PARE	0	R/WC0	Parity Error* Sets when the parity checker produces a fail result. This bit is cleared by writing 0. If bits REIE and PAEI in the control register are set this causes an interrupt. 0: Parity check correct 1: Parity error

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
9	PREE	0	R/WC0	Preamble Error* Sets when the start of word preamble fails to appear in the correct place. This bit is cleared by writing 0. If bits REIE and PREI in the control register are set this causes an interrupt. Note: Only set after a start of block preamble has occurred. 0: Preamble is in the correct place 1: Preamble error
8	CSE	0	R/WC0	Channel Status Error*1 Sets when the channel status information is written before the 32nd frame of the current block. This bit is cleared by writing 0. If bits TEIE and CSEI in the control register are set this causes an interrupt. 0: Channel status correct 1: Channel status error
7	ABO	0	R/WC0	Audio Buffer Overrun*1 Indicates that the receiver audio buffer is full in both the first and second stages and that data has been overwritten. This bit is cleared by writing 0. If bits REIE and ABOI in the control register are set then this causes an interrupt. 0: Receiver audio buffer has not overrun 1: Receiver audio buffer has overrun
6	ABU	0	R/WC0	Audio Buffer Underrun*1 Indicates that the transmitter audio buffer is empty in both the first and second stages and that the last data transmission has been repeated. This bit is cleared by writing 0. If bits TEIE and ABUI in the control register are set then this causes an interrupt. 0: Transmitter audio buffer has not underrun 1: Transmitter audio buffer has underrun
5	RUIR	0	R	Receiver User Information Register Status Indicates the status of the receiver user information register. This bit is cleared by reading from the receiver user register. If bit RUII in the control register is set then this causes an interrupt. 0: Receiver user information register is empty 1: Receiver user information register is full
4	TUIR	0	R	Transmitter User Information Register Status Indicates the status of the transmitter user information register. This bit is cleared by writing to the transmitter user register. If bit TUII in the control register is set then this causes an interrupt. 0: Transmitter user information register is full 1: Transmitter user information register is empty
3	CSRX	0	R	Channel 1 and Channel 2 Status for Receiver Indicates the status of the receiver channel status registers. This bit is cleared by reading from the receiver channel status registers. If bit RCSI in the control register is set this causes an interrupt. 0: Receiver channel status registers are empty 1: Receiver channel status registers are full
2	CBRX	0	R	Channel 1 and Channel 2 Buffers for Receiver Indicates the status of the receiver audio channel registers. This bit is cleared by reading from the receiver audio channel registers. If bit RCBI in the control register is set this causes an interrupt. 0: Receiver audio channel registers are empty 1: Receiver audio channel registers are full
1	CSTX	0	R	Channel 1 and Channel 2 Status for Transmitter Indicates the status of the transmitter channel status registers. This bit is cleared by writing to the transmitter channel status registers. If bit TCSI in the control register is set this causes an interrupt. 0: Transmitter channel status register is full 1: Transmitter channel status register is empty

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
0	CBTX	0	R	Channel 1 and Channel 2 Buffers for Transmitter Indicates the status of the transmitter audio channel registers. This bit is cleared by writing to the transmitter audio channel registers. If bit TCBI in the control register is set this causes an interrupt. 0: Transmitter audio channel registers are full 1: Transmitter audio channel registers are empty

Note 1. When an error bit is detected during DMA transfer, DMA transfer settings must be made again. In this case, the Renesas SPDIF's module enable bit (either the RME or TME bit) and the DMA enable bit (either the RDE or TDE bit) must be disabled and the error status must be cleared before making DMA transfer settings again. Then the module enable bit should be set and DMA transfer can be started again.

23.7.3 Transmitter Channel 1 Audio Register (TLCA)

Bit:	31	30	29	28	27	26	25	24
	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W

Bit:	23	22	21	20	19	18	17	16
	Audio PCM Data							
Initial value:	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W

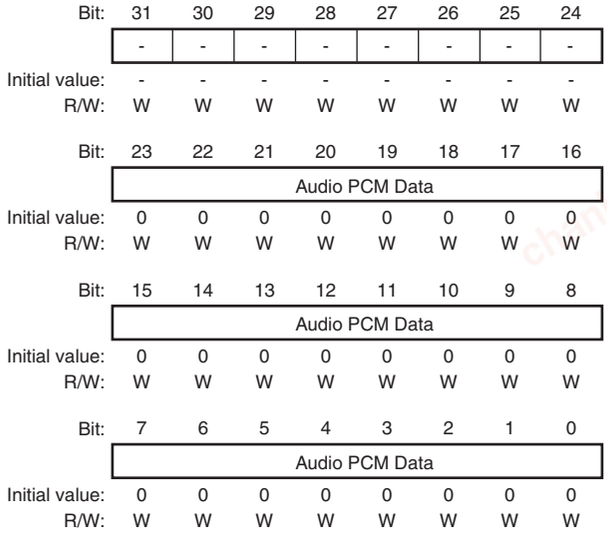
Bit:	15	14	13	12	11	10	9	8
	Audio PCM Data							
Initial value:	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W

Bit:	7	6	5	4	3	2	1	0
	Audio PCM Data							
Initial value:	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	—	W	Reserved
23 to 0	Audio PCM Data	All 0	W	Audio PCM Data LSB aligned PCM encoded audio data.

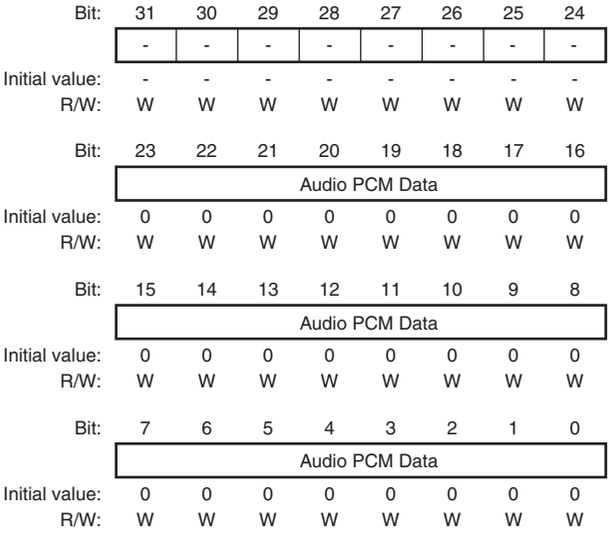
chanhvinh.com

23.7.4 Transmitter Channel 2 Audio Register (TRCA)



Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	—	W	Reserved
23 to 0	Audio PCM Data	All 0	W	Audio PCM Data LSB aligned PCM encoded audio data.

23.7.5 Transmitter DMA Audio Data Register (TDAD)



Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	—	W	Reserved
23 to 0	Audio PCM Data	All 0	W	Audio PCM Data LSB aligned PCM encoded audio data.

chanhvinh.com

23.7.6 Transmitter User Data Register (TUI)

U-bit data in subframes is written in to this register. Because U-bit data is transmitted in a sequence of subframes 1 and 2, you need to update the data on a 16-frame basis. For the contents of the user bytes refer to the appropriate standard for the device in use. The user bits to be transmitted are set in sequence starting at the LSB.

Bit:	31	30	29	28	27	26	25	24
	User Byte 4							
Initial value:	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W
Bit:	23	22	21	20	19	18	17	16
	User Byte 3							
Initial value:	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W
Bit:	15	14	13	12	11	10	9	8
	User Byte 2							
Initial value:	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W
Bit:	7	6	5	4	3	2	1	0
	User Byte 1							
Initial value:	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	User Byte 4	All 0	W	U-bit information is stored here.
23 to 16	User Byte 3	All 0	W	
15 to 8	User Byte 2	All 0	W	
7 to 0	User Byte 1	All 0	W	

chanhvinh.com

23.7.7 Transmitter Channel 1 Status Register (TLCS)

The 30-bit register stores the channel status information to be transmitted. For each channel, channel status information per frame consists of 192 bits. Because necessary data covers only the 30 bits that are set in the following register, zeros continue to be sent after the transmission of the first 30 bits.

Bit:	31	30	29	28	27	26	25	24
	-	-	CLAC[1:0]		FS[3:0]			
Initial value:	-	-	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W

Bit:	23	22	21	20	19	18	17	16
	CHNO[3:0]				SRCNO[3:0]			
Initial value:	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8
	CATCD[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W

Bit:	7	6	5	4	3	2	1	0
	-	-	CTL[4:0]					-
Initial value:	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	—	W	Reserved
29, 28	CLAC[1:0]	All 0	W	Clock Accuracy 00: Level 2 01: Level 1 10: Level 3 11: Reserved
27 to 24	FS[3:0]	All 0	W	Sample Frequency (FS) 0000: 44.1 kHz 0010: 48 kHz 0011: 32 kHz
23 to 20	CHNO[3:0]	All 0	W	Channel Number 0000: Don't care 0001: A (left channel) 0010: B (right channel) 0011: C
19 to 16	SRCNO[3:0]	All 0	W	Source Number 0000: Don't care 0001: 1 0010: 2 0011: 3
15 to 8	CATCD[7:0]	All 0	W	Category Code (Example) 00000000: 2-channel general format 00000001: 2-channel compact disc (IEC 908) 00000010: 2-channel PCM encoder/decoder 00000011: 2-channel digital audio tape recorder
7, 6	—	All 0	W	Reserved The write value should always be 0.
5 to 1	CTL[4:0]	All 0	W	Control The control bits are copied from the source (see IEC60958 standard).
0	—	0	W	Reserved The write value should always be 0.

chanhvinh.com

23.7.8 Transmitter Channel 2 Status Register (TRCS)

The 30-bit register stores the channel status information to be transmitted. For each channel, channel status information per frame consists of 192 bits. Because necessary data covers only the 30 bits that are set in the following register, zeros continue to be sent after the transmission of the first 30 bits.

Bit:	31	30	29	28	27	26	25	24
	-	-	CLAC[1:0]		FS[3:0]			
Initial value:	-	-	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W

Bit:	23	22	21	20	19	18	17	16
	CHNO[3:0]				SRCNO[3:0]			
Initial value:	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W

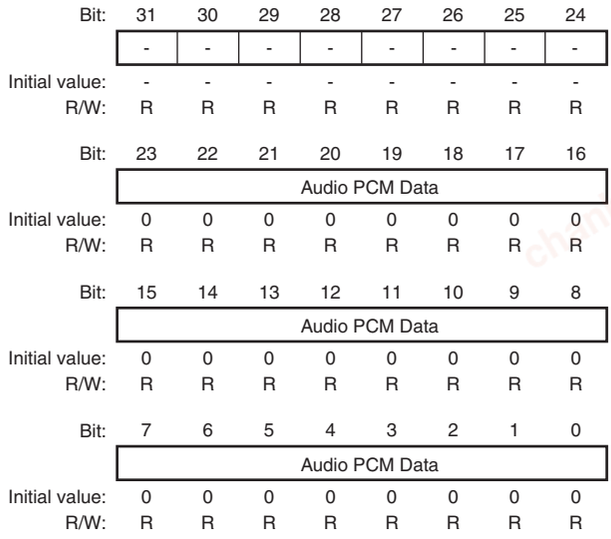
Bit:	15	14	13	12	11	10	9	8
	CATCD[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W

Bit:	7	6	5	4	3	2	1	0
	-	-	CTL[4:0]					-
Initial value:	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	—	W	Reserved
29, 28	CLAC[1:0]	All 0	W	Clock Accuracy 00: Level 2 01: Level 1 10: Level 3 11: Reserved
27 to 24	FS[3:0]	All 0	W	Sample Frequency (FS) 0000: 44.1 kHz 0010: 48 kHz 0011: 32 kHz
23 to 20	CHNO[3:0]	All 0	W	Channel Number 0000: Don't care 0001: A (left channel) 0010: B (right channel) 0011: C
19 to 16	SRCNO[3:0]	All 0	W	Source Number 0000: Don't care 0001: 1 0010: 2 0011: 3
15 to 8	CATCD[7:0]	All 0	W	Category Code (Example) 00000000: 2-channel general format 00000001: 2-channel compact disc (IEC 908) 00000010: 2-channel PCM encoder/decoder 00000011: 2-channel digital audio tape recorder
7, 6	—	All 0	W	Reserved The write value should always be 0.
5 to 1	CTL[4:0]	All 0	W	Control The control bits are copied from the source (see IEC60958 standard).
0	—	0	W	Reserved The write value should always be 0.

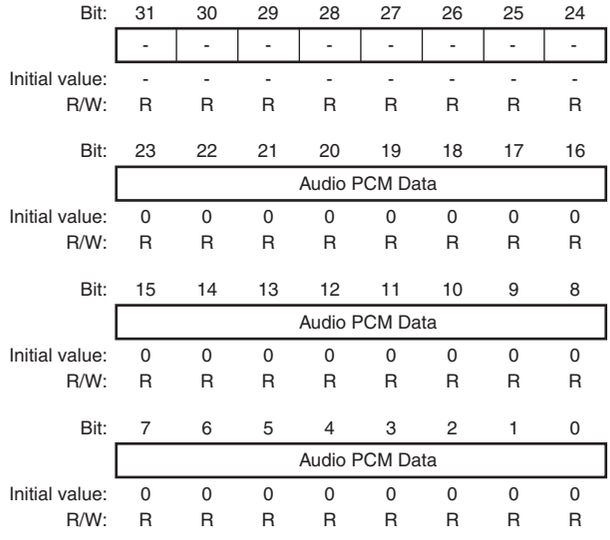
chanhvinh.com

23.7.9 Receiver Channel 1 Audio Register (RLCA)



Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	—	R	Reserved
23 to 0	Audio PCM Data	All 0	R	Audio PCM Data LSB aligned PCM encoded audio data.

23.7.10 Receiver Channel 2 Audio Register (RRCA)



Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	—	R	Reserved
23 to 0	Audio PCM Data	All 0	R	Audio PCM Data LSB aligned PCM encoded audio data.

chanhvinh.com

23.7.11 Receiver DMA Audio Data (RDAD)

Bit:	31	30	29	28	27	26	25	24
	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R

Bit:	23	22	21	20	19	18	17	16
	Audio PCM Data							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8
	Audio PCM Data							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	Audio PCM Data							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	—	R	Reserved
23 to 0	Audio PCM Data	All 0	R	Audio PCM Data LSB aligned PCM encoded audio data.

chanhvinh.com

23.7.12 Receiver User Data Register (RUI)

The register stores the U-bit data received through the Renesas SPDIF. Because U-bit data is stored in a sequence of subframes 1 and 2 starting at the LSB, you need to read the data on a 16-frame basis. For the contents of the user bytes refer to the appropriate standard for the device in use.

Bit:	31	30	29	28	27	26	25	24
	User Byte 4							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	23	22	21	20	19	18	17	16
	User Byte 3							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8
	User Byte 2							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	User Byte 1							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	User Byte 4	All 0	R	U-bit information is stored here.
23 to 16	User Byte 3	All 0	R	
15 to 8	User Byte 2	All 0	R	
7 to 0	User Byte 1	All 0	R	

chanhvinh.com

23.7.13 Receiver Channel 1 Status Register (RLCS)

The channel status is stored starting at the register's LSB in a way that subframe 1 received from the beginning of the block is stored. For the contents of the channel status register, refer to the IEC-60958 standard.

Bit:	31	30	29	28	27	26	25	24
	-	-	CLAC[1:0]		FS[3:0]			
Initial value:	-	-	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	23	22	21	20	19	18	17	16
	CHNO[3:0]				SRCNO[3:0]			
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8
	CATCD[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	-	-	CTL[4:0]					-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	—	R	Reserved
29, 28	CLAC[1:0]	All 0	R	Clock Accuracy 00: Level 2 01: Level 1 10: Level 3 11: Reserved
27 to 24	FS[3:0]	All 0	R	Sample Frequency (FS) 0000: 44.1 kHz 0010: 48 kHz 0011: 32 kHz
23 to 20	CHNO[3:0]	All 0	R	Channel Number 0000: Don't care 0001: A (left channel) 0010: B (right channel) 0011: C
19 to 16	SRCNO[3:0]	All 0	R	Source Number 0000: Don't care 0001: 1 0010: 2 0011: 3
15 to 8	CATCD[7:0]	All 0	R	Category Code (Example) 00000000: 2-channel general format 00000001: 2-channel compact disc (IEC 908) 00000010: 2-channel PCM encoder/decoder 00000011: 2-channel digital audio tape recorder
7, 6	—	All 0	R	Reserved
5 to 1	CTL[4:0]	All 0	R	Control The control bits are copied from the source (see IEC60958 standard).
0	—	0	R	Reserved

chanhvinh.com

23.7.14 Receiver Channel 2 Status Register (RRCS)

The channel status is stored starting at the register's LSB in a way that subframe 2 received from the beginning of the block is stored. For the contents of the channel status register, refer to the IEC-60958 standard.

Bit:	31	30	29	28	27	26	25	24
	-	-	CLAC[1:0]		FS[3:0]			
Initial value:	-	-	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	23	22	21	20	19	18	17	16
	CHNO[3:0]				SRCNO[3:0]			
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8
	CATCD[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	-	-	CTL[4:0]					-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	—	R	Reserved
29, 28	CLAC[1:0]	All 0	R	Clock Accuracy 00: Level 2 01: Level 1 10: Level 3 11: Reserved
27 to 24	FS[3:0]	All 0	R	Sample Frequency (FS) 0000: 44.1 kHz 0010: 48 kHz 0011: 32 kHz
23 to 20	CHNO[3:0]	All 0	R	Channel Number 0000: Don't care 0001: A (left channel) 0010: B (left channel) 0011: C
19 to 16	SRCNO[3:0]	All 0	R	Source Number 0000: Don't care 0001: 1 0010: 2 0011: 3
15 to 8	CATCD[7:0]	All 0	R	Category Code (Example) 00000000: 2-channel general format 00000001: 2-channel compact disc (IEC 908) 00000010: 2-channel PCM encoder/decoder 00000011: 2-channel digital audio tape recorder
7, 6	—	All 0	R	Reserved
5 to 1	CTL[4:0]	All 0	R	Control The control bits are copied from the source (see IEC60958 standard).
0	—	0	R	Reserved

chanhvinh.com

23.8 Functional Description—Transmitter

23.8.1 Transmitter Module

The transmitter module transmits PCM data and auxiliary information after encoding it according to the method of biphase-mark modulation that complies with the IEC60958 standard (SPDIF).
The clock for the transmitter module is an oversampling clock supplied from the outside. This clock usually selects a value that serves as an oversample at a frequency eight times larger than the clock frequency required for biphase-mark encoding. In this case, the clock frequency required to transmit 32 time slots in a subframe is 512 times as large as the sample frequency for audio data.
Audio data and channel status information are first written into the module's channel 1 and then into channel 2. Generally, the channel status need to be written only when the information changes. The SPDIF module requests that the channel status be written in 30 frames -- when all the current channel status data have been transmitted. You need to write somewhere between frame 31 and the beginning of the next block of 192 frames.
The audio data is stored in a double buffer arrangement. To make sure that the first stage buffer is empty, you can send an interrupt request or poll the status register. DMA transfers send channel 1 audio data on the first request and channel 2 data on the second.
The channel status information is stored in the 30-bit registers of channels 1 and 2. For each channel, the channel status information per frame consists of 192 bits. Because necessary data covers only 30 bits, zeros continue to be sent after the transmission of the first 30 bits until the block is completed.
User data forms a 32-bit double buffer arrangement. You can make sure that the first stage buffer is empty by either sending an interrupt request or polling the status register. Usually, information about the user data will become insufficient with the length of data between blocks. Transmission takes place in a sequence of channels 1 and 2. For the user data within a block, 384 bits are transmitted before the next block is continuously transmitted.
The audio data handled by the Renesas SPDIF module is a linear PCM, making it possible to set up to 24 bits. For this reason, the V flag indicating that audio data is a linear PCM remains to be 0. The V flag involves no register-based setting. An even parity is created for each 32 bits of serial output data (excluding the preamble).

Note: • When transmitter user buffer underrun occurs, the current data in the buffer data of SPDIF is transmitted until the next data is filled.

23.8.2 Transmitter Module Initialization

The device defaults to an idle state when it comes out of reset, or can be put into an idle state when 0 is written to the TME bit in the CTRL register. When the transmitter module is idle, it has the following settings:

- The transmitter idle status bit (TIS) is set to 1, all other status bits are cleared to 0.
- Preamble generation is invalid.
- Synchronization between channels 1 and 2 is set to 0 (0 for channel 1, 1 for channel 2).
- Both word_count and frame_count are set to 0.
- The output from the biphase-mark encoder is set to 0.

Channel status, user and audio data registers will retain its value prior to putting the module into idle. To exit the idle state the user must write 1 to the TME bit in the CTRL register.

chanhvinh.com

23.8.3 Initial Settings for Transmitter Module

When the TME bit is set to 1, the TUIR and CSTX bits are set to 1. After that, if data is written in the order of 1) TUI and 2) TLCS and TRCS, a channel status error will occur. To avoid this, be sure to write data in the order of 1) TLCS and TRCS and 2) TUI.

Before writing the first audio data (write access to TLCA or TRCA by the CPU or write access to TDAD by the DMA transfer) after setting the TME bit to 1, be sure to check that the CSTX and TUIR bits are cleared by writing to TLCS, TRCS, and TUI.

23.8.4 Transmitter Module Data Transfer

Once the transmitter module has left the idle state, it is ready for data transfer. Data transfer timing can be achieved in three ways. Either the transfer is done by interrupts, DMA requests or by polling the status register. There is a shared interrupt line (for both transmit and receive) and a single transmitter DMA request line.

Figure 23.5 shows a data transfer with an interrupt for the transmitter.

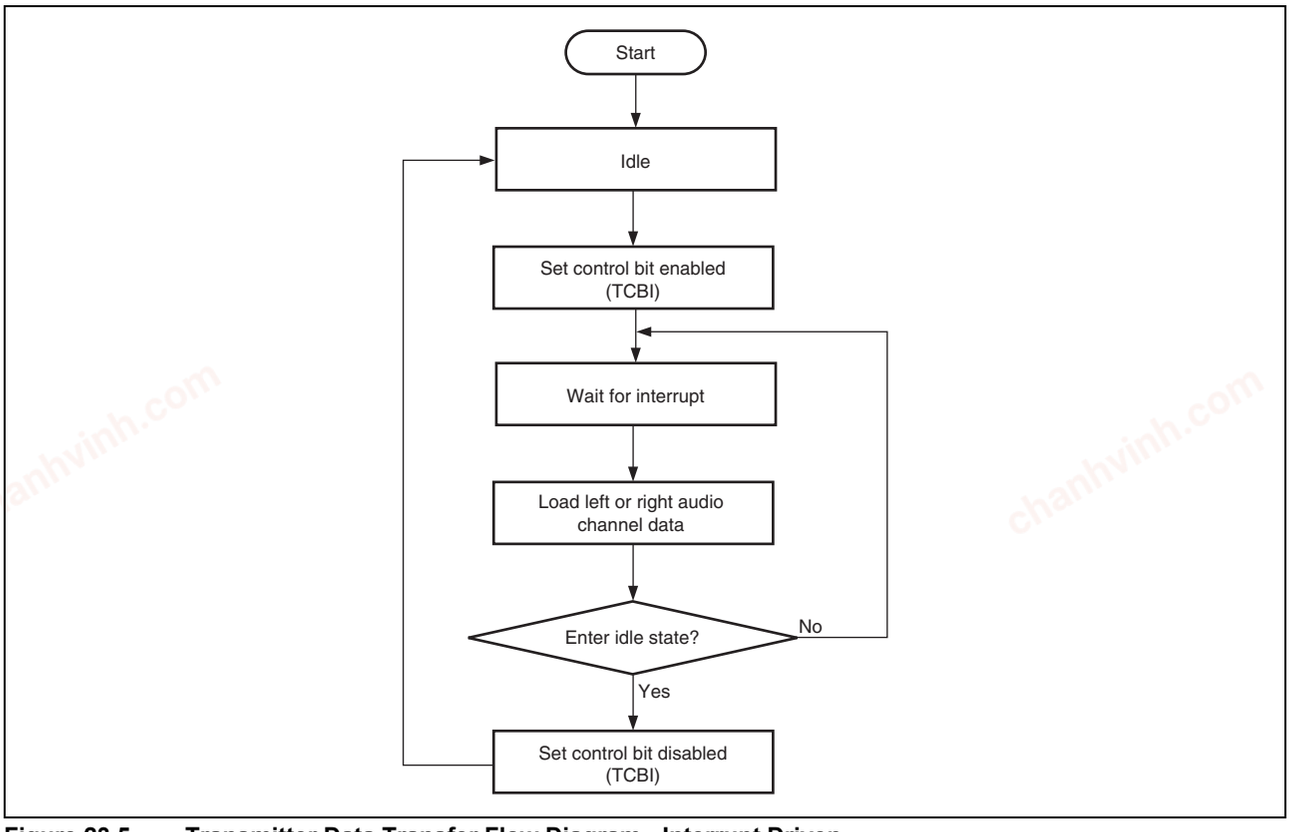


Figure 23.5 Transmitter Data Transfer Flow Diagram - Interrupt Driven

chanhvinh.com

Figure 23.6 shows a data transfer with a DMA transfer for the transmitter.

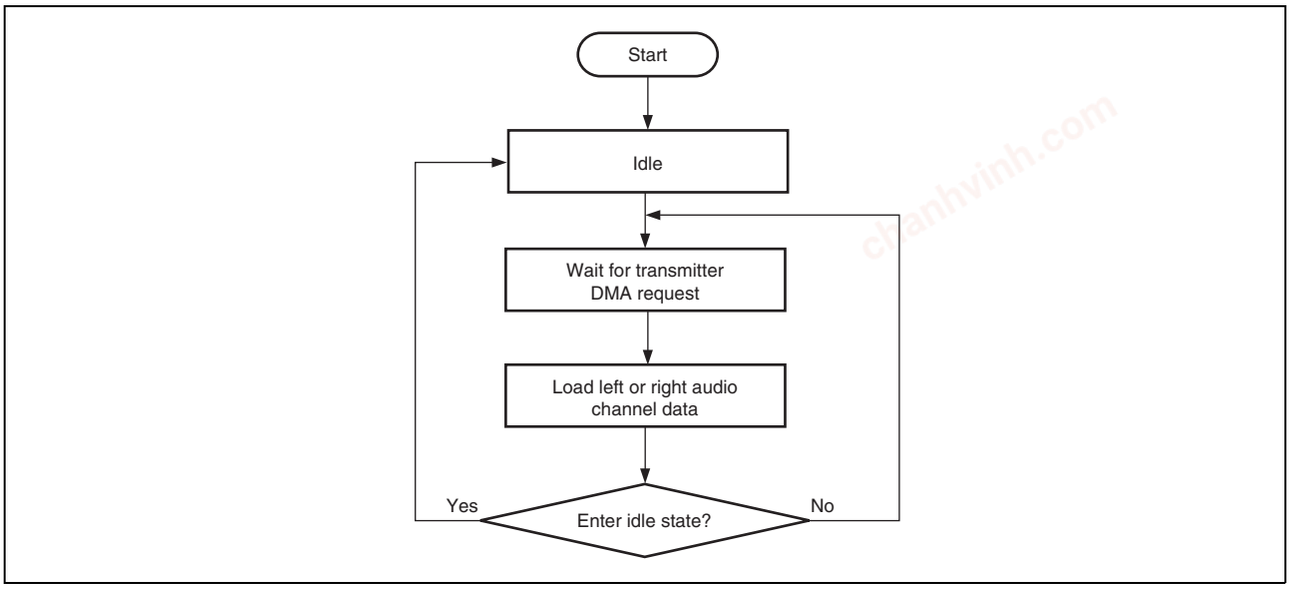


Figure 23.6 Transmitter Data Transfer Flow Diagram—DMA Request Driven

Channel status information is required to be updated when the information has changed. Because the updating needs to be done before the transmission of the next block, the channel status to be updated should be written after 30 frames have been sent; this is indicated either by an interrupt or by polling the status bit. If channel status is written before 30 frames have been sent (while current information is being sent) then an interrupt indicates that the channel status error bit (CSE) in the status register has been set.

Note: • 30 frames contains all the valid information in a single channel status block.

chanhvinh.com

23.9 Functional Description—Receiver

23.9.1 Receiver Module

The receiver module demodulates data and clock signals from the input encoded according to the IEC60958 standard. The encoded data, shown in linear PCM format, is stored into the audio data register. The register also stores the channel status and user information being received simultaneously as auxiliary information.
The main clock for the receiver module is an oversampling clock supplied from the outside. The module operates at a frequency four times as large as the oversampling clock.

Note: • The oversampling clock is the same for the transmitter and receiver.

Clock recovery is performed using a pulse width counter and averaging filters to produce a sampling pulse in the middle of each bit in the datastream. A clock error status bit indicates clock synchronization loss. Synchronization is achieved when a preamble occurs on the data stream for the first time. Continuous adjustment prevents jitter and/or clock drift from affecting clock recovery, provided that they fall within the clock recovery specifications.
Once the clock recovery is successful the biphase-mark decoder initiates its preamble detection. The decoder searches for the start of block preamble (see Table 23.2). A preamble error status bit indicates that following preambles have not appeared at the correct time, such failures are most likely caused by transmission loss or interference.
Even parity checking is performed on the decoded data. A discrepancy will result in the parity error status bit being set.
The SPDIF module acquires user data and channel status information in addition to audio data. The audio is stored in a double buffer arrangement. Either an interrupt request because of a full buffer or polling of the status bit will indicate when the data is ready to be read. DMA transfers receive channel 1 audio data on the first request and channel 2 data on the second.
Channel status is stored in a 30-bit register. Channel status information is received at 1-bit per subframe. Therefore the registers will not be full until a total of 30 frames for each channel have been received. New channel status is compared with the current data to see if it has changed and is only read by the processor if it has. User data, which is also received at the same time, is stored into the register on a subframe basis, so that the reception is completed when 16 frames are reached.

Note 1. Channel status data requests do not support DMA.
Note 2. When receiver user buffer overrun occurs, the current data in the buffer data of SPDIF is overwritten by the next incoming data from SPDIF interface.

23.9.2 Receiver Module Initialization

The device defaults to an idle state when it comes out of reset, or can be put into an idle state by writing 0 to bit RME in the CTRL register. Whilst idle the module has the following settings:

- The receiver idle status bit is set to 1, all other status bits are cleared to 0.
- Synchronization between channels 1 and 2 is set to 0 (0 for channel 1, 1 for channel 2).
- Both Word_count and frame_count are set to 0.

Channel status registers, user data registers and audio data registers will retain its value prior to putting the module into idle. To exit the idle state the user must write 1 to the bit RME in the CTRL register.

chanhvinh.com

23.9.3 Receiver Module Data Transfer

Once the module has left the idle state it is ready for data transfer. Data transfer timing can be achieved in three ways. The transfer can be done by interrupts, or by polling the status register, or by DMA. There is a shared interrupt line (transmit and receive) and a single receiver DMA request line. Data transfer for the receiver can be interrupted by error signals caused by:

1. Clock recovery failure.
2. Transmission loss or interference – indicated by a preamble error.
3. Parity check failure.

Transmission loss or interference can cause the start of subframe or start of block preamble to be misplaced or not present.
Parity check failure occurs when the parity bit is incorrect, this can be caused by any of the above.

- Clock Recovery Deviation
  The receive margin for clock recovery is based on the following equation:

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5)\, F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100\%$$

where M = receive margin
N = oversampling rate
L = frame length = 33
D = duty cycle = 0.6
F = oversampling clock deviation = Level II accuracy = 1000 in $10e^{-6}$

Figure 23.7 indicates what the receive margin M represents.

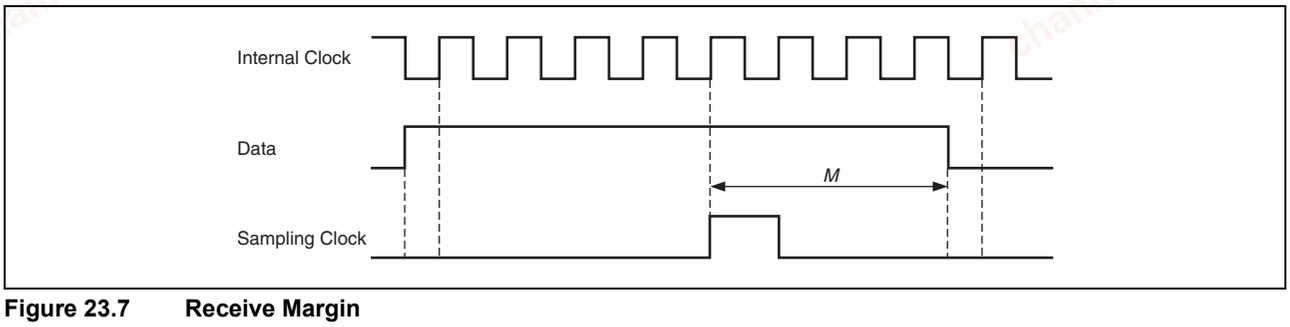


Figure 23.7 Receive Margin

Introducing jitter into the equation gives the following inequality.

$$j \leq \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5)\, F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100\%$$

J = clock jitter
Eight times oversampling produces a receive margin = 39.25%
Four times oversampling produces a receive margin = 31.75%
Two times oversampling produces a receive margin = 16.75%

The fastest sample frequency is 48 kHz. This requires a clock speed of 128 × 48 kHz = 6.144 MHz. The worst case jitter in one cycle is specified at 40 ns = 24.5% of the period. This means that an oversampling rate of 4 or more will satisfy the inequality and therefore be sufficient for clock recovery.

chanhvinh.com

Figure 23.8 illustrates the receiver data transfer using interrupts.

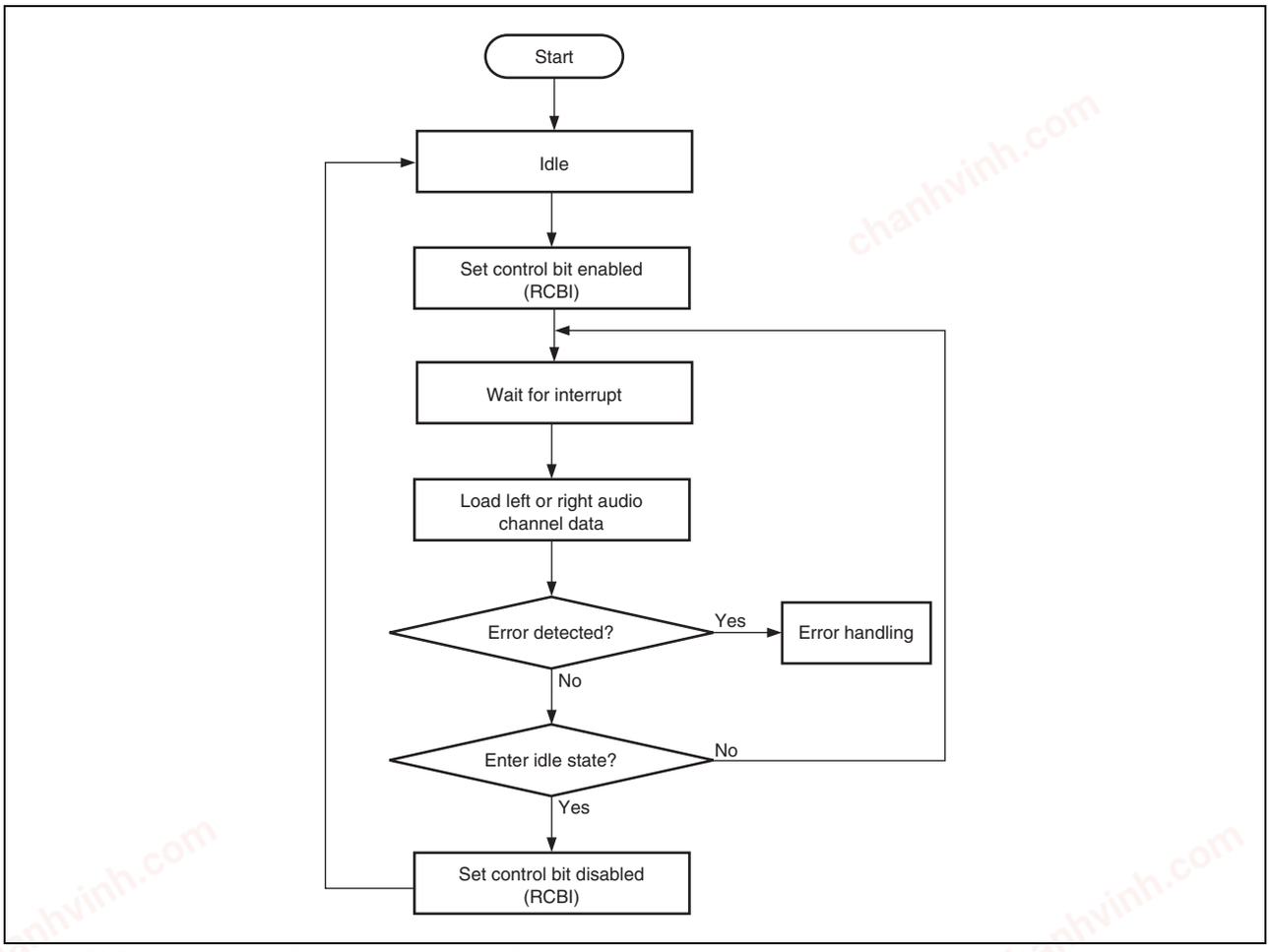


Figure 23.8 Receiver Data Transfer Flow Diagram - Interrupt Driven

Interrupts to indicate that the channel status information register is full occur after frame 30 has been received and only if the information has changed. When the first four bytes have been stored an interrupt occurs.

chanhvinh.com

23.10 Disabling the Module

23.10.1 Transmitter and Receiver Idle

The transmitter or receiver modules can be disabled by writing 0 to the idle bit in the control register (TME for the transmitter and RME for the receiver). The idle state can be detected by polling the idle bit in the status register (TIS and RIS).

23.11 Compressed Mode Data

Compressed mode data is defined in the IEC 61937 specification. This module only detects compressed mode data. This is done by checking the parity flag (V flag) and bit 1 in the channel status data. If both are one then the data is in compressed mode. This is indicated by the setting of the CMD bit in the status register.

Note: • Only the receiver detects compressed mode data since the information is not relevant to the transmitter.

23.12 References

IEC60958 Digital Audio Interface
IEC61937 Compressed Mode Digital Audio Interface

chanhvinh.com

23.13 Usage Notes

23.13.1 Clearing TUIR

After TUI is written to, the TUIR bit is cleared only after transmission of a maximum of one frame is completed. When using a transmitter user information interrupt to write data to TUI, check that the TUIR bit is cleared before terminating the interrupt handling routine so that the interrupt is not unexpectedly accepted again.

23.13.2 Frequency of Clock Input for Audio

The frequency of the clock input to the AUDIO_X1 and AUDIO_X2 or AUDIO_CLK must be lower than the Bϕ frequency.

chanhvinh.com

24. CD-ROM Decoder

This module is only incorporated in the RZ/A1L.

The CD-ROM decoder decodes streams of data transferred from the CD-DSP. When the medium is CD-DA*1, the data stream is not input to the CD-ROM decoder because it consists of PCM data. In the case of CD-ROM*2, the stream of data is input and the CD-ROM decoder performs sync code detection and maintenance, descrambling, ECC correction, and EDC checking, and outputs the resulting stream of data.

However, since the stream received by the CD-ROM decoder is assumed to consist of data from a CD-ROM transferred via the serial sound interface, the decoder does not bother with the subcodes defined in the CD-DA standard.

Note 1. Compliant with JIS S 8605 (Red Book)

Note 2. Compliant with JIS X 6281 (Yellow Book)

24.1 Features

- Sync-code detection and maintenance
  Detects sync codes from the CD-ROM and is capable of providing sync-code maintenance (automatic interpolation of sync codes) when the sync code cannot be detected because of defects such as scratches on the disc.
  Five sector-synchronization modes are supported: automatic sync maintenance mode, external sync mode, interpolated sync mode, and interpolated sync plus external sync mode.
- Descrambling
- ECC correction
  P-parity-based correction, Q-parity-based correction, PQ correction, and QP correction are available.
  PQ correction and QP correction can be applied repeatedly up to three times. This, however, depends on the speed of the CD. For example, three iterations are possible when the CD-ROM decoder is operating at 60 MHz with a double-speed CD drive.
  Two buffers are provided due to the need for ECC correction. This allows parallel operation, where ECC correction is performed in one buffer while the data stream is being received in the other.
- EDC checking
  The EDC is checked before and after ECC correction. An operating mode is available in which, if the result of pre-ECC correction EDC checking indicates no errors, ECC correction is not performed regardless of the result of syndrome calculation.
- Data buffering control
  The CD-ROM decoder outputs data to the buffer area in a specific format where the sync code is at the head of the data for each sector.

chanhvinh.com

24.1.1 Formats Supported by CD-ROM Decoder

This module supports the five formats shown in Figure 24.1.

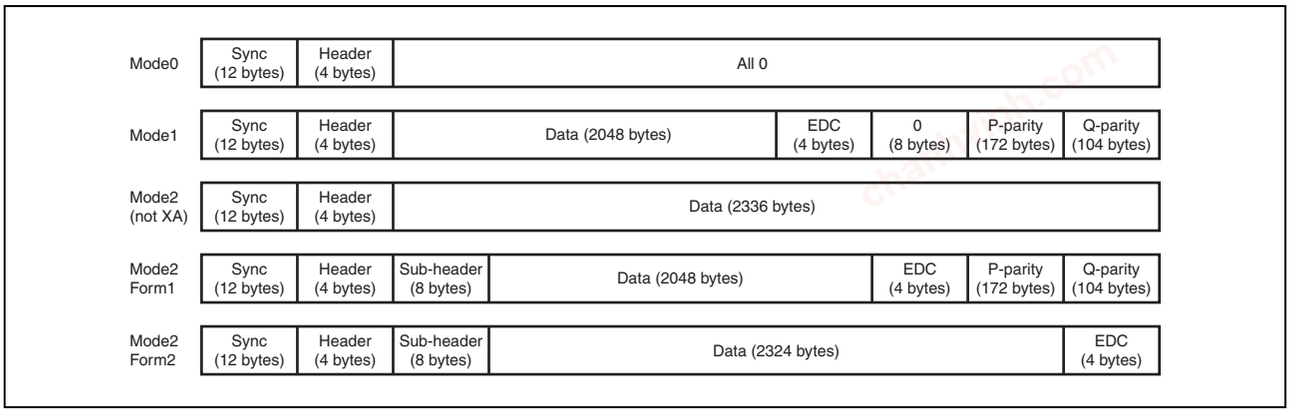


Figure 24.1 Formats Supported by CD-ROM Decoder

24.2 Block Diagrams

Figure 24.2 is a block diagram of the CD-ROM decoder functions of this LSI and the bus bridge for connection to the bus, that is, of the elements required to implement the CD-ROM decoder function.

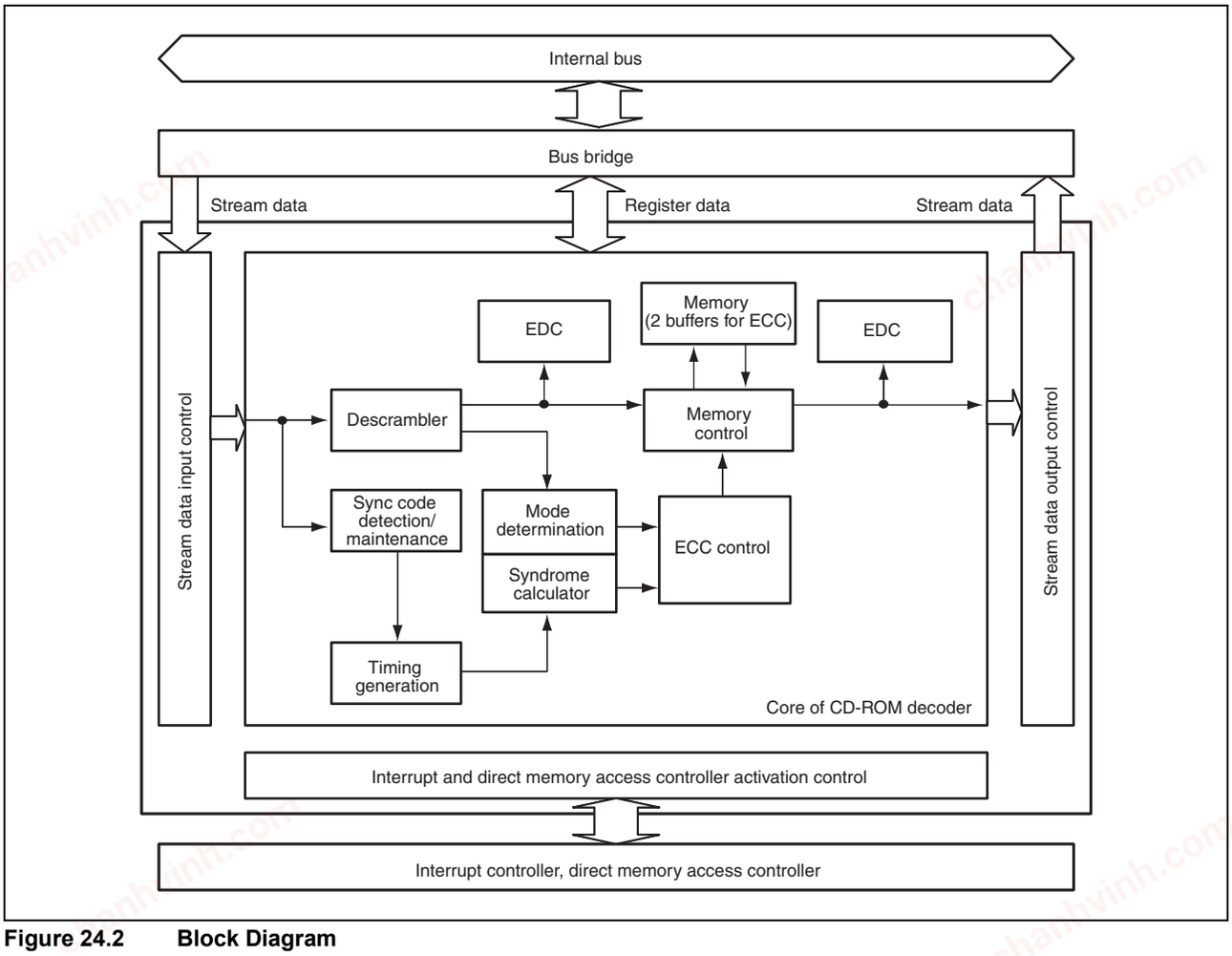


Figure 24.2 Block Diagram

chanhvinh.com

The core of the CD-ROM decoder executes a series of processing required for CD-ROM decoding, including descrambling, sync code detection, ECC correction (P- and Q-parity-based correction), and EDC checking. The core includes sufficient memory to hold two sectors of data.
Input data come from the internal bus and output data go out via the internal bus along a single line each, but the bus bridge logic sets up branches for the register access port and stream data port.
The stream data from the CD-DSP are transferred via the serial sound interface to the stream data input control block. They are then subjected to descrambling, ECC correction, and EDC checking as they pass through the CD-ROM decoder. After these processes, data from one sector are obtained. The data are subsequently transferred to the stream-data buffer via the stream-data output control block. Stream data can be transferred by either the direct memory access controller or the CPU.
Figure 24.3 is a block diagram of the bus-bridge logic.
Since the input stream is transferred over the serial sound interface, transfer is relatively slow. On the other hand, data from the output stream can be transferred at high speeds because they are already in the core of the CD-ROM decoder. Since the data for output are buffered in SDRAM or other memory, they must be transferred at high speeds in order to reduce the busy rate of the SDRAM. For this reason, the data for the output stream are read out before the CD-ROM decoder receives an output stream data read request from the internal bus. This allows the accumulation of streaming data in the registers of the bus bridge, so that the data are ready for immediate output to the internal bus upon a request from the internal bus. Accordingly, the reception of a request to read from registers other than the stream-data registers after the stream data has already been read out and stored in the register of the bus bridge is possible. To cope with this, the CD-ROM decoder is provided with separate intermediary registers for the output stream-data register and the other registers.

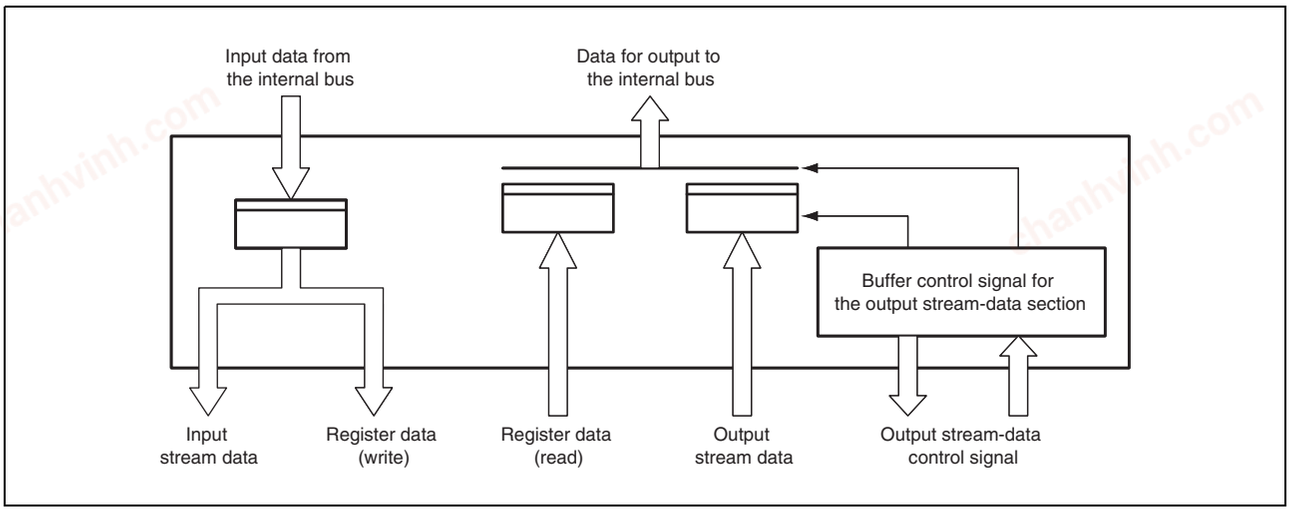


Figure 24.3 Schematic Diagram of the Bus Bridge

Figure 24.4 is a schematic diagram of the stream-data input control block. The stream-data input controller contains logic that controls the stream of input data and a register that is used to change the control mode of the CD-ROM decoder.
The serial sound interface mode used to transfer the stream data may affect the order (through the endian setting) or lead to padding before the data is transferred. To handle the different arrangements of data appropriately, the stream-data input control block includes a register for changing the operating mode and generates signals to control the core of the CD-ROM decoder.
The data holding registers for the input stream consists of two 16-bit registers. The data holding registers are controlled according to the mode set in the control register. For example, controlling the order in which 16-bit data is supplied to the core of the CD-ROM decoder (sending the second 16-bytes first or vice versa). It is also possible to stop the supply of padding data to the core of the CD-ROM decoder.

chanhvinh.com

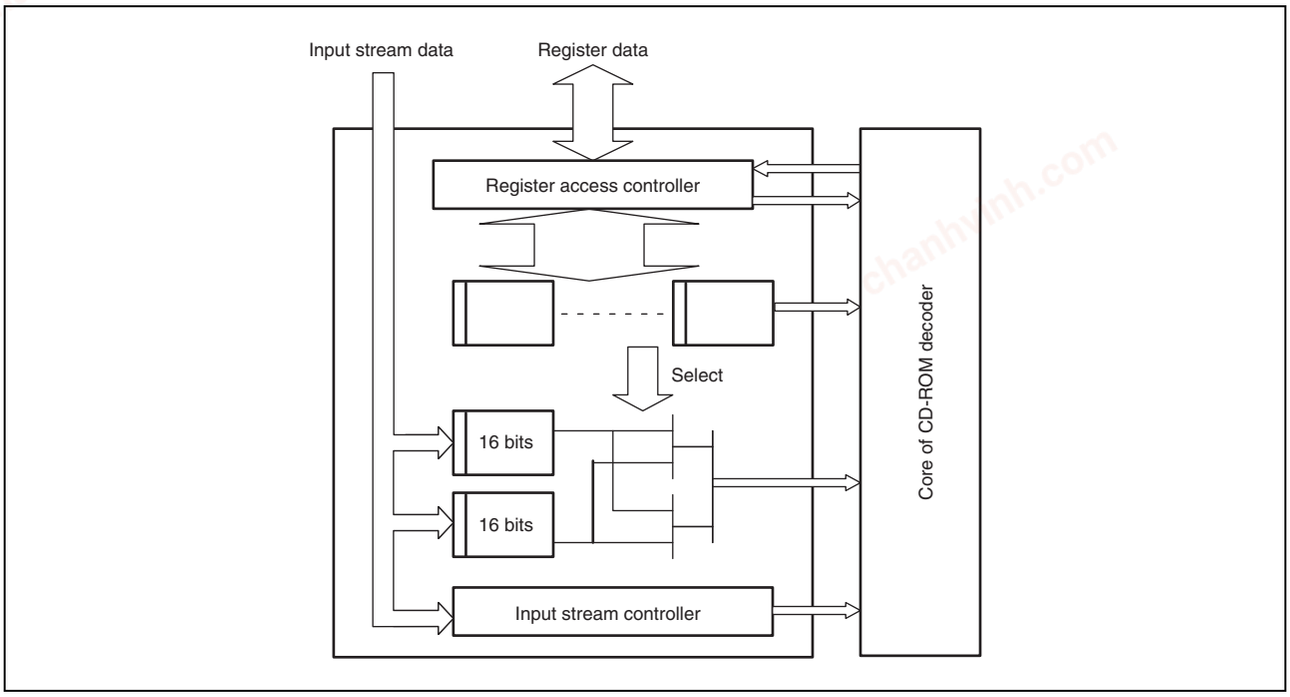


Figure 24.4 Schematic Diagram of the Stream-Data Input Control Block

Figure 24.5 is a schematic diagram of the stream-data output control block.
On recognizing that one sector of CD-ROM data is ready in the core of the CD-ROM decoder, this block ensures that the output stream-data register in the bus bridge section is empty and then starts to acquire the data for output from the core of the CD-ROM decoder.

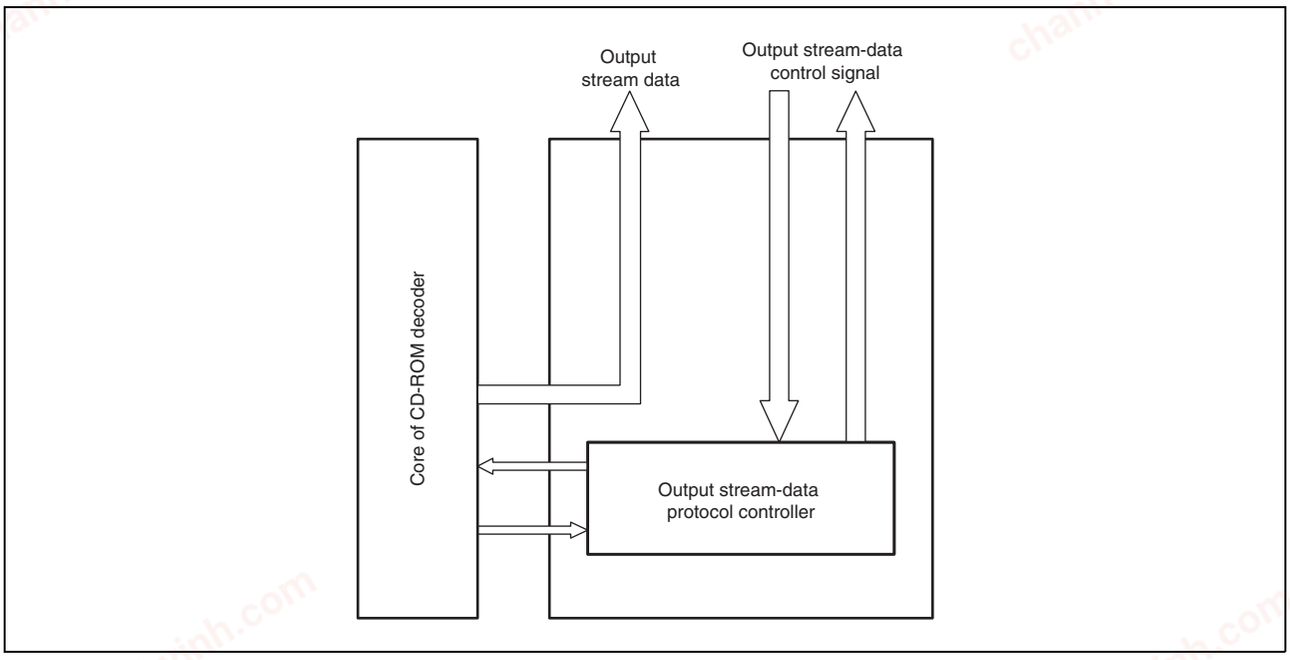


Figure 24.5 Schematic Diagram of the Stream-Data Output Control Block

This block has functions related to interrupts and direct memory access controller activation control such as suspending and masking of interrupts, turning interrupt flags off after they are read, asserting the activation signal to the direct

chanhvinh.com

memory access controller, and negating the activation signal according to the detected amount of data that has been transferred.

24.3 Register Descriptions

This module has the following registers.

Table 24.1 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Enable control register	CROMEN	R/W	H'00	0xE8005000	8
Sync code-based synchronization control register	CROMSY0	R/W	H'89	0xE8005001	8
Decoding mode control register	CROMCTL0	R/W	H'82	0xE8005002	8
EDC/ECC check control register	CROMCTL1	R/W	H'D1	0xE8005003	8
Automatic decoding stop control register	CROMCTL3	R/W	H'00	0xE8005005	8
Decoding option setting control register	CROMCTL4	R/W	H'00	0xE8005006	8
HEAD20 to HEAD22 representation control register	CROMCTL5	R/W	H'00	0xE8005007	8
Sync code status register	CROMST0	R	H'00	0xE8005008	8
Post-ECC header error status register	CROMST1	R	H'00	0xE8005009	8
Post-ECC subheader error status register	CROMST3	R	H'00	0xE800500B	8
Header/subheader validity check status register	CROMST4	R	H'00	0xE800500C	8
Mode determination and link sector detection status register	CROMST5	R	H'00	0xE800500D	8
ECC/EDC error status register	CROMST6	R	H'00	0xE800500E	8
Buffer status register	CBUFST0	R	H'00	0xE8005014	8
Decoding stoppage source status register	CBUFST1	R	H'00	0xE8005015	8
Buffer overflow status register	CBUFST2	R	H'00	0xE8005016	8
Pre-ECC correction header: minutes data register	HEAD00	R	H'00	0xE8005018	8
Pre-ECC correction header: seconds data register	HEAD01	R	H'00	0xE8005019	8
Pre-ECC correction header: frames (1/75 second) data register	HEAD02	R	H'00	0xE800501A	8
Pre-ECC correction header: mode data register	HEAD03	R	H'00	0xE800501B	8
Pre-ECC correction subheader: file number (byte 16) data register	SHEAD00	R	H'00	0xE800501C	8
Pre-ECC correction subheader: channel number (byte 17) data register	SHEAD01	R	H'00	0xE800501D	8
Pre-ECC correction subheader: sub-mode (byte 18) data register	SHEAD02	R	H'00	0xE800501E	8
Pre-ECC correction subheader: data type (byte 19) data register	SHEAD03	R	H'00	0xE800501F	8
Pre-ECC correction subheader: file number (byte 20) data register	SHEAD04	R	H'00	0xE8005020	8
Pre-ECC correction subheader: channel number (byte 21) data register	SHEAD05	R	H'00	0xE8005021	8
Pre-ECC correction subheader: sub-mode (byte 22) data register	SHEAD06	R	H'00	0xE8005022	8
Pre-ECC correction subheader: data type (byte 23) data register	SHEAD07	R	H'00	0xE8005023	8
Post-ECC correction header: minutes data register	HEAD20	R	H'00	0xE8005024	8
Post-ECC correction header: seconds data register	HEAD21	R	H'00	0xE8005025	8
Post-ECC correction header: frames (1/75 second) data register	HEAD22	R	H'00	0xE8005026	8
Post-ECC correction header: mode data register	HEAD23	R	H'00	0xE8005027	8
Post-ECC correction subheader: file number (byte 16) data register	SHEAD20	R	H'00	0xE8005028	8
Post-ECC correction subheader: channel number (byte 17) data register	SHEAD21	R	H'00	0xE8005029	8
Post-ECC correction subheader: sub-mode (byte 18) data register	SHEAD22	R	H'00	0xE800502A	8
Post-ECC correction subheader: data type (byte 19) data register	SHEAD23	R	H'00	0xE800502B	8

chanhvinh.com

Table 24.1 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Post-ECC correction subheader: file number (byte 20) data register	SHEAD24	R	H'00	0xE800502C	8
Post-ECC correction subheader: channel number (byte 21) data register	SHEAD25	R	H'00	0xE800502D	8
Post-ECC correction subheader: sub-mode (byte 22) data register	SHEAD26	R	H'00	0xE800502E	8
Post-ECC correction subheader: data type (byte 23) data register	SHEAD27	R	H'00	0xE800502F	8
Automatic buffering setting control register	CBUFCTL0	R/W	H'04	0xE8005040	8
Automatic buffering start sector setting: minutes control register	CBUFCTL1	R/W	H'00	0xE8005041	8
Automatic buffering start sector setting: seconds control register	CBUFCTL2	R/W	H'00	0xE8005042	8
Automatic buffering start sector setting: frames control register	CBUFCTL3	R/W	H'00	0xE8005043	8
ISY interrupt source mask control register	CROMST0M	R/W	H'00	0xE8005045	8
CD-ROM decoder reset control register	ROMDECRST	R/W	H'00	0xE8005100	8
CD-ROM decoder reset status register	RSTSTAT	R	H'00	0xE8005101	8
Serial sound interface data control register	SSI	R/W	H'18	0xE8005102	8
Interrupt flag register	INTHOLD	R/W	H'00	0xE8005108	8
Interrupt source mask control register	INHINT	R/W	H'00	0xE8005109	8
CD-ROM decoder stream data input register	STRMDIN0	R/W	H'0000	0xE8005200	R: 16 W: 16, 32
CD-ROM decoder stream data input register	STRMDIN2	R/W	H'0000	0xE8005202	16
CD-ROM decoder stream data output register	STRMDOUT0	R	H'0000	0xE8005204	16

chanhvinh.com

24.3.1 Enable Control Register (CROMEN)

The enable control register (CROMEN) enables subcode processing and CD-ROM decoding, and stops CD-ROM decoding forcibly.

Bit:	7	6	5	4	3	2	1	0
	SUBC_EN	CROM_EN	CROM_STP	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	SUBC_EN	0	R/W	Subcode Processing Enable This bit should be set and cleared simultaneously with CROM_EN. It is automatically cleared when decoding is automatically stopped due to an abnormal condition or when CROM_STP = 1
6	CROM_EN	0	R/W	CD-ROM Decoding Enable When this bit is set to 1, CD-ROM decoding starts after detection of a valid sync code. When the bit is cleared to 0, decoding stops on completion of the processing for the sector currently being decoded. This bit is automatically cleared to 0 when the automatic decode-stopping function woks or when CROM_STP = 1.
5	CROM_STP	0	R/W	Forcible Stop of CD-ROM Decoding When this bit is set to 1, CD-ROM decoding is stopped immediately and the SUBC_EN and CROM_EN bits are automatically reset to 0. Before decoding can resume, this bit must be cleared to 0.
4 to 0	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

24.3.2 Sync Code-Based Synchronization Control Register (CROMSY0)

The sync code-based synchronization control register (CROMSY0) selects the sync code maintenance function.

Bit:	7	6	5	4	3	2	1	0
	SY_AUT	SY_IEN	SY_DEN	-	-	-	-	-
Initial value:	1	0	0	0	1	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	SY_AUT	1	R/W	Automatic CD-ROM Sync Code Maintenance Mode When this bit is set to 1, automatic sync maintenance (insertion of sync codes) is applied to obtain the CD-ROM sync codes. While this bit is set, the settings of the SY_IEN and SY_DEN bits are invalid.
6	SY_IEN	0	R/W	Internal Sync Signal Enable Enables the internal sync signal that is produced by the counter in the CD-ROM decoder. When this bit is set to 1 while SY_AUT = 0, synchronization of the CD-ROM data is in interpolated mode, i.e. driven by the internal counter.
5	SY_DEN	0	R/W	Synchronization with External Sync Code Selects constant monitoring for the sync code in the input data and bases synchronization solely on detection of the code, regardless of the value of the internal counter. The setting of this bit is valid when SY_AUT = 0.
4	—	0	R/W	Reserved This bit is always read as 0. The write value should always be 0.
3	—	1	R/W	Reserved This bit is always read as 1. The write value should always be 1.
2, 1	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
0	—	1	R/W	Reserved This bit is always read as 1. The write value should always be 1.

Table 24.2 Register Settings for Sync Code Maintenance Function

SY_AUT	SY_IEN	SY_DEN	Operating Mode
1	—	—	Automatic sync maintenance mode
0	0	1	External sync mode
0	1	0	Interpolated sync mode
0	1	1	Interpolated sync plus external sync mode
0	0	0	Setting prohibited

chanhvinh.com

24.3.3 Decoding Mode Control Register (CROMCTL0)

The decoding mode control register (CROMCTL0) enables/disables the various functions, selects criteria for mode or form determination, and specifies the sector type. The setting of this register becomes valid at the sector-to-sector transition.

Bit:	7	6	5	4	3	2	1	0
	MD_DESC	-	MD_AUTO	MD_AUTOS1	MD_AUTOS2	MD_SEC[2:0]		
Initial value:	1	0	0	0	0	0	1	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	MD_DESC	1	R/W	Descrambling Function ON/OFF 0: Disables descrambling function 1: Enables descrambling function
6	—	0	R/W	Reserved This bit is always read as 0. The write value should always be 0.
5	MD_AUTO	0	R/W	Automatic Mode/Form Detection ON/OFF 0: OFF 1: ON Detectable formats are Mode 0, Mode 1, Mode 2 (non-XA), Mode 2 Form 1, and Mode 2 Form 2. If the mode and form cannot be detected, the mode and form of the previous sector is used. If the mode and form of the first sector after decoding starts is undetectable, the setting of the MD_SEC[2:0] bits is used as the initial value.
4	MD_AUTOS1	0	R/W	Criteria for Mode Determination when MD_AUTO = 1 0: Mode determination is made only when the sync code is detected 1: Mode determination is always made The setting of this bit is valid only when the MD_AUTO bit is 1. If the mode cannot be determined, the mode of the previous sector is used. When this bit is cleared to 0, mode determination is made only when the sync code is detected for the sector.
3	MD_AUTOS2	0	R/W	Criteria for Mode 2 Form Determination when MD_AUTO = 1 0: The sector is assumed to be non-XA if the two form code bytes in the subheader do not match 1: No determination of XA or non-XA for the sector. The first form byte is regarded as valid. However, the two form bytes are compared, and the result is reflected in a status bit. The setting of this bit is valid only when the MD_AUTO bit is 1.
2 to 0	MD_SEC[2:0]	010	R/W	Sector Type 000: Setting prohibited 001: Mode 0 010: Mode 1 011: Long (Mode 0, Mode 1, or Mode 2 with no EDC/ECC data) 100: Setting prohibited 101: Mode 2 Form 1 110: Mode 2 Form 2 111: Mode 2 with automatic form detection If the form cannot be determined when set to B'111, it is processed as Mode 2 not XA.

chanhvinh.com

24.3.4 EDC/ECC Check Control Register (CROMCTL1)

The EDC/ECC check control register (CROMCTL1) controls EDC/ECC checking. The setting of this register becomes valid at the sector-to-sector transition.

Bit:	7	6	5	4	3	2	1	0
	M2F2 EDC	MD_DEC[2:0]			-	-	MD_PQREP[1:0]	
Initial value:	1	1	0	1	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	M2F2EDC	1	R/W	For Mode 2 Form 2, disables the EDC function for sectors where all bits of the EDC are 0. When this bit set to 1 and all bits of the EDC for a Mode 2 Form 2 sector are 0, an IERR interrupt is not generated even if the result of EDC checking is 'fail'.
6 to 4	MD_DEC[2:0]	101	R/W	EDC/ECC Checking Mode Select 000: No checking 001: EDC only 010: Q correction + EDC 011: P correction + EDC 100: QP correction + EDC 101: PQ correction + EDC 110: Setting prohibited 111: Setting prohibited
3, 2	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	MD_PQREP[1:0]	01	R/W	Number of Iterations of PQ or QP Correction Number of correction iterations when PQ- or QP- correction is specified by MD_DEC[2:0]. 00: Setting prohibited 01: One iteration 10: Two iterations 11: Three iterations

chanhvinh.com

24.3.5 Automatic Decoding Stop Control Register (CROMCTL3)

The automatic decoding stop control register (CROMCTL3) is used to select abnormal conditions on which decoding will be automatically stopped. When decoding is stopped in response to any of the selected conditions, an IBUF interrupt is generated and the condition is indicated in the CBUFST1 register. The setting of this register becomes valid at the sector-to-sector transition.

Bit:	7	6	5	4	3	2	1	0
	STP_ECC	STP_EDC	-	STP_MD	STP_MIN	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	STP_ECC	0	R/W	When this bit is set to 1, decoding is stopped if an error is found to be not correctable by ECC correction.
6	STP_EDC	0	R/W	When this bit is set to 1, decoding is stopped if post-ECC correction EDC checking indicates an error.
5	—	0	R/W	Reserved This bit is always read as 0. The write value should always be 0.
4	STP_MD	0	R/W	When this bit is set to 1, decoding is stopped if the sector has a mode or form setting that does not match those of the immediately preceding sector.
3	STP_MIN	0	R/W	When this bit is set to 1, decoding is stopped if a non-sequential minutes, seconds, or frames (1 frame = 1/75 second) value is encountered.
2 to 0	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

24.3.6 Decoding Option Setting Control Register (CROMCTL4)

The decoding option setting control register (CROMCTL4) enables/disables buffering control at link block detection, specifies the information indicated by the status register, and controls the ECC correction mode. The setting of this register becomes valid at the sector-to-sector transition.

Bit:	7	6	5	4	3	2	1	0
	-	LINK2	-	ER0SEL	NO_ECC	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved Either 0 or 1 can be written to this bit. When it is read, the value written will be read.
6	LINK2	0	R/W	Link Block Detection Condition 0: The block is regarded as a link block when either run-out 1 or 2 and both run-in 3 and 4 have been detected. 1: The block is regarded as a link block when two out of run-out 1 and 2 and “link” have been detected. The condition for setting of the LINK_ON bit in CROMST5 is decoding of the link sector.
5	—	0	R/W	Reserved This bit is always read as 0. The write value should always be 0.
4	ER0SEL	0	R/W	CD-ROM Data-Related Status Register Setting Condition 0: Information is on the sector being decoded. 1: Information is on the latest sector that has been buffered. This condition affects the information given by bits 5 to 0 in the CROMST0 register, bits 7 to 1 in the CROMST4 and CROMST5 registers, and HEAD00 to HEAD02.
3	NO_ECC	0	R/W	ECC correction mode when the result of the EDC check before ECC correction was ‘pass’ When this bit is set to 1, ECC correction is not performed if the result of pre-ECC correction EDC checking is a ‘pass’, regardless of the results of syndrome calculation.
2 to 0	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

24.3.7 HEAD20 to HEAD22 Representation Control Register (CROMCTL5)

The HEAD20 to HEAD22 representation control register (CROMCTL5) specifies the representation mode for HEAD20 to HEAD22.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	MSF_LBA_SEL
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
0	MSF_LBA_SEL	0	R/W	HEAD20 to HEAD22 Representation Mode 0: Header MSF is represented in BCD (decimal) as is 1: Total sector number is represented in HEX (hexadecimal)

24.3.8 Sync Code Status Register (CROMST0)

The sync code status register (CROMST0) indicates various status information in sync code maintenance modes.

Bit:	7	6	5	4	3	2	1	0
	-	-	ST_SYIL	ST_SYNO	ST_BLKS	ST_BLKL	ST_SECS	ST_SECL
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
5	ST_SYIL	0	R	Indicates that a sync code was detected at a position where the value in the word counter (used to measure intervals between sync codes) was not correct, but the sync code was ignored and not taken into account in synchronization. This bit is only valid in automatic sync maintenance mode and interpolated sync mode.
4	ST_SYNO	0	R	Indicates that a sync code has not been detected despite the word counter having reached the final value, and synchronization has been continued with the aid of an interpolated sync code. This bit is only valid in automatic sync maintenance mode and interpolated sync mode.
3	ST_BLKS	0	R	Indicates that a sync code was detected at a position where the value in the word counter was not correct, and the sync code was used in synchronization. This bit is only valid in automatic sync maintenance mode and external sync mode.
2	ST_BLKL	0	R	Indicates that a sync code has not been detected despite the word counter having reached the final value, and the period of the sector has been prolonged. This bit is only valid in external sync mode.
1	ST_SECS	0	R	Indicates that a sector has been processed as a short sector with the aid of interpolated sync codes. If this bit is set to 1, stop decoding immediately and retry the procedure starting from the sector prior to the sector currently being decoded.
0	ST_SECL	0	R	Indicates that a sector has been processed as a long sector with the aid of interpolated sync codes. If this bit is set to 1, stop decoding immediately and retry the procedure starting from two sectors prior to the sector currently being decoded.

chanhvinh.com

24.3.9 Post-ECC Header Error Status Register (CROMST1)

The post-ECC header error status register (CROMST1) indicates error status in the post-ECC header.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	ER2_HEAD0	ER2_HEAD1	ER2_HEAD2	ER2_HEAD3
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
3	ER2_HEAD0	0	R	Indicates an error in the minutes field of the header after ECC correction.
2	ER2_HEAD1	0	R	Indicates an error status in the seconds field of the header after ECC correction.
1	ER2_HEAD2	0	R	Indicates an error in the frames (1 frame = 1/75 second) field of the header after ECC correction.
0	ER2_HEAD3	0	R	Indicates an error in the mode field of the header after ECC correction.

24.3.10 Post-ECC Subheader Error Status Register (CROMST3)

The post-ECC subheader error status register (CROMST3) indicates error status in the post-ECC subheader.

Bit:	7	6	5	4	3	2	1	0
	ER2_SHEAD0	ER2_SHEAD1	ER2_SHEAD2	ER2_SHEAD3	ER2_SHEAD4	ER2_HEAD5	ER2_HEAD6	ER2_HEAD7
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	ER2_SHEAD0	0	R	Indicates that the subheader (file number) still has an error after ECC correction. Indicates the error of the SHEAD20 register.
6	ER2_SHEAD1	0	R	Indicates that the subheader (channel number) still has an error after ECC correction. Indicates the error of the SHEAD21 register.
5	ER2_SHEAD2	0	R	Indicates that the subheader (sub-mode) still has an error after ECC correction. Indicates the error of the SHEAD22 register.
4	ER2_SHEAD3	0	R	Indicates that the subheader (data type) still has an error after ECC correction. Indicates the error of the SHEAD23 register.
3	ER2_SHEAD4	0	R	Indicates that the subheader (file number) still has an error after ECC correction. Indicates the error of the SHEAD24 register.
2	ER2_SHEAD5	0	R	Indicates that the subheader (channel number) still has an error after ECC correction. Indicates the error of the SHEAD25 register.
1	ER2_SHEAD6	0	R	Indicates that the subheader (sub-mode) still has an error after ECC correction. Indicates the error of the SHEAD26 register.
0	ER2_SHEAD7	0	R	Indicates that the subheader (data type) still has an error after ECC correction. Indicates the error of the SHEAD27 register.

chanhvinh.com

24.3.11 Header/Subheader Validity Check Status Register (CROMST4)

The header/subheader validity check status register (CROMST4) indicates errors relating to the automatic mode determination or form determination for Mode 2.

Bit:	7	6	5	4	3	2	1	0
	NG_MD	NG_MDCMP1	NG_MDCMP2	NG_MDCMP3	NG_MDCMP4	NG_MDDEF	NG_MDTIM1	NG_MDTIM2
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	NG_MD	0	R	Indicates that the sector mode could not be determined according to the automatic mode determination criteria.
6	NG_MDCMP1	0	R	Indicates a mismatch between the file number bytes (bytes 16 and 20) during the form determination for Mode 2.
5	NG_MDCMP2	0	R	Indicates a mismatch between the channel number bytes (bytes 17 and 21) during the form determination for Mode 2.
4	NG_MDCMP3	0	R	Indicates a mismatch between the sub-mode bytes (bytes 18 and 22) during the form determination for Mode 2.
3	NG_MDCMP4	0	R	Indicates a mismatch between the data-type bytes (bytes 19 and 23) during the form determination for Mode 2.
2	NG_MDDEF	0	R	Indicates that the mode and form differ from those of the previous sector.
1	NG_MDTIM1	0	R	Indicates that the minutes, seconds, or frames (1 frame = 1/75 second) value is out of sequence. In the continuity check for the next and subsequent sectors, the updated values will be used.
0	NG_MDTIM2	0	R	Indicates that the minutes, seconds, or frames (1 frame = 1/75 second) value was not a BCD value. Specifically, this bit means that any half-byte was beyond the range for BCD (i.e. was A to F), HEAD01 was greater than H'59, or HEAD02 was greater than H'74. In the continuity check for the next and subsequent sectors, interpolated values will be used.

chanhvinh.com

24.3.12 Mode Determination and Link Sector Detection Status Register (CROMST5)

The mode determination and link sector detection status register (CROMST5) indicates the result of automatic mode determination and link block detection.

Bit:	7	6	5	4	3	2	1	0
	ST_AMD[2:0]			ST_MDX	LINK_ON	LINK_ DET	LINK_ SDET	LINK_ OUT1
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	ST_AMD[2:0]	000	R	Result of Automatic Mode Determination These bits indicate the result of mode determination when the automatic mode determination function is used. 000: Automatic mode determination function is not used 001: Mode 0 010: Mode 1 011: — 100: Mode 2 not XA 101: Mode 2 Form 1 110: Mode 2 Form 2 111: —
4	ST_MDX	0	R	Indicates that, when the mode has been manually set rather than automatically determined, the mode setting disagrees with the mode as recognized by the logic. In this case, the manually set value takes priority.
3	LINK_ON	0	R	This bit is set to 1 when a link block was recognized in link block determination. For the criteria for link block determination, refer to the LINK2 bit in the CROMCTL4 register.
2	LINK_DET	0	R	Indicates that a link block (run-out 1 to run-in 4) was detected. Since detection is based on the data before ECC correction, LINK_DET may also be set to 1 if data erroneously happens to contain the same code as a link block.
1	LINK_SDET	0	R	Indicates that a link block was detected within seven sectors after the start of decoding.
0	LINK_OUT1	0	R	Indicates that the sector after ECC correction has been identified as a run-out 1 sector. This bit is only valid when an IERR interrupt is not generated (i.e. when ECC correction was successful).

chanhvinh.com

24.3.13 ECC/EDC Error Status Register (CROMST6)

The ECC/EDC error status register (CROMST6) indicates ECC processing error or EDC check error before/after ECC correction.

Bit:	7	6	5	4	3	2	1	0
	ST_ERR	-	ST_ECCABT	ST_ECCNG	ST_ECCP	ST_ECCQ	ST_EDC1	ST_EDC2
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	ST_ERR	0	R	Indicates that the decoded block after ECC correction contains any error (even in a single byte).
6	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
5	ST_ECCABT	0	R	Indicates that ECC processing was discontinued. This bit is set to 1 when a transition from sector to sector occurs while ECC correction is in progress. This does not indicate a problem for ECC correction if the BUF_NG bit in the CBUFST2 register is 0 at the same time. Whether or not this is so depends on the timing of the sector transition.
4	ST_ECCNG	0	R	Indicates that error correction was not possible. This bit is also set to 1 on detection of a short sector.
3	ST_ECCP	0	R	Indicates that P-parity errors were not corrected in ECC correction. This bit is only valid when synchronization is normal (the sector is neither short nor long). This bit is set to 1 when the result of syndrome calculation for P parity is other than all 0s.
2	ST_ECCQ	0	R	Indicates that Q-parity errors were not corrected in ECC correction. This bit is only valid when synchronization is normal (the sector is neither short nor long). This bit is set to 1 when the result of syndrome calculation for Q parity is other than all 0s.
1	ST_EDC1	0	R	Indicates that the result of the EDC check before ECC correction was ‘fail’. This bit is also set to 1 if a short sector is encountered while EDC is enabled.
0	ST_EDC2	0	R	Indicates that the result of the EDC check after ECC correction was ‘fail’.

24.3.14 Buffer Status Register (CBUFST0)

The buffer status register (CBUFST0) indicates that the system is searching for the first sector to be buffered, or that buffering is in progress.

Bit:	7	6	5	4	3	2	1	0
	BUF_REF	BUF_ACT	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	BUF_REF	0	R	Indicates that the search for the first sector to be buffered is in progress. This bit is only valid when the automatic buffering function is used (CBUF_AUT = 1).
6	BUF_ACT	0	R	Indicates that buffering is in progress.
5 to 0	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.

chanhvinh.com

24.3.15 Decoding Stoppage Source Status Register (CBUFST1)

The decoding stoppage source status register (CBUFST1) indicates that decoding/buffering has been stopped due to some errors.

A bit in this register can only be set when the corresponding bit in the CROMCTL3 register is set to 1.

Bit:	7	6	5	4	3	2	1	0
	BUF_ECC	BUF_EDC	-	BUF_MD	BUF_MIN	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	BUF_ECC	0	R	Indicates that decoding and buffering have been stopped because of an error that is not correctable by using the ECC.
6	BUF_EDC	0	R	Indicates that decoding and buffering have been stopped because the post-ECC correction EDC check indicated an error.
5	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
4	BUF_MD	0	R	Indicates that decoding and buffering have been stopped because the current sector is in a mode or form differing from that of the previous sector.
3	BUF_MIN	0	R	Indicates that decoding and buffering have been stopped because a non-sequential minutes, seconds, or frames (1 frame = 1/75 second) value has been encountered.
2 to 0	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.

24.3.16 Buffer Overflow Status Register (CBUFST2)

The buffer overflow status register (CBUFST2) indicates that a sector-to-sector transition occurred before data transfer to the buffer is completed.

Bit:	7	6	5	4	3	2	1	0
	BUF_NG	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	BUF_NG	0	R	Indicates that a sector-to-sector transition has occurred before the data transfer to the buffer is completed. This bit is set to 1 when the data of a third sector are input while data for the output stream from the CD-ROM decoder remains unread. No interrupt is generated. Once this bit has been set to 1, its value will not recover unless it is reset by the LOGICRST bit in the ROMDECRST register.
6 to 0	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.

chanhvinh.com

24.3.17 Pre-ECC Correction Header: Minutes Data Register (HEAD00)

The pre-ECC correction header: minutes data register (HEAD00) indicates the minutes value in the header before ECC correction.

Bit:	7	6	5	4	3	2	1	0
	HEAD00[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	HEAD00[7:0]	All 0	R	Minutes Value in Header before ECC Correction

24.3.18 Pre-ECC Correction Header: Seconds Data Register (HEAD01)

The pre-ECC correction header: seconds data register (HEAD01) indicates the seconds value in the header before ECC correction.

Bit:	7	6	5	4	3	2	1	0
	HEAD01[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	HEAD01[7:0]	All 0	R	Seconds Value in Header before ECC Correction

24.3.19 Pre-ECC Correction Header: Frames (1/75 Second) Data Register (HEAD02)

The pre-ECC correction header: frames (1/75 second) data register (HEAD02) indicates the frames value (1 frame = 1/75 second) in the header before ECC correction.

Bit:	7	6	5	4	3	2	1	0
	HEAD02[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	HEAD02[7:0]	All 0	R	Frames Value (1 frame = 1/75 second) in Header before ECC Correction

24.3.20 Pre-ECC Correction Header: Mode Data Register (HEAD03)

The pre-ECC correction header: mode data register (HEAD03) indicates the mode value in the header before ECC correction.

Bit:	7	6	5	4	3	2	1	0
	HEAD03[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	HEAD03[7:0]	All 0	R	Mode Value in Header before ECC Correction

chanhvinh.com

24.3.21 Pre-ECC Correction Subheader: File Number (Byte 16) Data Register (SHEAD00)

The pre-ECC correction subheader: file number (byte 16) data register (SHEAD00) indicates the file number value in the subheader before ECC correction (byte 16).

Bit:	7	6	5	4	3	2	1	0
	SHEAD00[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SHEAD00[7:0]	All 0	R	Indicates file number value in the subheader before ECC correction (byte 16). For sectors not in Mode 2, this register contains the byte of data at the corresponding position.

24.3.22 Pre-ECC Correction Subheader: Channel Number (Byte 17) Data Register (SHEAD01)

The pre-ECC correction subheader: channel number (byte 17) data register (SHEAD01) indicates the channel number value in the subheader before ECC correction (byte 17).

Bit:	7	6	5	4	3	2	1	0
	SHEAD01[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SHEAD01[7:0]	All 0	R	Indicates channel number value in the subheader before ECC correction (byte 17). For sectors not in Mode 2, this register contains the byte of data at the corresponding position.

24.3.23 Pre-ECC Correction Subheader: Sub-Mode (Byte 18) Data Register (SHEAD02)

The pre-ECC correction subheader: sub-mode (byte 18) data register (SHEAD02) indicates the sub-mode value in the subheader before ECC correction (byte 18).

Bit:	7	6	5	4	3	2	1	0
	SHEAD02[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SHEAD02[7:0]	All 0	R	Indicates sub-mode value in the subheader before ECC correction (byte 18). For sectors not in Mode 2, this register contains the byte of data at the corresponding position.

chanhvinh.com

24.3.24 Pre-ECC Correction Subheader: Data Type (Byte 19) Data Register (SHEAD03)

The pre-ECC correction subheader: data type (byte 19) data register (SHEAD03) indicates the data type value in the subheader before ECC correction (byte 19).

Bit:	7	6	5	4	3	2	1	0
	SHEAD03[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SHEAD03[7:0]	All 0	R	Indicates data type value in the subheader before ECC correction (byte 19). For sectors not in Mode 2, this register contains the byte of data at the corresponding position.

24.3.25 Pre-ECC Correction Subheader: File Number (Byte 20) Data Register (SHEAD04)

The pre-ECC correction subheader: file number (byte 20) data register (SHEAD04) indicates the file number value in the subheader before ECC correction (byte 20).

Bit:	7	6	5	4	3	2	1	0
	SHEAD04[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SHEAD04[7:0]	All 0	R	Indicates file number value in the subheader before ECC correction (byte 20). For sectors not in Mode 2, this register contains the byte of data at the corresponding position.

24.3.26 Pre-ECC Correction Subheader: Channel Number (Byte 21) Data Register (SHEAD05)

The pre-ECC correction subheader: channel number (byte 21) data register (SHEAD05) indicates the channel number value in the subheader before ECC correction (byte 21).

Bit:	7	6	5	4	3	2	1	0
	SHEAD05[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SHEAD05[7:0]	All 0	R	Indicates channel number value in the subheader before ECC correction (byte 21). For sectors not in Mode 2, this register contains the byte of data at the corresponding position.

chanhvinh.com

24.3.27 Pre-ECC Correction Subheader: Sub-Mode (Byte 22) Data Register (SHEAD06)

The pre-ECC correction subheader: sub-mode (byte 22) data register (SHEAD06) indicates the sub-mode value in the subheader before ECC correction (byte 22).

Bit:	7	6	5	4	3	2	1	0
	SHEAD06[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SHEAD06[7:0]	All 0	R	Indicates sub-mode value in the subheader before ECC correction (Byte 22). For sectors not in Mode 2, this register contains the byte of data at the corresponding position.

24.3.28 Pre-ECC Correction Subheader: Data Type (Byte 23) Data Register (SHEAD07)

The pre-ECC correction subheader: data type (byte 23) data register (SHEAD07) indicates the data type value in the subheader before ECC correction (byte 23).

Bit:	7	6	5	4	3	2	1	0
	SHEAD07[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SHEAD07[7:0]	All 0	R	Indicates data type value in the subheader before ECC correction (Byte 23). For sectors not in Mode 2, this register contains the byte of data at the corresponding position.

24.3.29 Post-ECC Correction Header: Minutes Data Register (HEAD20)

The post-ECC correction header: minutes data register (HEAD20) indicates the minutes value in the header after ECC correction.

Bit:	7	6	5	4	3	2	1	0
	HEAD20[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	HEAD20[7:0]	All 0	R	Indicates minutes value in the header after ECC correction. When MSF_LBA_SEL = 1, this register indicates the first byte (1/3) of the total number of sectors calculated from M, S, and F.

chanhvinh.com

24.3.30 Post-ECC Correction Header: Seconds Data Register (HEAD21)

The post-ECC correction header: seconds data register (HEAD21) indicates the seconds value in the header after ECC correction.

Bit:	7	6	5	4	3	2	1	0
	HEAD21[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	HEAD21[7:0]	All 0	R	Indicates seconds value in the header after ECC correction. When MSF_LBA_SEL = 1, this register indicates the second byte (2/3) of the total number of sectors calculated from M, S, and F.

24.3.31 Post-ECC Correction Header: Frames (1/75 Second) Data Register (HEAD22)

The post-ECC correction header: frames (1/75 second) data register (HEAD22) indicates the frames value (1 frame = 1/75 seconds) in the header after ECC correction.

Bit:	7	6	5	4	3	2	1	0
	HEAD22[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	HEAD22[7:0]	All 0	R	Indicates frames value in the header after ECC correction. When MSF_LBA_SEL = 1, this register indicates the third byte (3/3) of the total number of sectors calculated from M, S, and F.

24.3.32 Post-ECC Correction Header: Mode Data Register (HEAD23)

The post-ECC correction header: mode data register (HEAD23) indicates the mode value in the header after ECC correction.

Bit:	7	6	5	4	3	2	1	0
	HEAD23[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	HEAD23[7:0]	All 0	R	Indicates mode value in the header after ECC correction.

chanhvinh.com

24.3.33 Post-ECC Correction Subheader: File Number (Byte 16) Data Register (SHEAD20)

The post-ECC correction subheader: file number (byte 16) data register (SHEAD20) indicates the file number value in the subheader after ECC correction (byte 16).

Bit:	7	6	5	4	3	2	1	0
	SHEAD20[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SHEAD20[7:0]	All 0	R	Indicates file number value in the subheader after ECC correction (byte 16).

24.3.34 Post-ECC Correction Subheader: Channel Number (Byte 17) Data Register (SHEAD21)

The post-ECC correction subheader: channel number (byte 17) data register (SHEAD21) indicates the channel number value in the subheader after ECC correction (byte 17).

Bit:	7	6	5	4	3	2	1	0
	SHEAD21[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SHEAD21[7:0]	All 0	R	Indicates channel number value in the subheader after ECC correction (byte 17).

24.3.35 Post-ECC Correction Subheader: Sub-Mode (Byte 18) Data Register (SHEAD22)

The post-ECC correction subheader: sub-mode (byte 18) data register (SHEAD22) indicates the sub-mode value in the subheader after ECC correction (byte 18).

Bit:	7	6	5	4	3	2	1	0
	SHEAD22[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SHEAD22[7:0]	All 0	R	Indicates sub-mode value in the subheader after ECC correction (byte 18).

chanhvinh.com

24.3.36 Post-ECC Correction Subheader: Data Type (Byte 19) Data Register (SHEAD23)

The post-ECC correction subheader: data type (byte 19) data register (SHEAD23) indicates the data type value in the subheader after ECC correction (byte 19).

Bit:	7	6	5	4	3	2	1	0
	SHEAD23[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SHEAD23[7:0]	All 0	R	Indicates data type value in the subheader after ECC correction (byte 19).

24.3.37 Post-ECC Correction Subheader: File Number (Byte 20) Data Register (SHEAD24)

The post-ECC correction subheader: file number (byte 20) data register (SHEAD24) indicates the file number value in the subheader after ECC correction (byte 20).

Bit:	7	6	5	4	3	2	1	0
	SHEAD24[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SHEAD24[7:0]	All 0	R	Indicates file number value in the subheader after ECC correction (byte 20).

24.3.38 Post-ECC Correction Subheader: Channel Number (Byte 21) Data Register (SHEAD25)

The post-ECC correction subheader: channel number (byte 21) data register (SHEAD25) indicates the channel number value in the subheader after ECC correction (byte 21).

Bit:	7	6	5	4	3	2	1	0
	SHEAD25[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SHEAD25[7:0]	All 0	R	Indicates channel number value in the subheader after ECC correction (byte 21).

chanhvinh.com

24.3.39 Post-ECC Correction Subheader: Sub-Mode (Byte 22) Data Register (SHEAD26)

The post-ECC correction subheader: sub-mode (byte 22) data register (SHEAD26) indicates the sub-mode value in the subheader after ECC correction (byte 22).

Bit:	7	6	5	4	3	2	1	0
	SHEAD26[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SHEAD26[7:0]	All 0	R	Indicates sub-mode value in the subheader after ECC correction (byte 22).

24.3.40 Post-ECC Correction Subheader: Data Type (Byte 23) Data Register (SHEAD27)

The post-ECC correction subheader: data type (byte 23) data register (SHEAD27) indicates the data type value in the subheader after ECC correction (byte 23).

Bit:	7	6	5	4	3	2	1	0
	SHEAD27[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SHEAD27[7:0]	All 0	R	Indicates data type value in the subheader after ECC correction (byte 23).

chanhvinh.com

24.3.41 Automatic Buffering Setting Control Register (CBUFCTL0)

Bit:	7	6	5	4	3	2	1	0
	CBUF_AUT	CBUF_EN	-	CBUF_MD[1:0]		CBUF_TS	CBUF_Q	-
Initial value:	0	0	0	0	0	1	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	CBUF_AUT	0	R/W	Automatic Buffering Function ON/OFF When this bit is to be set or cleared while CROM_EN = 1, CBUF_EN should also be set or cleared simultaneously. Otherwise, the validity of the status indications in CBUFST0, CBUFST1 and CBUFST2 cannot be guaranteed. 0: Automatic buffering is OFF. 1: Automatic buffering is ON.
6	CBUF_EN	0	R/W	Buffering to Buffer RAM Enable This bit turns on/off buffering in both automatic and manual buffering modes. In manual buffering mode, set this bit after generation of the ISEC interrupt. This bit is automatically reset when automatic buffering stops. 0: Buffering is OFF. 1: Buffering is ON.
5	—	0	R/W	Reserved This bit is always read as 0. The write value should always be 0.
4, 3	CBUF_MD[1:0]	00	R/W	Start-sector detection mode when the automatic buffering function is in use 00: The header values for the previous and current sectors must be in sequence. 01: The header value detected in the current sector must be in sequence with the interpolated value. 10: A current sector with any header value is OK. 11: Start-sector detection is based on the interpolated value even if the current sector is not detected.
2	CBUF_TS	1	R/W	CBUFCTL1 to CBUFCTL3 Setting Mode 0: CBUFCTL1 to CBUFCTL3: BCD (in decimal) 1: Total number of sectors (in hexadecimal)
1	CBUF_Q	0	R/W	Q-channel code buffering data specification in the case of a CRC error in the Q-channel code 0: The values for the last sector for which the CRC returned a correct result are buffered. 1: The erroneous data is buffered as is. Note: Since subcodes are not input with this LSI, always set this bit to 1.
0	—	0	R/W	Reserved This bit is always read as 0. The write value should always be 0.

chanhvinh.com

24.3.42 Automatic Buffering Start Sector Setting: Minutes Control Register (CBUFCTL1)

The automatic buffering start sector setting: minutes control register (CBUFCTL1) indicates the minutes value in the header for the first sector to be buffered.

Bit:	7	6	5	4	3	2	1	0
	BS_MIN[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	BS_MIN[7:0]	All 0	R/W	Indicates setting of the minutes value in the header for the first sector to be buffered.

24.3.43 Automatic Buffering Start Sector Setting: Seconds Control Register (CBUFCTL2)

The automatic buffering start sector setting: seconds control register (CBUFCTL2) indicates the seconds value in the header for the first sector to be buffered.

Bit:	7	6	5	4	3	2	1	0
	BS_SEC[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	BS_SEC[7:0]	All 0	R/W	Indicates setting of the seconds value in the header for the first sector to be buffered.

24.3.44 Automatic Buffering Start Sector Setting: Frames Control Register (CBUFCTL3)

The automatic buffering start sector setting: frames control register (CBUFCTL3) indicates the frames (1 frame = 1/75 second) value in the header for the first sector to be buffered.

Bit:	7	6	5	4	3	2	1	0
	BS_FRM[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	BS_FRM[7:0]	All 0	R/W	Indicates setting of the frames (1 frame = 1/75 second) value in the header for the first sector to be buffered.

chanhvinh.com

24.3.45 ISY Interrupt Source Mask Control Register (CROMST0M)

The ISY interrupt source mask control register (CROMST0M) masks the ISY interrupt sources specified by the bits in the sync code status register (CROMST0).

Bit:	7	6	5	4	3	2	1	0
	-	-	ST_SYILM	ST_SYNOM	ST_BLKSM	ST_BLKLM	ST_SECSM	ST_SECLM
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
5	ST_SYILM	0	R/W	ISY interrupt ST_SYIL (bit 5 in the CROMST0 register) source mask
4	ST_SYNOM	0	R/W	ISY interrupt ST_SYNO (bit 4 in the CROMST0 register) source mask
3	ST_BLKSM	0	R/W	ISY interrupt ST_BLKS (bit 3 in the CROMST0 register) source mask
2	ST_BLKLM	0	R/W	ISY interrupt ST_BLKL (bit 2 in the CROMST0 register) source mask
1	ST_SECSM	0	R/W	ISY interrupt ST_SECS (bit 1 in the CROMST0 register) source mask
0	ST_SECLM	0	R/W	ISY interrupt ST_SECL (bit 0 in the CROMST0 register) source mask

24.3.46 CD-ROM Decoder Reset Control Register (ROMDECRST)

The CD-ROM decoder reset control register (ROMDECRST) resets the random logic of the CD-ROM decoder and clears the RAM in the CD-ROM decoder.

Bit:	7	6	5	4	3	2	1	0
	LOGICRST	RAMRST	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	LOGICRST	0	R/W	CD-ROM Decoder Random Logic Reset Signal A reset signal is output while this bit is set to 1.
6	RAMRST	0	R/W	CD-ROM Decoder RAM Clearing Signal Refer to the RAMCLRST bit in the RSTSTAT register to confirm that RAM clearing is complete.
5 to 0	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.

Note: • Before setting LOGICRST to 1, make sure that the RAMRST bit is cleared to 0 and then write B'10000000 to this register.

chanhvinh.com

24.3.47 CD-ROM Decoder Reset Status Register (RSTSTAT)

The CD-ROM decoder reset status register (RSTSTAT) indicates that the RAM in the CD-ROM decoder has been cleared.

Bit:	7	6	5	4	3	2	1	0
	RAM CLRST	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	RAMCLRST	0	R	This bit is set to 1 on completion of RAM clearing after the RAMRST bit in ROMDECRST is set to 1. The bit is cleared by writing a 0 to the RAMRST bit.
6 to 0	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.

chanhvinh.com

24.3.48 Serial Sound Interface Data Control Register (SSI)

The serial sound interface data control register (SSI) provides various settings related to the data stream. For the operation corresponding to the setting of this register, refer to section 24.4.1, Endian Conversion for Data in the Input Stream.

Bit:	7	6	5	4	3	2	1	0
	BYTEND	BITEND	BUFEND0[1:0]		BUFEND1[1:0]		-	-
Initial value:	0	0	0	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	BYTEND	0	R/W	Specifies the endian of input data from the serial sound interface. When this bit is set to 1, byte 0 and byte 1 in STRMDIN0 are swapped. This is the same for STRMDIN2.
6	BITEND	0	R/W	Specifies treatment of the bit order of the input data from the serial sound interface. When this bit is set to 1, the bits within each byte are rearranged to place them in reverse order, bit 0 → bit 7 to bit 7 → bit 0.
5, 4	BUFEND0[1:0]	01	R/W	These bits select whether to change the order of 16-bit units of data transferred from the serial sound interface or suppress the stream data. In the serial sound interface, either “padding mode” or “non-padding mode” is selectable. In non-padding mode, each 32 bits of data transferred from the serial sound interface are transferred as CD-ROM data. Since the CD-ROM decoder has two 16-bit input data registers, the order of the 16-bit data can be swapped within the 32 bits. On the other hand, in padding mode each 32 bits of data transferred from the serial sound interface includes padding. Since the padding has no meaning, it should be kept out of the input stream to the decoder. This suppression can be specified by the setting of this register. The CD-ROM decoder handles data as a stream of 16-bit data, and this register controls which 16-bit portion of each 32 bits of data transferred from the serial sound interface should be input first. 00: The 16 bits of stream data that would otherwise be processed first is discarded. 01: The higher-order 16 bits of each 32 bits of data received from the serial sound interface are placed first in the stream to the decoder. 10: The lower-order 16 bits of each 32 bits of data received from the serial sound interface are placed first in the stream to the decoder. 11: Setting prohibited
3, 2	BUFEND1[1:0]	10	R/W	These bits select whether to change the order of 16-bit units of data transferred from the serial sound interface or suppress the stream data. In the serial sound interface, either “padding mode” or “non-padding mode” is selectable. In non-padding mode, each 32 bits of data transferred from the serial sound interface are transferred as CD-ROM data. Since the CD-ROM decoder has two 16-bit input data registers, the order of the 16-bit data can be swapped within the 32 bits. On the other hand, in padding mode each 32 bits of data transferred from the serial sound interface includes padding. Since the padding has no meaning, it should be kept out of the input stream to the decoder. This suppression can be specified by the setting of this register. The CD-ROM decoder handles data as a stream of 16-bit data, and this register controls which 16-bit portion of each 32 bits of data transferred from the serial sound interface should be input second. 00: The 16 bits of stream data that would otherwise be processed second is discarded. 01: The higher-order 16 bits of each 32 bits of data received from the serial sound interface are placed second in the stream to the decoder. 10: The higher-order 16 bits of each 32 bits of data received from the serial sound interface are placed second in the stream to the decoder. 11: Setting prohibited
1, 0	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

24.3.49 Interrupt Flag Register (INTHOLD)

The interrupt flag register (INTHOLD) consists of various interrupt flags.

Bit:	7	6	5	4	3	2	1	0
	ISEC	ITARG	ISY	IERR	IBUF	IREADY	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ISEC	0	R/W	ISEC Interrupt Flag Writing 0 to this bit is only possible after 1 has been read from it.
6	ITARG	0	R/W	ITARG Interrupt Flag Writing 0 to this bit is only possible after 1 has been read from it.
5	ISY	0	R/W	ISY Interrupt Flag Writing 0 to this bit is only possible after 1 has been read from it.
4	IERR	0	R/W	IERR Interrupt Flag Writing 0 to this bit is only possible after 1 has been read from it.
3	IBUF	0	R/W	IBUF Interrupt Flag Writing 0 to this bit is only possible after 1 has been read from it.
2	IREADY	0	R/W	IREADY Interrupt Flag Writing 0 to this bit is only possible after 1 has been read from it.
1, 0	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

24.3.50 Interrupt Source Mask Control Register (INHINT)

The interrupt source mask control register (INHINT) controls masking of various interrupt requests in the CD-ROM decoder.

Bit:	7	6	5	4	3	2	1	0
	INH ISEC	INH ITARG	INH ISY	INH IERR	INH IBUF	INH IREADY	PREINH REQDM	PREINH IREADY
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	INHISEC	0	R/W	ISEC Interrupt Mask When set to 1, inhibits ISEC interrupt requests.
6	INHITARG	0	R/W	ITARG Interrupt Mask When set to 1, inhibits ITARG interrupt requests.
5	INHISY	0	R/W	ISY Interrupt Mask When set to 1, inhibits ISY interrupt requests.
4	INHIERR	0	R/W	IERR Interrupt Mask When set to 1, inhibits IERR interrupt requests.
3	INHIBUF	0	R/W	IBUF Interrupt Mask When set to 1, inhibits IBUF interrupt requests.
2	INHIREADY	0	R/W	IREADY Interrupt Mask When set to 1, inhibits IREADY interrupt requests.
1	PREINHREQDM	0	R/W	Inhibits setting of the DMA-transfer-request interrupt source flag for the output data stream. When this bit is set to 1, the DMA-transfer-request interrupt source is not retained.
0	PREINHIREADY	0	R/W	Inhibits setting of the IREADY interrupt flag. When this bit is set to 1, the IREADY interrupt source is not retained.

24.3.51 CD-ROM Decoder Stream Data Input Register (STRMDIN0)

The CD-ROM decoder stream data input register (STRMDIN0) holds the higher 2 bytes (from MSB) of the 4 bytes of data that is to be input to the CD-ROM decoder.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STRMDIN[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	STRMDIN[31:16]	All 0	R/W	Indicates the higher 2 bytes (from MSB) of the 4-bytes of data that is to be input to the CD-ROM decoder. The CD-ROM decoder has a 4-byte wide data window as a data input register to handle the data input to this register as a stream data. The amount of data for one sector is 2352 bytes.

chanhvinh.com

24.3.52 CD-ROM Decoder Stream Data Input Register (STRMDIN2)

The CD-ROM decoder stream data input register (STRMDIN2) holds the lower 2 bytes (from LSB) of the 4 bytes of data that is to be input to the CD-ROM decoder.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STRMDIN[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	STRMDIN[15:0]	All 0	R/W	Indicates the lower 2 bytes (from LSB) of the 4-bytes of data that is to be input to the CD-ROM decoder. The CD-ROM decoder has a 4-byte wide data window as a data input register to handle the data input to this register as a stream data. The amount of data for one sector is 2352 bytes.

24.3.53 CD-ROM Decoder Stream Data Output Register (STRMDOUT0)

The CD-ROM decoder stream data output register (STRMDOUT0) holds 2 bytes that is to be output from the CD-ROM decoder.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STRMDOUT[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	STRMDOUT [15:0]	H'0000	R	Indicates 2 bytes that are to be output from the CD-ROM decoder. The CD-ROM decoder has a 2-byte wide data window or set of registers for the output of decoded data. Every time the relevant register is accessed, further data of access size are output sequentially in the output format that is separately defined. The amount of data for one sector is 2768 bytes. Always read 2768 bytes.

chanhvinh.com

24.4 Operation

24.4.1 Endian Conversion for Data in the Input Stream

Stream data must be input to the core of the CD-ROM decoder in order according to the CD-ROM data format specifications. In some systems, however, the order of the data from the serial sound interface may have to be changed or the data will have been padded before transfer. To cope with this, the stream data input control section is capable of swapping the order of the data and preventing the input of padding data to the core of the CD-ROM decoder. These functions are controlled through the serial sound interface data control register (SSI).

Figure 24.6 shows a case where the upper and lower 16 bits of the data, consisting of padding data plus the first 2 bytes of sync code, that is, H'000000FF, are swapped (H'00FF0000) and input to the CD-ROM decoder as the stream data.

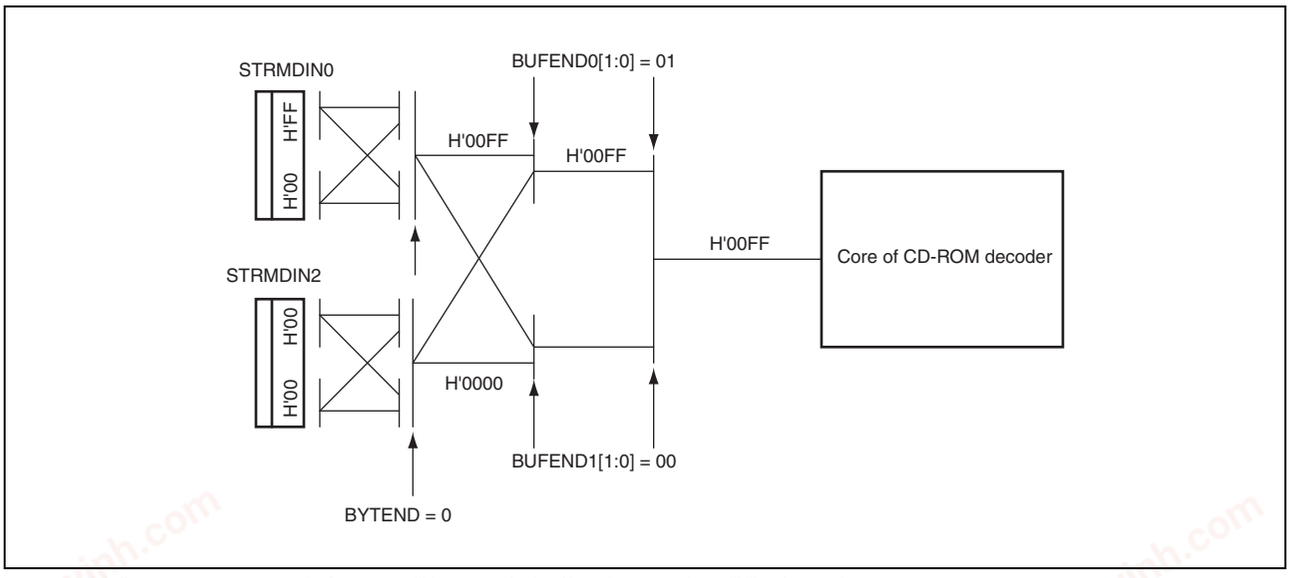


Figure 24.6 Example of Padded Stream Data Control by the SSI Register

Figure 24.7 shows a case of input stream data that has no padding (H'12345678). The upper and lower 16 bits of data are swapped (H'56781234) for input to the CD-ROM decoder.

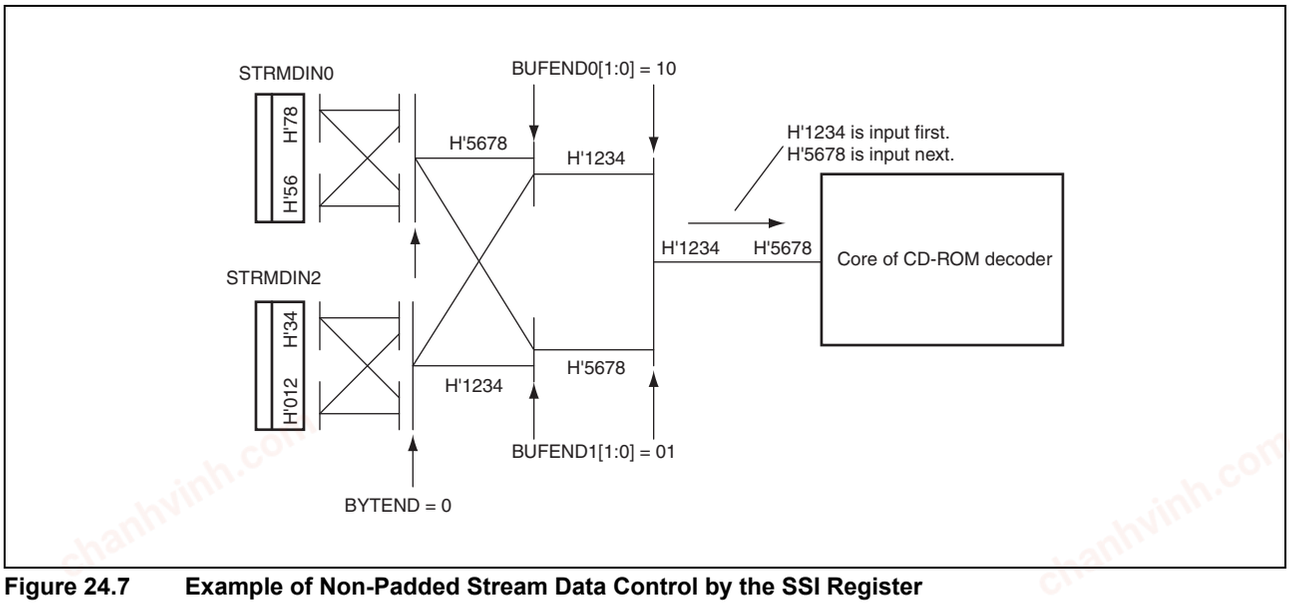


Figure 24.7 Example of Non-Padded Stream Data Control by the SSI Register

chanhvinh.com

24.4.2 Sync Code Maintenance Function

Each sector of CD-ROM data consists of 2352 bytes starting with H'00FFFFFFFFFFFFFFFFFF00 (sync code). However, a scratch on the disc or some other factor might lead to erroneous recognition of the sync code sequence at the wrong time. Conversely, a sync code might not be detected at a point where it should be detected. As a solution to these problems, the CD-ROM decoder of this LSI has a sync-code maintenance function, which operates to ignore sync codes detected at abnormal times and maintain the appearance of the sync code at the expected times when it is not actually detected on the disc.

The operating modes of the sync-code maintenance function are listed below. For details on the settings, refer to section 24.3.2, Sync Code-Based Synchronization Control Register (CROMSY0), and Table 24.2.

- Automatic sync maintenance mode
- External sync mode
- Interpolated sync mode
- Interpolated sync plus external sync mode

(1) Automatic Sync Maintenance Mode

In automatic sync maintenance mode, the sync code is ignored if detected within the one-sector (2352-byte) period. Furthermore, if a sync code is not detected at the point where a next sector should start, sync code maintenance is applied. If synchronization timing has changed, re-synchronization is performed at the point where a sync code is detected within 2352 bytes after the change.

Therefore, this mode is effective in rejecting abnormal sync patterns and following changes in synchronization timing. Note, however, that this mode cannot achieve synchronization with the first sector after a change to the synchronization timing.

Figure 24.8 shows operation in the case of normal sync-code detection, Figure 24.9 shows a case where a sync code is detected before a current one-sector period has elapsed, and Figure 24.10 shows the case where the actual sync code is only detected some time after a full one-sector period has elapsed.

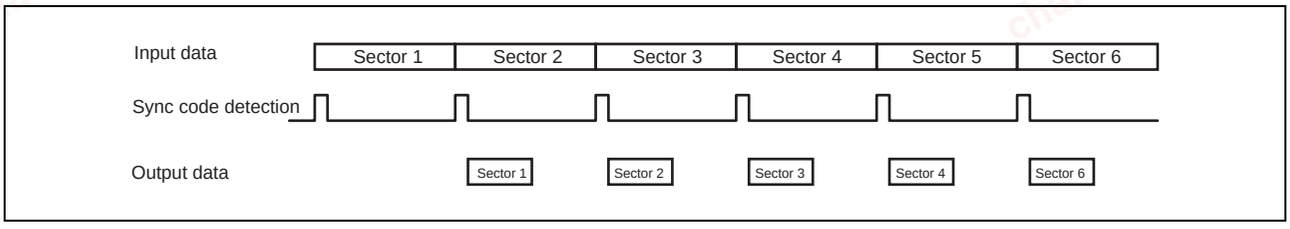


Figure 24.8 Operation in Automatic Sync Maintenance Mode (Normal Timing)

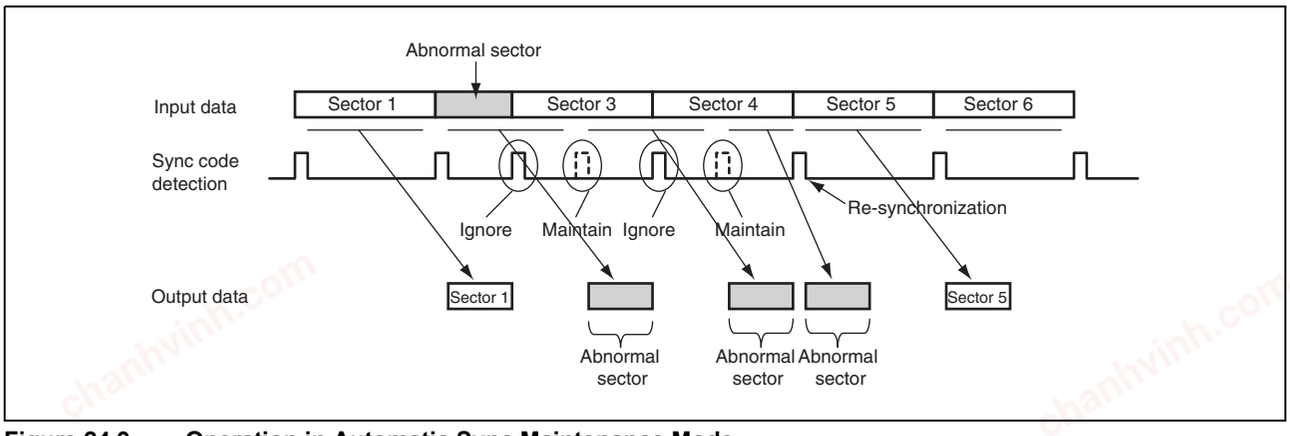


Figure 24.9 Operation in Automatic Sync Maintenance Mode (When an Abnormally Short Sector is Encountered)

chanhvinh.com

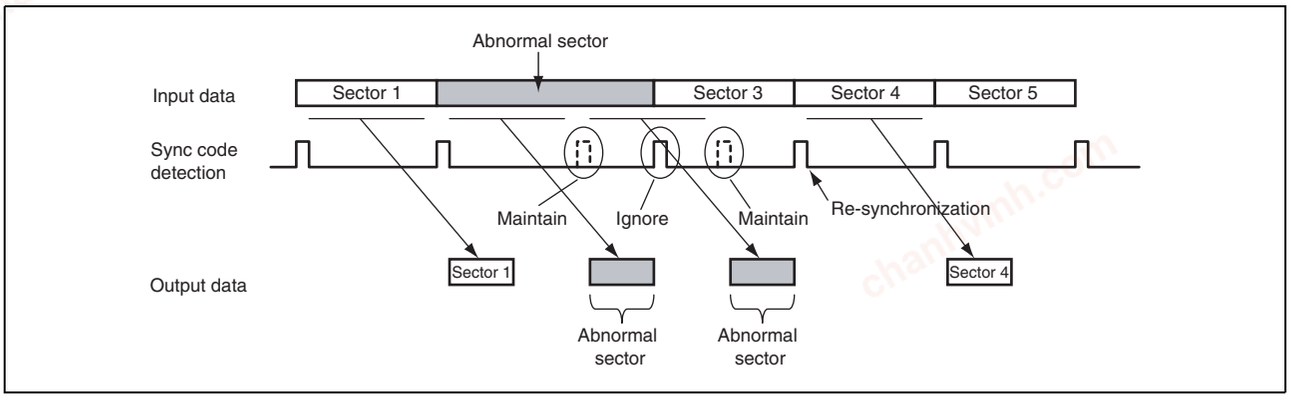


Figure 24.10 Operation in Automatic Sync Maintenance Mode (When an Abnormally Long Sector is Encountered)

(2) External Sync Mode

In external sync mode, synchronization is always based on the sync codes in the incoming data. Even if the next sync code is not detected at the 2352nd byte, decoding does not proceed until the next sync code is detected.
Accordingly, this mode is effective in that it strictly follows the external synchronization timing. Note, however, that decoding will not be performed normally when the sync-code pattern is input with abnormal timing.
Figure 24.11 shows the operation in external sync mode.

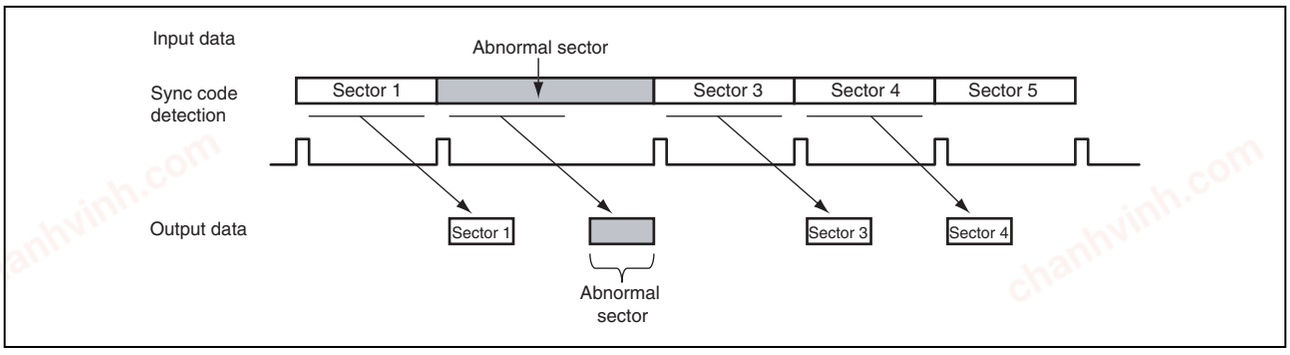


Figure 24.11 Operation in External Sync Mode

(3) Interpolated Sync Mode

In interpolated sync mode, synchronization is always driven by the internal counter after a sync code pattern has been detected at the start of decoding. Accordingly, this mode is effective when the sync patterns have been damaged. However, decoding becomes incorrect after a change to the synchronization timing, since the change in timing is not followed.
Figure 24.12 shows the operation in interpolated sync mode.

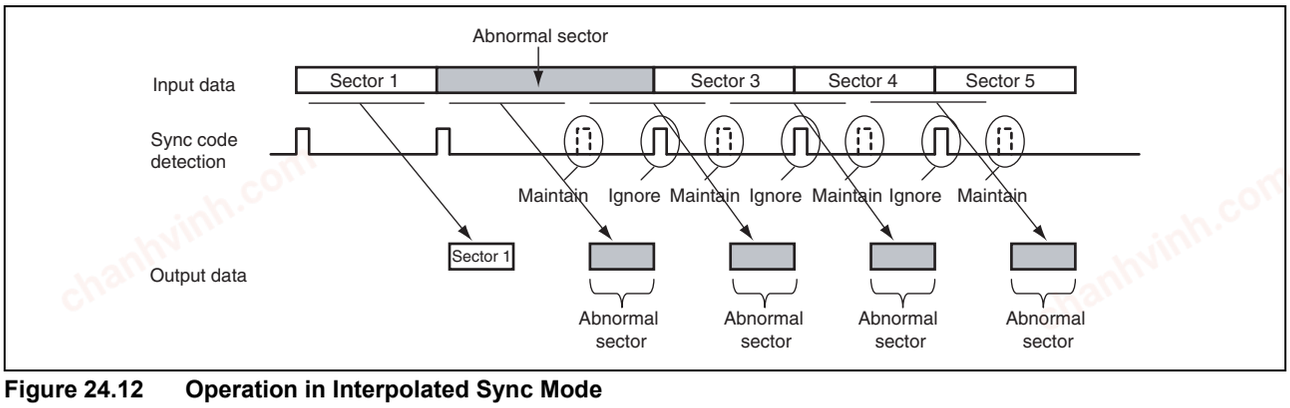


Figure 24.12 Operation in Interpolated Sync Mode

chanhvinh.com

(4) Interpolated Sync Plus External Sync Mode

In interpolated sync plus external sync mode, synchronization is based on the detected sync code patterns as long as they are present, and if a sync pattern is not detected at the 2352nd byte, the sync code maintenance is applied. Synchronization in this mode is more quickly responsive to changes in synchronization timing than synchronization in the automatic sync maintenance mode.
However, decoding still becomes incorrect when a sync pattern is input with abnormal timing.
Figure 24.13 and Figure 24.14 show the operation in interpolated sync plus external sync mode in the cases of abnormally short and long sectors, respectively.

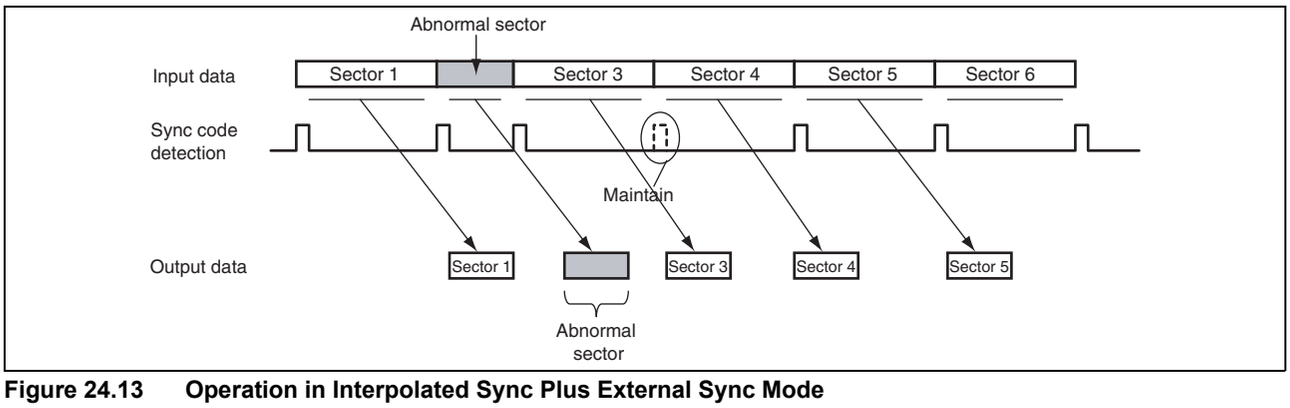


Figure 24.13 Operation in Interpolated Sync Plus External Sync Mode (When an Abnormally Short Sector is Encountered)

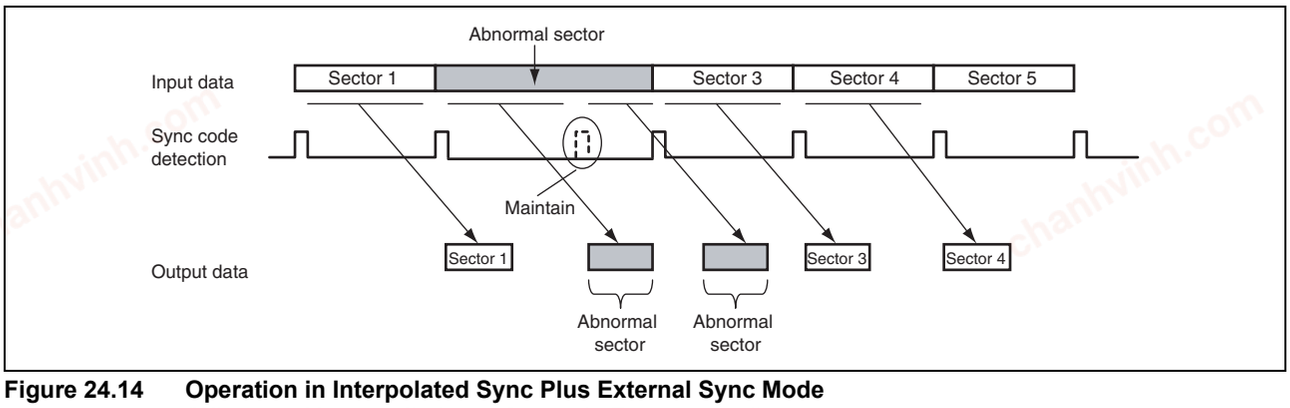


Figure 24.14 Operation in Interpolated Sync Plus External Sync Mode (When an Abnormally Long Sector is Encountered)

24.4.3 Error Correction

The CD-ROM decoder handles data in the formats containing information relevant to error correction, including the EDC, P parity, and Q parity. The CD-ROM decoder includes the following functions for use in error correction.

- Syndrome calculation
- ECC correction
- EDC checking

(1) Syndrome Calculation

After the data of a sector in Mode 1 or Form 1 of Mode 2 has been input, the ECC is used in correction if any error is detected (the result of syndrome calculation is non-zero). After correction, the results of syndrome operation for the corrected data are output to bits ST_ECCP (P parity) and ST_ECCQ (Q parity) in the CROMST6 register, respectively.

chanhvinh.com

(2) ECC correction and EDC Checking

For CD-ROM format data that contains EDC, P-parity, and Q-parity fields, the CD-ROM decoder performs EDC checking and ECC correction. Supported correction modes are P correction, Q correction, PQ correction (P correction followed by Q correction), and QP correction (Q correction followed by P correction). In PQ and QP correction modes, up to three iterations of correction are possible (the number of iterations is limited by the playback speed).
The EDC check is performed twice, before and after correction.
The mode of ECC correction and EDC checking is specified by bits MD_DEC[2:0] in the CROMCTL1 register. When the PQ or QP correction mode is selected, the number of iterations of correction is specified by bits MD_PQREP[1:0] in the CROMCTL1 register.
When the automatic mode/form detection function is in use, the sector mode determines whether or not ECC correction and EDC checking can be performed. For sectors in Mode 0 and Mode 2 (non-XA), which include neither parity bits nor EDC, ECC correction and EDC checking are not performed. For sectors in Form 2 of Mode 2, ECC correction is not performed.

(a) ECC Correction

When ECC correction is in use and an error in a sector is identified as non-correctable, the CD-ROM decoder generates an IERR interrupt and sets the ST_ECCNG bit of the CROMST6 register to 1. The CD-ROM detector also sets this bit to 1 on detecting a short sector.
While the NO_ECC bit of the CROMCTL4 register is set to 1, a ‘pass’ result in pre-ECC correction EDC checking makes the CD-ROM decoder skip ECC correction, regardless of the results of the syndrome operation.

(b) EDC Checking

When EDC checking is in use, checking is in line with the specified or detected sector mode and form, depending on whether or not automatic sector mode and form detection is selected.
The results of EDC checking before and after correction are reflected in the ST_EDC1 and ST_EDC2 bits of the CROMST6 register, respectively. If EDC checking after ECC correction indicates that an error remains, an IERR interrupt is generated.

24.4.4 Automatic Decoding Stop Function

Decoding can be stopped automatically in response to an error during the decoding of CD-ROM data.
The possible conditions for automatically stopping the decoding process are listed below. The applicable conditions are specified in the CROMCTL3 register.

- An error is found to be not correctable by ECC correction.
- Post-ECC correction EDC checking indicates that an error remains.
- A change of the sector mode or form
- A non-sequential MSF (minutes, seconds, frames (1/75 second)) value

When automatic stopping is set up and any of the above conditions is encountered in a certain sector, the decoding is stopped after the results of decoding for that sector have been output.
After decoding has been stopped in response to a condition specified in the CROMCTL3 register, the condition can be identified by reading the CBUFST1 register.
The CD-ROM decoder has buffer space for two sectors. If input of the data stream continues and the output stream of data is not read, the CD-ROM decoder stops at the point where the data of a third sector starts to be input. At this time, the BUF_NG bit in the CBUFST2 register is set to 1, but no interrupt is generated. Once the BUF_NG bit in the CBUFST2 register has been set to 1, recovery can only be accomplished by using the LOGICRST bit in the ROMDECRST register to reset the CD-ROM decoder function. When the LOGICRST bit in the ROMDECRST register is set to 1, a reset signal is output and any registers in which settings have been made are cleared to their initial values.

chanhvinh.com

24.4.5 Buffering Format

Figure 24.15 shows the format of the output data stream produced by CD-ROM decoding.
A 2-byte-wide window register STRMDOUT0 is provided for the output. When this window register is accessed after decoding of a CD-ROM sector has finished, the bytes of data are output in order from the sync code.

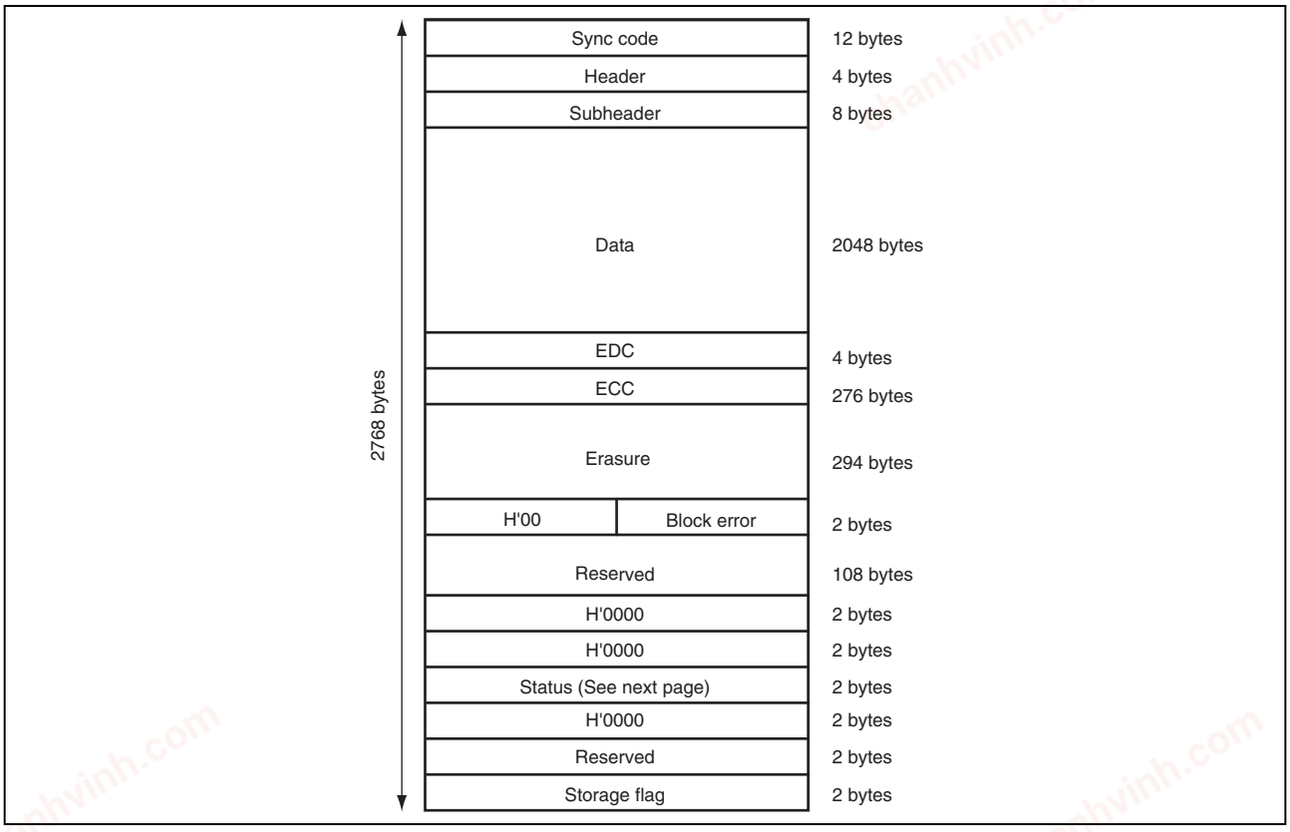


Figure 24.15 Output Data Stream Format

The meanings of bits in the two-byte status field shown in Figure 24.15 are given below. The values of the non-assigned bits are undefined.

Status

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PERR	QERR	EDCE	—	—	—	—	—	SD	SY		FM[2:0]		HD	—	—

[Legend]
PERR: Indicates that a P-parity error remains.
QERR: Indicates that a Q-parity error remains.
EDCE: Indicates that a remaining error was detected in post-ECC correction EDC checking.
SD: Indicates that a short sector was encountered.
SY: Indicates that a sync code was interpolated.
FM: Indicates the data format.
001: Mode 0
010: Mode 1
011: Long (format with no EDC and ECC)
100: Mode 2 (non-XA)
101: Mode 2 Form 1
110: Mode 2 Form 2
HD: Header continuity (minutes, seconds, and frames (1 frame = 1/75 second) are non-sequential)

The value of the storage flag field in Figure 24.15 is incremented every time the data for one sector are output. The value starts at H'0000 and wraps back around to H'0000 after it reaches H'FFFF. Note that the upper byte and lower byte in the storage flag are swapped.

chanhvinh.com

24.4.6 Target-Sector Buffering Function

In the CD-ROM decoder, the sector for output can be designated in two ways: automatic buffering, where the CD-ROM decoder itself detects the presence of target sector that has been set beforehand, and manual buffering, where the target sector for output is designated by software and the software also recognizes the sectors buffered in the CD-ROM decoder.

The following describes the procedures for setting the registers in the CD-ROM decoder to set up automatic or manual buffering.

(1) Setting Up Automatic Buffering

Figure 24.16 shows an example of setting up the automatic buffering. Set the relevant CD-ROM decoder registers and start input of the data stream; the CD-ROM decoder then detects the target sector and starts the output of the stream data.

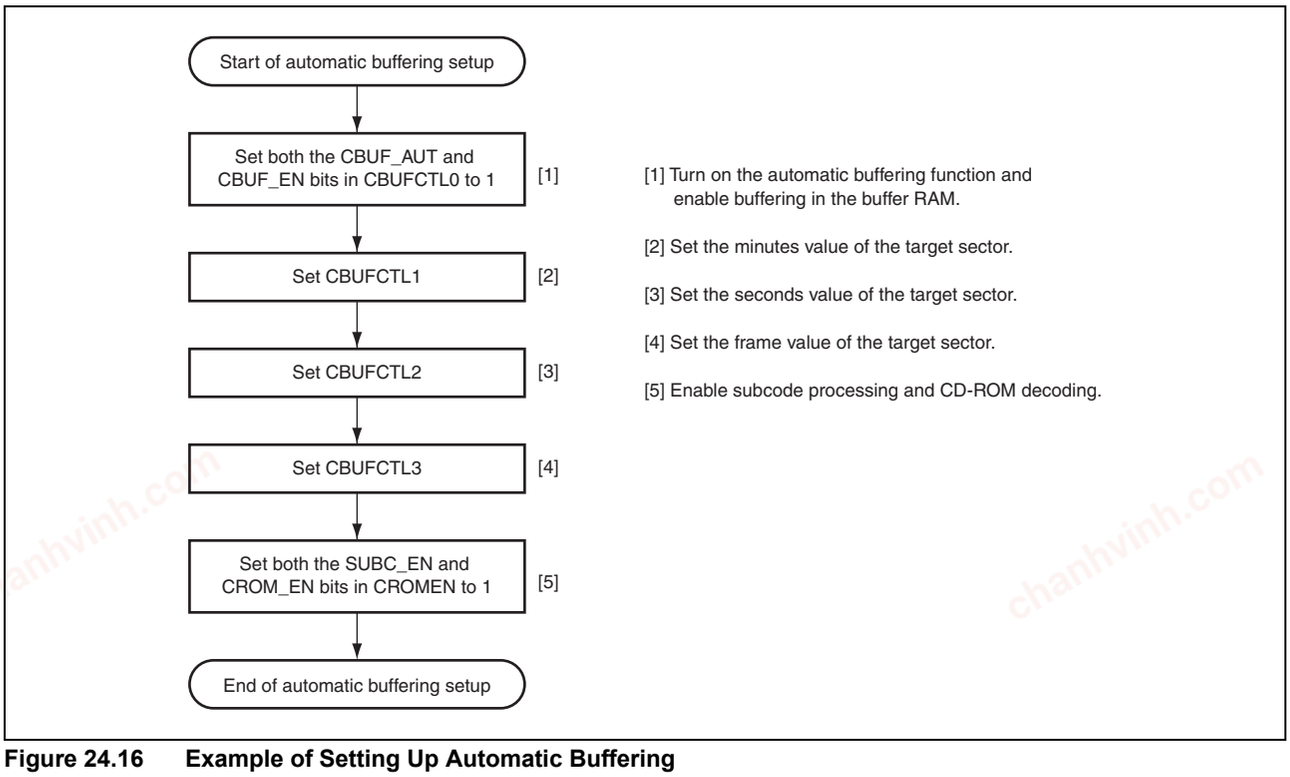


Figure 24.16 Example of Setting Up Automatic Buffering

chanhvinh.com

(2) Setting Up Manual Buffering

Figure 24.17 shows an example of setting up manual buffering. Each time an ISEC interrupt is generated, the software checks whether or not the sector is the target sector and starts buffering when the target sector is found.

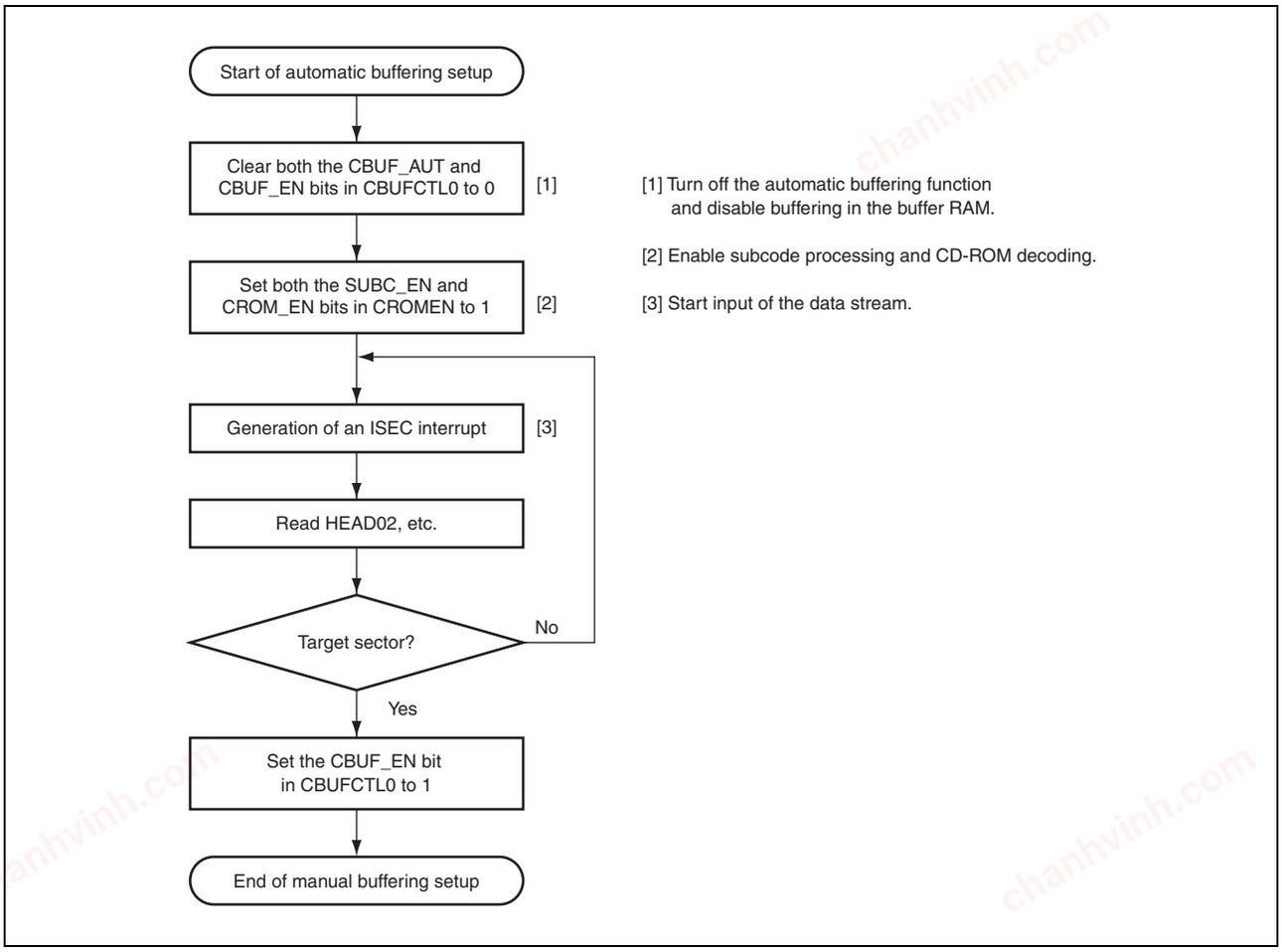


Figure 24.17 Example of Setting Up Manual Buffering

chanhvinh.com

24.5 Interrupt Sources

24.5.1 Interrupt and DMA Transfer Request Signals

Table 24.3 lists the interrupt signals and DMA transfer request signal generated by the CD-ROM decoder, along with the meanings and the modules to which the signals are connected.

Table 24.3 Interrupt and DMA Transfer Request Signals

Name	Condition	Connected To
ISEC	Transition from sector to sector	Interrupt controller
ITARG	Access to a CD-ROM sector that is not the expected target sector	Interrupt controller
ISY	A sync code from the CD-ROM with abnormal timing	Interrupt controller
IERR	An error that was not correctable by ECC correction or an error indicated by EDC checking after ECC correction	Interrupt controller
IBUF	State changes in data transfer to the buffer	Interrupt controller
IREADY	Request for data transfer to the buffer for CD-ROM	Interrupt controller
DMA transfer request	Request for data transfer to the buffer for CD-ROM	Direct memory access controller

(1) ISEC Interrupt

This interrupt is generated when the sync code indicates a transition from sector to sector.

(2) ITARG Interrupt

This interrupt is generated when the stream data transferred from the CD-DSP is not the data of the target sector. The CD-ROM decoder checks the time data in the subcode. In correct operation, data transfer is expected to start slightly before the target sector. An ITARG interrupt is generated in the following cases.

- When data of a sector preceding the target sector by quite a few sectors have been transferred
- When data of a sector that comes after the target sector have been transferred

For the generation of this interrupt, ITARG is detected from the subcode. However, this interrupt has no meaning in this LSI because CD-ROM data are transferred from the serial sound interface.

(3) ISY Interrupt

This interrupt can be generated in the following cases.

- When a sync code was detected at a position where the value in the word counter (counter for checking sync code intervals) was not correct and the sync code was ignored
- When a sync code has not been detected although the word counter has reached the final value and a sync code has been interpolated (for sync maintenance)
- When a sync code was detected at a position where the value in the word counter (counter for checking sync code intervals) was not correct and the sync code was used in resynchronization
- When a sync code has not been detected although the word counter has reached the final value, so the period taken up by the sector has been prolonged
- When the sector has been processed as a short sector with the aid of interpolated sync codes
- When the sector has been processed as a long sector with the aid of interpolated sync codes

chanhvinh.com

(4) IERR Interrupt

This interrupt is generated in the following cases.

- When ECC correction was incapable of correcting an error
- When ECC correction was OK but the subsequent EDC check indicated an error

(5) IBUF Interrupt

This interrupt is generated when the following transitions occur.

- Data transfer to the buffer → Data transfer complete (searching for data for the next transfer)
- Data for transfer to the buffer are being searched for → Data transfer started

(6) IREADY Interrupt

This interrupt is generated when decoding of data for one sector is completed. This interrupt should be used to start the CPU buffering stream data for output to SDRAM.

(7) DMA Transfer Request

The source of direct memory access controller activation is the same as that of IREADY. An interrupt request is generated when output stream data for one sector becomes ready, and after the 2768 bytes of data shown in figure 24.15 have been transferred, the request signal is negated. This is because a certain amount of time is required before the output data for the next sector is ready, so the transfer request from the direct memory access controller should be turned off between transfers.

24.5.2 Timing of Status Registers Updates

The status information registers of the CD-ROM decoder are updated on each ISEC interrupt. The sector for which information is reflected in the status registers is selected by the ER0SEL bit of the CROMCTL4 register.

chanhvinh.com

24.6 Usage Notes

24.6.1 Stopping and Resuming Buffering Alone during Decoding

When the data of the output stream are not being read out but operation of the CD-ROM decoder has continued until the buffers are full, the BUF_NG bit in the CBUFST2 register is set to 1; after that, the CD-ROM decoder becomes incapable of operation.
To stop buffering alone, clear the CBUF_EN bit in the CBUFCTL0 register to 0. If the automatic buffering function is in use, clear the CBUF_AUT bit in the CBUFCTL0 register to 0 at the same time. In this case, the sectors currently in the buffers must be read out.
To resume automatic buffering, set the CBUF_AUT and CBUF_EN bits in the CBUFCTL0 register to 1 at the same time.

24.6.2 Setting Sync Code Status Register (CROMST0)

1. When the ST_SECS bit in the CROMST0 register becomes 1, stop decoding immediately and retry from one sector before the sector that was being decoded.
2. When the ST_SECL bit in the CROMST0 register becomes 1, stop decoding immediately and retry from two sectors before the sector that was being decoded.

24.6.3 Link Blocks

The CD-ROM decoder uses the header information before ECC correction to detect link blocks. Accordingly, an input data stream that contains an error may be erroneously detected as a link block. To prevent this, the following measures should be implemented in software.

- During buffering (BUF_ACT = 1 in the CBUFST0 register), check the LINK_OUT1 bit in the CROMST5 register on each ISEC interrupt. If it is set to 1, check to see if an IERR interrupt has also occurred; if an IERR interrupt has not occurred, save the MFS values from the HEAD20 to HEAD23 registers. If an IERR interrupt has occurred, do not save the MSF values.
- Perform the following processing for seven sectors (indicated by ISEC being generated seven times) after finding that the LINK_OUT1 bit has been set to 1.
- In either of cases 1 and 2 below,
  1. LINK_ON = 1 (in the CROMST5 register) is confirmed at each ISEC interrupt, and LINK_ON = 1 is detected again within the subsequent two-sector period
  2. LINK_ON = 1 was not detected at any ISEC interrupt

forcibly stop decoding, set the CROMSY0 register to place the decoder in external sync mode, and retry decoding by specifying the MSF value stored as described above + 7 (immediately after a link block) as the MSF value for the target sector.
The start sector address will be the address where RUN_OUT is stored + 7.

24.6.4 Stopping and Resuming CD-DSP Operation

When stopping and then resuming the stream data input to the CD-ROM decoder, if the input data stream does not stop immediately before a sync code and is then resumed, the CD-ROM decoder may recognize the data as incorrect. This happens because the system holds the data up to the point where input was stopped and the data that is input from the point of resumption at the same time. Take care on this point when stopping and then resuming input.

chanhvinh.com

24.6.5 Note on Clearing the IREADY Flag

To clear the IREADY flag to 0 in interrupt processing etc., be sure to read one sector of data (2768 bytes) beforehand. If the IREADY flag is cleared to 0 before reading of one sector of data is complete, decoding of the subsequent sectors will not be possible. For recovery from this situation, write 1 to the LOGICRST bit in the CD-ROM decoder reset control register (ROMDECRST), and then clear the bit to 0.

24.6.6 Note on Stream Data Transfer (1)

If reading of the output data stream is slower than writing of the input data stream, the buffer of the CD-ROM decoder will overflow. This causes the CD-ROM decoder to abnormally stop.

When DMA transfer is in use, ensure that reading of the output data stream is faster than writing of the input data stream by making settings as listed below.

- Set a larger transfer size for reading of the output data stream than for writing of the input data stream.
- Give reading of the output data stream higher priority than writing of the input data stream.
- Set a smaller interval count for reading of the output data stream than for writing of the input data stream.

When the CPU handles transfer, ensure that reading of the output data stream is faster than writing of the input data stream by taking similar measures to those in the case of DMA transfer.

24.6.7 Note on Stream Data Transfer (2)

When reading the stream data, be sure to use either the direct memory access controller or the CPU. If both the direct memory access controller and the CPU are used for reading, the stream data may not be recognized as being in the CD-ROM format.

24.6.8 Note on Software Reset

For transitions to the software reset state by the LOGICRST bit in the ROMDECRST register, see section 42.3.6, Software Reset. However, where the procedure refers to the SRST bit, read this as the LOGICRST bit.

chanhvinh.com

25. LIN Interface

This module is only incorporated in the RZ/A1L.

This section contains a generic description of the LIN Interface (RLIN3). The first part of this section describes all specific properties of this product, such as the register base addresses, etc. The remainder of the section describes the functions and registers of RLIN3.

25.1 Features

25.1.1 Channels

This LSI has following number of channels of the LIN Interface.

Table 25.1 Channels of RLIN3n

LIN Interface	RZ/A1L
Channels	1
Name	RLIN30

Index n

Throughout this section, the channel of the LIN Interface (RLIN3) is identified by the index “n” (n = 0), for example, RLN3nLMD for the LIN mode register.

25.1.2 Register Addresses

The register base address of the LIN interface is listed in the following table.

All LIN interface register addresses are given as address offsets to the individual base address.

Table 25.2 Register Base Addresses

Base Address Name	Base Address
<RLIN30_base>	FCFE 9000$_H$

25.1.3 Clock Supply

The following clock is provided for the LIN interface.

Table 25.3 RLIN3n Clock Supply

RLIN3n	Clock	Connected to
RLIN3n	LIN Communication Clock Source	P0φ

chanhvinh.com

25.1.4 Interrupts and DMA

The LIN Interface can generate the following interrupt and DMA requests:

Table 25.4 RLIN3n Interrupt and DMA Requests

RLIN3n signal	Function
RLIN30	
LIN0_INT_M	LIN0 interrupt
LIN0_INT_T	LIN0 transmission interrupt
LIN0_INT_R	LIN0 successful reception interrupt
LIN0_INT_S	LIN0 status error interrupt

25.1.5 I/O Signals

The I/O signals of the LIN Interface are used for various purposes, as listed in Table 25.5.

Table 25.5 RLIN3n I/O Signals

Name	Function
RLIN30	
Port RLIN30RX	RLIN30 receive data input
Port RLIN30TX	RLIN30 transmit data output

chanhvinh.com

25.2 Function

The LIN Interface is a hardware LIN communication controller that supports LIN Specification Package Revision 1.3, 2.0, 2.1, 2.2, and SAEJ2602, and automatically performs frame communication and error determination.

The LIN master mode is only available.

LIN master

- LIN reset mode
- LIN mode (LIN master mode)
  - LIN wake-up mode
  - LIN operation mode
- LIN self-test mode

chanhvinh.com

Table 25.6 gives the LIN Interface specifications and Figure 25.1 shows a block diagram of the LIN Interface.

Table 25.6 LIN Interface Specifications

Item		Specifications	
	Channel count	1	
LIN communication function	Protocol	LIN Specification Package Revision 1.3, 2.0, 2.1, 2.2, and SAEJ2602	
	Variable frame structure	Master	• Break transmission width: 13 to 28 Tbits • Break delimiter transmission width: 1 to 4 Tbits • Transmission inter-byte space width (header): 0 to 7 Tbits (space between Sync field and ID field)*1 • Transmission response space width: 0 to 7 Tbits*1 • Transmission inter-byte space width: 0 to 3 Tbits (space between data bytes in response area) • Transmit wake-up width: 1 to 16 Tbits
	Checksum	• Automatic operation for both transmission and reception • Classic or enhanced selectable (for each frame)	
	Response field data byte count	Variable from 0 to 8 bytes Multi-byte (9 or more bytes) response transmission and reception also possible	
	Frame communication modes	Master	• Mode in which header transmission and response transmission/reception is started with a single transmission start request • Mode in which header transmission and response transmission are started with separate transmission start requests (frame separate mode) (Setting of the frame separate mode is prohibited in this product.)
	Wake-up transmission and reception	LIN wake-up mode provided • Wake-up transmission (1 to 16 Tbits) • Wake-up reception Low-level width of input signals measured	
	Status	Master	• Successful frame/wake-up transmission • Successful header transmission • Successful frame/wake-up reception*2 • Successful data 1 reception • Error detection • Operation mode (LIN reset mode, LIN wake-up mode, LIN operation mode, LIN self-test mode)
	Error status	Master	• Bit error • Checksum error • Frame timeout error/response timeout error • Physical bus error • Framing error • Response preparation error
	Baud rate selection	Baud rate conforming to the LIN specifications generated using baud rate generator	
	Test mode	Self-test mode for user evaluation	
	Interrupt function	Master	• Successful header/frame/wake-up transmission • Successful frame/wake-up reception*2 • Error detection

Note 1. Since the same register is used for setting, the inter-byte space (header) = response space.
Note 2. For wake-up reception, the low level width of the input signal is indicated.

chanhvinh.com

25.2.1 Block Diagram

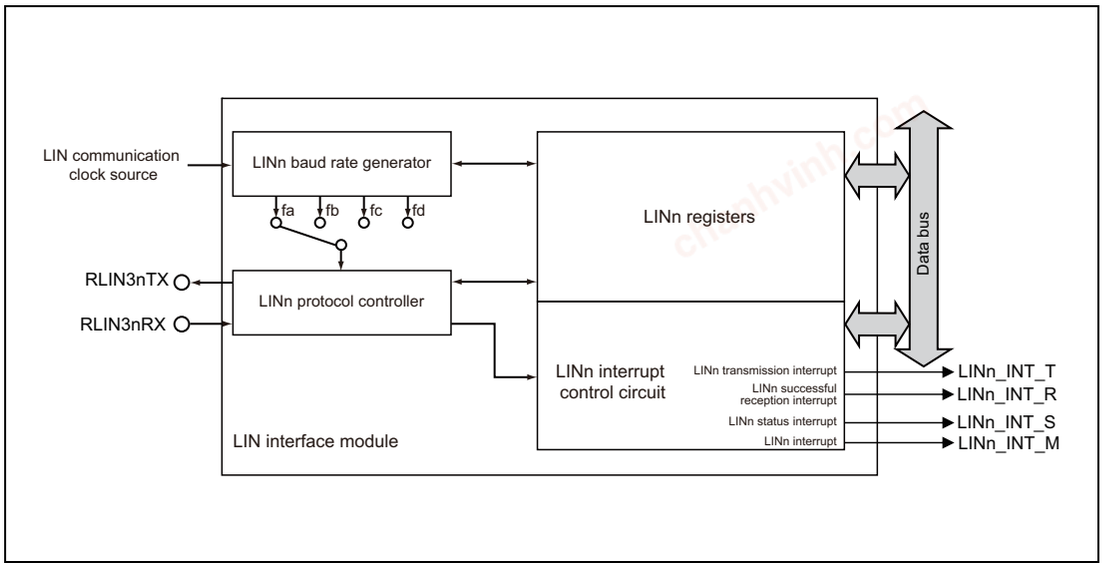


Figure 25.1 LIN Interface Block Diagram

25.2.2 Description of Blocks

- RLIN3nTX, RLIN3nRX: LIN Interface I/O pins
- LINn baud rate generator: Generates the LIN Interface communication clock signal.
- LINn registers: LIN Interface registers
- LINn interrupt controller: Controls interrupt requests generated by the LIN Interface

chanhvinh.com

25.3 Registers

Table 25.7 lists the LIN Interface registers.

Table 25.7 List of LIN Interface Registers

Register Name	Symbol
LIN wake-up baud rate selector register	RLN3nLWBR
LIN baud rate prescaler 0 register	RLN3nLBRP0
LIN baud rate prescaler 1 register	RLN3nLBRP1
LIN self-test control register	RLN3nLSTC
LIN mode register	RLN3nLMD
LIN break field configuration register	RLN3nLBFC
LIN space configuration register	RLN3nLSC
LIN wake-up configuration register	RLN3nLWUP
LIN interrupt enable register	RLN3nLIE
LIN error detection enable register	RLN3nLEDE
LIN control register	RLN3nLCUC
LIN transmission control register	RLN3nLTRC
LIN mode status register	RLN3nLMST
LIN status register	RLN3nLST
LIN error status register	RLN3nLEST
LIN data field configuration register	RLN3nLDFC
LIN ID buffer register	RLN3nLIDB
LIN checksum buffer register	RLN3nLCBR
LIN data buffer 1 register	RLN3nLDBR1
LIN data buffer 2 register	RLN3nLDBR2
LIN data buffer 3 register	RLN3nLDBR3
LIN data buffer 4 register	RLN3nLDBR4
LIN data buffer 5 register	RLN3nLDBR5
LIN data buffer 6 register	RLN3nLDBR6
LIN data buffer 7 register	RLN3nLDBR7
LIN data buffer 8 register	RLN3nLDBR8

Note: When writing to a register not used, write 00_H.

chanhvinh.com

25.3.1 LIN Master Related Registers

25.3.1.1 RLN3nLWBR — LIN Wake-up Baud Rate Select Register

Access: This register can be read/written in 8-bit units.

Address: <RLIN3n_base> + 01_H

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	NSPB[3:0]				LPRS[2:0]			LWBR0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.8 RLN3nLWBR register contents

Bit Position	Bit Name	Function
7 to 4	NSPB[3:0]	Bit Sampling Count Select b7 b4 0 0 0 0: 16 sampling 1 1 1 1: 16 sampling Settings other than the above are prohibited.
3 to 1	LPRS[2:0]	Prescaler Clock Select b3 b1 0 0 0: 1/1 0 0 1: 1/2 Other than above: Setting prohibited
0	LWBR0	Wake-up Baud Rate Select 0: In LIN wake-up mode, the clock specified by the LCKS bit setting in the RLN3nLMD register is used (when LIN1.3 is used). 1: In LIN wake-up mode, the clock fa is used regardless of the setting of the LCKS bit in the RLN3nLMD register (when LIN2.x is used).

Set the RLN3nLWBR register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

NSPB[3:0] bits (bit sampling count select bits)

These bits select the number of sampling in one Tbit (reciprocal of the baud rate).

In LIN master mode, set these bits to 0000_B or 1111_B (16 sampling).

LPRS[2:0] bits (prescaler clock select bits)

These bits select the frequency division ratio for the prescaler.

LWBR0 bit (wake-up baud rate select bit)

When LIN Specification Package Revision 1.3 is used, set the LWBR0 bit in the RLN3nLWBR register to 0. This allows the 2.5-Tbit or longer low-level width of the input signal to be measured. When LIN Specification Package Revision 2.x is used, set the LWBR0 bit to 1. Setting the LWBR0 bit to 1 selects fa as the LIN system clock (fLIN) during LIN wake-up mode regardless of the setting of the RLN3nLMD.LCKS bit (the LCKS bit is not changed). This allows the 2.5-Tbit or longer low-level width of the input signal to be measured.

Setting the baud rate to 19200 bps while fa is selected allows the 130 µs or longer low-level width of the input signal to be detected during LIN wake-up mode regardless of the setting of the RLN3nLMD.LCKS bit.

chanhvinh.com

25.3.1.2 RLN3nLBRP0 — LIN Baud Rate Prescaler 0 Register

Access: This register can be read/written in 8-bit units.

Address: <RLIN3n_base> + 02_H

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	LBRP0[7:0]							
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.9 RLN3nLBRP0 register contents

Bit Position	Bit Name	Function
7 to 0	LBRP0[7:0]	Assuming that the value set in this register is N (4 to 255), the baud rate prescaler divides the frequency of the prescaler clock by N + 1. Setting Range: 04_H to FF_H Note: Set the LPRS and LBRP0 bits so that the frequency of the prescaler clock becomes no more than that of the clock source for LIN communications divided by nine.

Set the RLN3nLBRP0 register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

The value set in this register is used to control the frequency of baud rate clock sources fa, fb, and fc.

Assuming that the value set in this register is N, baud rate prescaler 0 divides the frequency of the clock that is selected by the LPRS bits by N + 1.

25.3.1.3 RLN3nLBRP1 — LIN Baud Rate Prescaler 1 Register

Access: This register can be read/written in 8-bit units.

Address: <RLIN3n_base> + 03_H

Initial value: 00H

Bit	7	6	5	4	3	2	1	0
	LBRP1[7:0]							
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.10 RLN3nLBRP1 register contents

Bit Position	Bit Name	Function
7 to 0	LBRP1[7:0]	Assuming that the value set in this register is M (4 to 255), the baud rate prescaler divides the frequency of the prescaler clock by M + 1. Setting Range: 04_H to FF_H Note: Set the LPRS and LBRP1 bits so that the frequency of the prescaler clock becomes no more than that of the clock source for LIN communications divided by nine.

Set the RLN3nLBRP1 register when the OMM0 bit in the RLN2uunLcMST register is 0_B (in LIN reset mode).

The value set in this register is used to control the frequency of baud rate clock source fd.

Assuming that the value set in this register is M, baud rate prescaler 1 divides the frequency of the clock that is selected by the LPRS bits (prescaler clock select bits) by M + 1.

chanhvinh.com

25.3.1.4 RLN3nLSTC — LIN Self-Test Control Register

Access: This register can be read/written in 8-bit units.

Address: <RLIN3n_base> + 04_H

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	—							LSTM
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 25.11 RLN3nLSTC register contents

Bit Position	Bit Name	Function
7 to 1	Reserved	Writing $A7_H$, 58_H, and 01_H successively to the RLN3nLSTC register places the module into LIN self-test mode.
0	LSTM	LIN Self-Test Mode 0: The module is not in LIN self-test mode 1: The module is in LIN self-test mode.

The RLN3nLSTC register cancels protection of LIN self-test mode.

Set the RLN3nLSTC register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

Writing $A7_H$, 58_H, and 01_H successively to the RLN3nLSTC register places the module into LIN self-test mode.

When successive writing is completed thus placing LIN self-test mode to be entered, the LSTM bit is set to 1.

Do not write any other value during successive writing.

For making transition to LIN self-test mode, refer to Section 25.8, LIN Self-Test Mode.

When read, bits 6 to 1 return 000000_B, and bit 7 returns an undefined value.

LSTM bit (LIN self test mode bit)

When transition to LIN self-test mode is completed, the LSTM bit is set to 1.

For leaving LIN self-test mode, refer to Section 25.8, LIN Self-Test Mode.

Writing 1 to this bit does not affect the value of the RLN3nLSTC register if it is not a part of successive writing of $A7_H$, 58_H, and 01_H.

chanhvinh.com

25.3.1.5 RLN3nLMD — LIN Mode Register

Access: This register can be read/written in 8-bit units.

Address: <RLIN3n_base> + 08_H

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	LRDNFS	LIOS	LCKS[1:0]		LMD[1:0]	
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.12 RLN3nLMD register contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, an initial value is returned. When written, write an initial value.
5	LRDNFS	LIN Reception Data Noise Filtering Disable 0: The noise filter is enabled. 1: The noise filter is disabled.
4	LIOS	LIN Interrupt Output Select 0: LINn interrupt is used. 1: LINn transmission interrupt, LINn successful reception interrupt, and LINn status error interrupt are used.
3, 2	LCKS[1:0]	LIN System Clock Select b3 b2 0 0: fa (Clock generated by baud rate prescaler 0) 0 1: fb (1/2 clock generated by baud rate prescaler 0) 1 0: fc (1/8 clock generated by baud rate prescaler 0) 1 1: fd (1/2 clock generated by baud rate prescaler 1)
1, 0	LMD[1:0]	LIN Mode Select b1 b0 0 0: LIN master mode

Set the RLN3nLMD register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

LRDNFS bit (LIN reception data noise filtering disable bit)

The LRDNFS bit enables or disables the noise filter when receiving data.

With 0 set, the noise filter is enabled when receiving data.

With 1 set, the noise filter is disabled when receiving data.

LIOS bit (LIN interrupt output select bit)

The LIOS bit selects the number of interrupt outputs from the LIN Interface.

With 0 set, the LINn interrupt is generated from the LIN Interface.

With 1 set, the LINn transmission interrupt, LINn successful reception interrupt, and LINn status interrupt are generated from the LIN Interface.

For each interrupt source, refer to Section 25.4, Interrupt Sources.

LCKS[1:0] bits (LIN system clock select bits)

The LCKS bits select the clock to be input to the protocol controller.

With 00_B set, the protocol controller is provided with fa (clock generated by baud rate prescaler 0).

chanhvinh.com

With 01_B set, the protocol controller is provided with fb (1/2 clock generated by baud rate prescaler 0).

With 10_B set, the protocol controller is provided with fc (1/8 clock generated by baud rate prescaler 0).

With 11_B set, the protocol controller is provided with fd (1/2 clock generated by baud rate prescaler 1).

With 1_B is set in the LWBR0 bit in the RLN3nLWBR register (LIN 2.x is used), and the RLN3nLMST register is 01_H (LIN wake-up mode), the protocol controller is provided with fa regardless of the setting of the bit (the LCKS bit is not changed).

LMD[1:0] bits (LIN mode select bits)

The LMD bits select the LIN Interface mode.

To use the LIN Interface as an LIN master, set these bits to 00_B.

chanhvinh.com

25.3.1.6 RLN3nLBFC — LIN Break Field Configuration Register

Access: This register can be read/written in 8-bit units.

Address: <RLIN3n_base> + 09_H

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	BDT[1:0]		BLT[3:0]			
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.13 RLN3nLBFC register contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, an initial value is returned. When written, write an initial value.
5, 4	BDT[1:0]	Break Delimiter (High level) width select bit b5 b4 0 0: 1 Tbit 0 1: 2 Tbits 1 0: 3 Tbits 1 1: 4 Tbits
3 to 0	BLT[3:0]	Transmit Break (Low level) width select bit b3 b0 0 0 0 0: 13 Tbits 0 0 0 1: 14 Tbits 0 0 1 0: 15 Tbits : 1 1 1 0: 27 Tbits 1 1 1 1: 28 Tbits

Set the RLN3nLBFC register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

Some combinations of the set values result in the length of a frame exceeding the timeout time. Set the appropriate values in this register.

BDT[1:0] bit (Transmission Break Delimiter high level width setting bit)

This bit is used to set the break high **level** width of transmission frame header.

1 Tbit to 4 Tbits can be set.

BLT[3:0] bit (Transmission Break Low level width setting bit)

This BLT bits set the break low **level** width of transmission frame header.

13 Tbits to 28 Tbits can be set.

chanhvinh.com

25.3.1.7 RLN3nLSC — LIN Space Configuration Register

Access: This register can be read/written in 8-bit units.

Address: <RLIN3n_base> + $0A_H$

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	IBS[1:0]		—	IBHS[2:0]		
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

Table 25.14 RLN3nLSC register contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, an initial value is returned. When written, write an initial value.
5, 4	IBS[1:0]	Inter-Byte Space Select b5 b4 0 0: 0 Tbit 0 1: 1 Tbit 1 0: 2 Tbits 1 1: 3 Tbits
3	Reserved	When read, an initial value is returned. When written, write an initial value.
2 to 0	IBHS[2:0]	Inter-Byte Space (Header)/Response Space Select b2 b0 0 0 0: 0 Tbit 0 0 1: 1 Tbit 0 1 0: 2 Tbits 0 1 1: 3 Tbits 1 0 0: 4 Tbits 1 0 1: 5 Tbits 1 1 0: 6 Tbits 1 1 1: 7 Tbits

Set the RLN3nLSC register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

Some combinations of the set values result in the length of a frame or a response exceeding the timeout time. Set the appropriate values in this register.

IBS[1:0] bits (inter-byte space select bits)

The IBS bits set the width of the inter-byte space of the transmission frame response field.

0 Tbit to 3 Tbits can be set.

These bits are enabled only during response transmission; these are disabled during response reception.

IBHS[2:0] bits (inter-byte space (header)/response space select bits)

The IBHS bits set the width of the inter-byte space (header) of the transmission frame header field and the response space.

0 Tbit to 7 Tbits can be set.

The response space setting is enabled only during response transmission; setting is disabled during response reception.

The inter-byte space (header) is equal to the response space.

chanhvinh.com

25.3.1.8 RLN3nLWUP — LIN Wake-up Configuration Register

Access: This register can be read/written in 8-bit units.

Address: <RLIN3n_base> + $0B_H$

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	WUTL[3:0]				—	—	—	—
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 25.15 RLN3nLWUP register contents

Bit Position	Bit Name	Function
7 to 4	WUTL[3:0]	Wake-up Transmission Low level Width Select b7 b4 0 0 0 0: 1 Tbit 0 0 0 1: 2 Tbits 0 0 1 0: 3 Tbits 0 0 1 1: 4 Tbits : 1 1 0 0: 13 Tbits 1 1 0 1: 14 Tbits 1 1 1 0: 15 Tbits 1 1 1 1: 16 Tbits
3 to 0	Reserved	When read, an initial value is returned. When written, write an initial value.

Set the RLN3nLWUP register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

WUTL[3:0] bits (wake-up transmission low level width select bits)

The WUTL bits set the low level width of the wake-up signal transmission.

1 Tbit to 16 Tbits can be set.

While 1 is set in the LWBR0 bit in the RLN3nLWBR register (LIN 2.x is used), fa is selected as the LIN system clock (fLIN) regardless of the setting of the RLN3nLMD.LCKS bit (the LCKS bit is not changed).

chanhvinh.com

25.3.1.9 RLN3nLIE — LIN Interrupt Enable Register

Access: This register can be read/written in 8-bit units.

Address: <RLIN3n_base> + $0C_H$

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	SHIE	ERRIE	FRCIE	FTCIE
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 25.16 RLN3nLIE register contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, an initial value is returned. When written, write an initial value.
3	SHIE	Successful Header Transmission Interrupt Request Enable 0: Disables successful header transmission interrupt request. 1: Enables successful header transmission interrupt request.
2	ERRIE	Error Detection Interrupt Request Enable 0: Disables error detection interrupt request. 1: Enables error detection interrupt request.
1	FRCIE	Successful Frame/Wake-up Reception Interrupt Request Enable 0: Disables successful frame/wake-up reception interrupt request. 1: Enables successful frame/wake-up reception interrupt request.
0	FTCIE	Successful Frame/Wake-up Transmission Interrupt Request Enable 0: Disables successful frame/wake-up transmission interrupt request. 1: Enables successful frame/wake-up transmission interrupt request.

Set the RLN3nLIE register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

SHIE bit (successful header transmission interrupt enable bit)

The SHIE bit enables or disables interrupt request upon successful transmission of a header.

With 0 set, the interrupt request for LINn transmission is not generated when the HTRC flag in the RLN3nLST register is set to 1.

With 1 set, the interrupt request for LINn transmission is generated when the HTRC flag in the RLN3nLST register is set to 1.

ERRIE bit (error detection interrupt request enable bit)

The ERRIE bit enables or disables interrupt request upon detection of an error.

With 0 set, the interrupt request for LINn status is not generated when the ERR flag in the RLN3nLST register is set to 1.

With 1 set, the interrupt request for LINn status is generated when the ERR flag in the RLN3nLST register is set to 1.

Interrupt sources can be the bit error, physical bus error, frame/response timeout error, framing error, checksum error, and response preparation error.

Detection of the bit error, physical bus error, frame/response timeout error, and framing error can be enabled or disabled using the RLN3nLEDE register.

chanhvinh.com

FRCIE bit (successful frame/wake-up reception interrupt request enable bit)

The FRCIE bit enables or disables interrupt request upon successful reception of a frame or a wake-up signal (counting of low level width of the input signal).

With 0 set, the interrupt request for successful LINn reception is not generated when the FRC flag in the RLN3nLST register is set to 1.

With 1 set, the interrupt request for successful LINn reception is generated when the FRC flag in the RLN3nLST register is set to 1.

FTCIE bit (successful frame/wake-up transmission interrupt request enable bit)

The FTCIE bit enables or disables interrupt request upon successful transmission of a frame or a wake-up signal.

With 0 set, the interrupt request for LINn transmission is not generated when the FTC flag in the RLN3nLST register is set to 1.

With 1 set, the interrupt request for LINn transmission is generated when the FTC flag in the RLN3nLST register is set to 1.

chanhvinh.com

25.3.1.10 RLN3nLEDE —LIN Error Detection Enable Register

Access: This register can be read/written in 8-bit units.

Address: <RLIN3n_base> + $0D_H$

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	LTES	—	—	—	FERE	FTERE	PBERE	BERE
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R/W	R/W	R/W	R/W

Table 25.17 RLN3nLEDE register contents

Bit Position	Bit Name	Function
7	LTES	Timeout Error Select 0: Frame timeout error 1: Response timeout error
6 to 4	Reserved	When read, an initial value is returned. When written, write an initial value.
3	FERE	Framing Error Detection Enable 0: Disables framing error detection. 1: Enables framing error detection.
2	FTERE	Timeout Error Detection Enable 0: Disables frame/response timeout error detection. 1: Enables frame/response timeout error detection.
1	PBERE	Physical Bus Error Detection Enable 0: Disables physical bus error detection. 1: Enables physical bus error detection.
0	BERE	Bit Error Detection Enable 0: Disables bit error detection. 1: Enables bit error detection.

Set the RLN3nLEDE register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

LTES bit (timeout error select bit)

The LTES bit selects the specific timeout function to be used.

With 0 set, the timeout function applies to frame timeout.

With 1 set, the timeout function applies to response timeout.

For details of the timeout error, refer to Section 25.7.6, Error Status.

FERE bit (framing error detection enable bit)

The FERE bit enables or disables detection of the framing error.

With 0 set, the framing error is not detected.

With 1 set, the framing error is detected.

When this bit is set to 1, the detection result is indicated in the FER flag in the RLN3nLEST register.

For details of the framing error, refer to Section 25.7.6, Error Status.

chanhvinh.com

FTERE bit (timeout error detection enable bit)

The FTERE bit enables or disables detection of the frame timeout error or the response timeout error.

With 0 set, the frame timeout error or response timeout error is not detected.

With 1 set, the frame timeout error or response timeout error is detected.

When this bit is set to 1, the detection result is indicated in the FTER flag in the RLN3nLEST register.

With the LTES bit, either the frame timeout error or response timeout error can be selected.

Do not use the timeout error if response data of 9 bytes or more is to be transmitted or received.

For details of the timeout error, refer to Section 25.7.6, Error Status.

PBERE bit (physical bus error detection enable bit)

The PBERE bit enables or disables detection of the physical bus error.

With 0 set, the physical bus error is not detected.

With 1 set, the physical bus error is detected.

When this bit is set to 1, the detection result is indicated in the PBER flag in the RLN3nLEST register.

For details of the physical bus error, refer to Section 25.7.6, Error Status.

BERE bit (bit error detection enable bit)

The BERE bit enables or disables detection of the bit error.

With 0 set, the bit error is not detected.

With 1 set, the bit error is detected.

When this bit is set to 1, the detection result is indicated in the BER flag in the RLN3nLEST register.

For details of the bit error, refer to Section 25.7.6, Error Status.

chanhvinh.com

25.3.1.11 RLN3nLCUC — LIN Control Register

Access: This register can be read/written in 8-bit units.

Address: <RLIN3n_base> + $0E_H$

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OM1	OM0
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 25.18 RLN3nLCUC register contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, an initial value is returned. When written, write an initial value.
1	OM1	LIN Mode Select 0: LIN wake-up mode is caused. 1: LIN operation mode is caused.
0	OM0	LIN Reset 0: LIN reset mode is caused. 1: LIN reset mode is canceled.

Set the RLN3nLCUC register to 01_H to cause a transition to LIN wake-up mode after canceling LIN reset mode, and set the register to 03_H to cause a transition to LIN operation mode.

In LIN self-test mode, set the RLN3nLCUC register to 03h after a transition to LIN self-test mode is completed.

After a value is written to this register, confirm that the value written is actually indicated in the RLN3nLMST register before writing another value.

OM1 bit (LIN mode select bit)

The OM1 bit selects the specific LIN operation mode (either LIN wake-up mode or LIN operation mode) after canceling LIN reset mode.

With 0 set, LIN wake-up mode is caused.

With 1 set, LIN operation mode is caused.

This bit is valid only when the OMM0 bit in the RLN3nLMST register is 1.

Writing a value to this bit is disabled while the FTS bit in the RLN3nLTRC register is 1.

OM0 bit (LIN reset bit)

The OM0 bit selects either causing a transition to LIN reset mode or canceling LIN reset mode.

With 0 set, LIN reset mode is caused.

With 1 set, LIN reset mode is canceled.

chanhvinh.com

25.3.1.12 RLN3nLTRC — LIN Transmission Control Register

Access: This register can be read/written in 8-bit units.

Address: <RLIN3n_base> + 10_H

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RTS	FTS
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 25.19 RLN3nLTRC register contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, an initial value is returned. When written, write an initial value.
1	RTS	Response Transmission/Reception Start 0: Response transmission/reception is stopped in frame separate mode. 1: Response transmission/reception is started in frame separate mode. Note: Setting of the frame separate mode is prohibited in this product.
0	FTS	Frame Transmission/wake-up Transmission /Reception Start 0: Frame Transmission/wake-up transmission /reception is stopped. 1: Frame Transmission/wake-up transmission reception is started.

RTS bit (response transmission/reception start bit)

Set the RTS bit to 1 in frame separate mode after header transmission is started (FTS bit is 1) and response transmission data is ready. Once set, this bit is automatically cleared to 0 upon completion of frame communication (including error detection) or transition to LIN reset mode.
Only 1 can be written to this bit; 0 cannot be written. To write 1 to this bit, write 02_H to the RLN3nLTRC register using the store instruction.

Writing a value to this bit is disabled when the OMM0 bit of the RLN3nLMST register is 0_B (in LIN reset mode).

Writing a value to this bit is disabled when the FTS bit is 0 (frame transmission or wake-up transmission/reception is halted).

When response data of 9 bytes or more is to be transmitted or received, set this bit to 1 each time a data group (variable from 0 to 8 bytes) is transmitted or received. Once set, this bit is automatically cleared to 0 upon completion of data group communication or transition to LIN reset mode.

FTS bit (frame transmission/wake-up transmission/reception start bit)

Set the FTS bit to 1 to start frame transmission and reception.

Also set this bit to 1 to allow wake-up transmission and wake-up reception (counting of the low level width of the input signal).

Only 1 can be written to this bit; 0 cannot be written. Writing a value to this bit is disabled when the OMM0 bit of the RLN3nLMST register is 0_B (in LIN reset mode).

This bit is set to 0 upon completion of frame or wake-up communication (including error detection) and transition to LIN reset mode.

chanhvinh.com

25.3.1.13 RLN3nLMST — LIN Mode Status Register

Access: This register can be read/written in 8-bit units.

Address: <RLIN3n_base> + 11_H

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OMM1	OMM0
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 25.20 RLN3nLMST register contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, an initial value is returned. When written, write an initial value.
1	OMM1	LIN Mode Status Monitor 0: The module is in LIN wake-up mode. 1: The module is in LIN operation mode.
0	OMM0	LIN Reset Status Monitor 0: The module is in LIN reset mode. 1: The module is not in LIN reset mode.

OMM1 bit (LIN mode status monitor)

The OMM1 bit indicate the current operating mode.

OMM0 bit (LIN reset status monitor)

The OMM0 bit indicates the current operating mode.

chanhvinh.com

25.3.1.14 RLN3nLST — LIN Status Register

Access: This register can be read/written in 8-bit units.

Address: <RLIN3n_base> + 12_H

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	HTRC	D1RC	—	—	ERR	—	FRC	FTC
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R/W	R/W

Table 25.21 RLN3nLST register contents

Bit Position	Bit Name	Function
7	HTRC	Successful Header Transmission Flag 0: Header transmission has not been completed. 1: Header transmission has been completed.
6	D1RC	Successful Data 1 Reception Flag These bits are always read as 0. The write value should always be 0.
5, 4	Reserved	When read, an initial value is returned. When written, write an initial value.
3	ERR	Error Detection Flag 0: No error has been detected. 1: Error has been detected.
2	Reserved	When read, an initial value is returned. When written, write an initial value.
1	FRC	Successful Frame/Wake-up Reception Flag 0: Frame or wake-up reception has not been completed. 1: Frame or wake-up reception has been completed.
0	FTC	Successful Frame/Wake-up Transmission Flag 0: Frame or wake-up transmission has not been completed. 1: Frame or wake-up transmission has been completed.

The RLN3nLST register is automatically cleared to 00_H upon transition to LIN reset mode and start of the next communication (when the FTS bit of the RLN3nLTRC register is 1).

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00_H. To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

HTRC flag (successful header transmission flag)

Only 0 can be written to the HTRC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The HTRC flag is set to 1 upon completion of header transmission. Here, an interrupt request for LINn transmission is generated if the SHIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.

D1RC flag (successful data 1 reception flag)

Only 0 can be written to the D1RC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The D1RC flag is set to 1 upon completion of data 1 reception. Here, an interrupt request is not generated. To clear the bit to 0 before the next communication (when the value of the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.

chanhvinh.com

When response data of 9 bytes or more is to be received, this bit is set to 1 each time data 1 of a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

ERR flag (error detection flag)

The ERR flag is set to 1 upon detection of an error (when the value of any of the flags of the RLN3nLEST registers is 1). Here, an interrupt request for LINn status is generated if the ERRIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (when the value of the FTS bit of the RLN3nLTRC register is 1), write 0 to the RPER, CSER, FER, FTER, PBER, and BER flags in the RLN3nLEST register in LIN operation mode or LIN wake-up mode. This clears the ERR flag to 0.

FRC flag (successful frame/wake-up reception flag)

Only 0 can be written to the FRC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The FRC flag is set to 1 upon completion of frame or wake-up reception. Here, an interrupt request for successful LINn reception is generated if the FRCIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (when the value of the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

When response data of 9 bytes or more is to be received, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

FTC flag (successful frame/wake-up transmission flag)

Only 0 can be written to the FTC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The FTC flag is set to 1 upon completion of frame or wake-up transmission. Here, an interrupt request for LINn transmission is generated if the FTCIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (when the value of the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

When response data of 9 bytes or more is to be transmitted, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is transmitted. Write 0 before starting transmission of the next data group.

chanhvinh.com

25.3.1.15 RLN3nLEST — LIN Error Status Register

Access: This register can be read/written in 8-bit units.

Address: <RLIN3n_base> + 13_H

Initial value: 00H

Bit	7	6	5	4	3	2	1	0
	RPER	—	CSER	—	FER	FTER	PBER	BER
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R	R/W	R/W	R/W	R/W

Table 25.22 RLN3nLEST register contents

Bit Position	Bit Name	Function
7	RPER	Response Preparation Error Flag 0: Response preparation error has not been detected. 1: Response preparation error has been detected.
6	Reserved	When read, an initial value is returned. When written, write an initial value.
5	CSER	Checksum Error Flag 0: Checksum error has not been detected. 1: checksum error has been detected.
4	Reserved	When read, an initial value is returned. When written, write an initial value.
3	FER	Framing Error Flag 0: Framing error has not been detected. 1: Framing error has been detected.
2	FTER	Timeout Error Flag 0: Frame/response timeout error has not been detected. 1: Frame/response timeout error has been detected.
1	PBER	Physical Bus Error Flag 0: Physical bus error has not been detected. 1: Physical bus error has been detected.
0	BER	Bit Error Flag 0: Bit error has not been detected. 1: Bit error has been detected.

The RLN3nLEST register is automatically cleared to 00_H upon transition to LIN reset mode and start of the next communication (when the value of the FTS bit of the RLN3nLTRC register is 1).

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00_H.

When the FTS bit in the RLN3nLTRC register is 1 (frame transmission or wake-up transmission/ reception is started), do not write a value to this register. To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

RPER flag (response preparation error flag)

Only 0 can be written to the RPER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The RPER flag is set to 1 upon response preparation error detection. To clear the bit to 0 before the next communication (when the value of the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.

chanhvinh.com

CSER flag (checksum error flag)

Only 0 can be written to the CSER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The CSER flag is set to 1 upon checksum error detection. To clear the bit to 0 before the next communication (when the value of the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.

FER flag (framing error flag)

Only 0 can be written to the FER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The FER flag is set to 1 upon frame timeout detection when the FERE bit of the RLN3nLEDE register is 1 (frame timeout detection enabled). To clear the bit to 0 before the next communication (when the value of the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.

FTER flag (timeout error flag)

Only 0 can be written to the FTER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The FTER flag is set to 1 upon frame timeout error or response timeout error detection when the FTERE bit of the RLN3nLEDE register is 1 (frame/response timeout error detection enabled). To clear the bit to 0 before the next communication (when the value of the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.

PBER flag (physical bus error flag)

Only 0 can be written to the PBER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The PBER flag is set to 1 upon physical bus error detection when the PBERE bit of the RLN3nLEDE register is 1 (physical bus error detection enabled). To clear the bit to 0 before the next communication (when the value of the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

BER flag (bit error flag)

Only 0 can be written to the BER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The BER flag is set to 1 upon bit error detection when the BERE bit of the RLN3nLEDE register is 1 (bit error detection enabled). To clear the bit to 0 before the next communication (when the value of the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

chanhvinh.com

25.3.1.16 RLN3nLDFC — LIN Data Field Configuration Register

Access: This register can be read/written in 8-bit units.

Address: <RLIN3n_base> + 14_H

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	LSS	FSM	CSM	RFT	RFDL[3:0]			
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.23 RLN3nLDFC register contents

Bit Position	Bit Name	Function
7	LSS	Transmission/Reception Continuation Select 0: The data group to be transmitted/received next is the last one. 1: The data group to be transmitted/received next is not the last one. (Checksum is not included.)
6	FSM	Frame Separate Mode Select 0: Frame separate mode is not set. 1: Frame separate mode is set. Note: Setting of the frame separate mode is prohibited in this product.
5	CSM	Checksum Select 0: Classic checksum mode 1: Enhanced checksum mode
4	RFT	Response Field Communication Direction Select 0: Reception 1: Transmission
3 to 0	RFDL[3:0]	Response Field Length Select b3 b0 0 0 0 0: 0 byte (+ checksum) 0 0 0 1: 1 byte (+ checksum) 0 0 1 0: 2 bytes (+ checksum) : 0 1 1 1: 7 bytes (+ checksum) 1 0 0 0: 8 bytes (+ checksum) Settings other than the above are prohibited.

LSS bit (transmission/reception continuation select bit)

The LSS bit indicates that the data group to be transmitted or received next is not the last data group when response data of 9 bytes or more is to be transmitted or received. With 0 set, data and checksum are transmitted or received because the next data group to be transmitted or received is the last one.

With 1 set, only data is transmitted or received, and the checksum is not included because the next data group to be transmitted or received is not the last one.

Set the LSS bit only when the FSM bit is 1 (frame separate mode) and response data of 9 bytes or more is to be transmitted or received.

Set the LSS bit only when the RTS bit in the RLN3nLTRC is 0 (response transmit/receive is stopped).

FSM bit (frame separate mode select bit)

The FSM bit sets the response communication mode.

chanhvinh.com

With 0 set, frame separate mode is not selected. In this case, after header transmission is started (the FTS bit in the RLN3nLTRC register is 1), response is transmitted/received without the RTS bit in the RLN3nLTRC register being set.

With 1 set, frame separate mode is selected. If the RTS bit of the RLN3nLTRC register is set to 1 during header transmission, response transmission is executed after header transmission is completed.

For response reception which is 8 bytes or less (the RFT bit is 0), set the FSM bit to 0.
When causing a transition to LIN self-test mode, set this bit to 0 before transition.
For details of frame separate mode, refer to Section 25.7.3.1, Transmission of LIN Frames.
Set this bit when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

When response data of 9 bytes or more is to be transmitted or received, set the FSM bit to 1.

CSM bit (checksum select bit)

The CSM bit sets the checksum mode.
With 0 set, classic checksum mode is selected.
With 1 set, enhanced checksum mode is selected.
When the timeout error is used (the FTERE bit in the RLN3nLEDE register is 1), the specific timeout time depends on the setting of this bit. For details of the bit error, refer to Section 25.7.6, Error Status.
Set this bit when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

When response data of 9 bytes or more is to be transmitted or received, do not change the CSM bit setting after the first data group through the last data group.

During communication of response data of 9 bytes or more, only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit is 1) include the checksum.

RFT bit (response field communication direction select bit)

The RFT bits set the direction of the response field/wake-up signal communication.
With 0 set, reception is performed in the response field. In LIN wake-up mode, wake-up reception is performed (low level width of the input signal is counted).
With 1 set, transmission is performed in the response field. In LIN wake-up mode, wake-up transmission is performed.
Set this bit when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

When response data of 9 bytes or more is to be transmitted or received, do not change the RFT bit setting after the first data group through the last data group.

RFDL[3:0] bits (response field length select bits)

The RFDL bits set the length of the response field data.
The data length can be 0 to 8 bytes excluding the checksum size.
To transmit response data with the FSM bit set to 0 (not frame separate mode), set the RFDL bits before header transmission (the FTS bit in the RLN3nLTRC register is 0).
To transmit response data with the FSM bit set to 1 (frame separate mode), set the RFDL bits before response transmission (the FTS bit in the RLN3nLTRC register is 0).
To receive response data, set the RFDL bits before header transmission (the FTS bit in the RLN3nLTRC register is 0).

chanhvinh.com

When response data of 9 bytes or more is to be transmitted or received, set the RFDL bits before data group transmission/reception (RTS bit in the RLN3nLTRC register is 0).

Only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit is 1) include the checksum.

chanhvinh.com

25.3.1.17 RLN3nLIDB — LIN ID Buffer Register

Access: This register can be read/written in 8-bit units.

Address: <RLIN3n_base> + 15_H

Initial value: 00_H

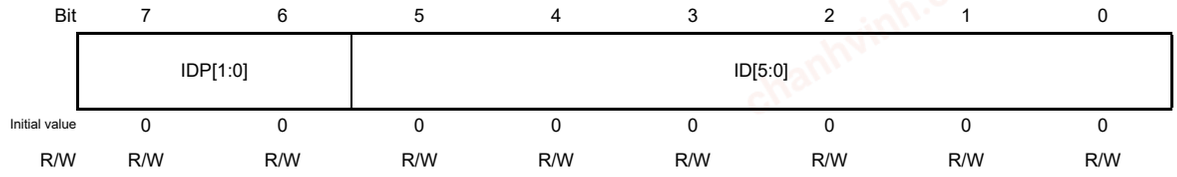


Table 25.24 RLN3nLIDB register contents

Bit Position	Bit Name	Function
7	IDP1	Parity Setting (P1) Sets the parity bit (P1) to be transmitted in the ID field.
6	IDP0	Parity Setting (P0) Sets the parity bit (P0) to be transmitted in the ID field.
5 to 0	ID[5:0]	ID Setting Sets the 6-bit ID value to be transmitted in the ID field.

Set the RLN3nLIDB register when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

In LIN self-test mode, this register operates as follows:

Write the value to be transmitted before communication. After completion of frame transmission/ reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see Section 25.8, LIN Self-Test Mode.

IIDP[1:0] bits (parity setting bits)

The IDP bits set the parity bits (P0 and P1) to be transmitted in the ID field of the LIN frame (IDP0 for P0 and IDP1 for P1). Since parity is not automatically calculated, set the calculation result. Note that if the erroneous result is set, it is transmitted as is.

ID[5:0] bits (ID setting bits)

The ID bit sets the 6-bit ID value to be transmitted in the ID field of the LIN frame.

chanhvinh.com

25.3.1.18 RLN3nLCBR — LIN Checksum Buffer Register

Access: This register can only be read in 8-bit units. In LIN self-test mode, this register can be read/written in 8-bit units.

Address: <RLIN3n_base> + 16_H

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	CKSM[7:0]							
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.25 RLN3nLCBR register contents

Bit Position	Bit Name	Function
7 to 0	CKSM[7:0]	Holds the checksum value transmitted or received.

In LIN mode, this register operates as follows:

- When the RFT bit in the RLN3nLDFC register is 1 (transmission):
  The value transmitted can be read from the register. Read the value after transmission is completed.
  Writing to this register is invalid.
- When the RFT bit in the RLN3nLDFC register is 0 (reception):
  The value received can be read from the register. Read the value after reception is completed.
  Writing to this register is invalid.

In LIN self-test mode, this register operates as follows:

- When the RFT bit in the RLN3nLDFC register is 1 (transmission):
  After completion of the frame transmission (after loopback), the reversed value of the received value can be read.
- When the RFT bit in the RLN3nLDFC register is 0 (reception):
  Write the value to be received before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.
  For details about the LIN self-test mode, see Section 25.8, LIN Self-Test Mode.

Set the RLN3nLCBR register when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

When response data of 9 bytes or more is to be transmitted or received, the checksum is appended only to the last data group; this register is not updated for the other data groups.

chanhvinh.com

25.3.1.19 RLN3nLDBRm — LIN Data Buffer m Register (m = 1 to 8)

Access: This register can be read/written in 8-bit units.

Address: RLN3nLDBR1: <RLIN3n_base> + 18_H
RLN3nLDBR2: <RLIN3n_base> + 19_H
RLN3nLDBR3: <RLIN3n_base> + $1A_H$
RLN3nLDBR4: <RLIN3n_base> + $1B_H$
RLN3nLDBR5: <RLIN3n_base> + $1C_H$
RLN3nLDBR6: <RLIN3n_base> + $1D_H$
RLN3nLDBR7: <RLIN3n_base> + $1E_H$
RLN3nLDBR8: <RLIN3n_base> + $1F_H$

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	LDB[7:0]							
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.26 RLN3nLDBRm (m = 1 to 8) register contents

Bit Position	Bit Name	Function
7 to 0	LDB[7:0]	Sets the data to be transmitted or allows the received data to be read. Setting Range: 00_H to FF_H

For response transmission:

The LDBRn registers set the data to be transmitted in the response field.
Use these registers with the following settings.

- RFT in RLN3nLDFC register is 1 (transmission)
- FSM in RLN3nLDFC register is 0 (not frame separate mode)
- FTS bit in RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted)

or

- RFT in RLN3nLDFC register is 1 (transmission)
- FSM in RLN3nLDFC register is 1 (frame separate mode)
- RTS in RLN3nLTRC register is 0 (response transmission/reception is halted)

For response reception:

The LDBRn registers hold the data received in the response field.
The received data is overwritten. If an error is detected, the data up to the byte in which the error was detected are stored in the register. Do not read these registers when the FTS bit is 1 (frame transmission or wake-up transmission/reception is started)

chanhvinh.com

For transmission of response data of 9 bytes or more:

Use the LDBRn registers with the following settings.

- RFT in RLN3nLDFC register is 1 (transmission)
- FSM in RLN3nLDFC register is 1 (frame separate mode)
- RTS in RLN3nLTRC register is 0 (response transmission/reception is halted)

For reception of response data of 9 bytes or more:

Do not read these registers when the RTS bit is 1 (response transmission/reception is started).

In LIN self-test mode, these registers operate as follows:

Write the value to be transmitted before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see Section 25.8, LIN Self-Test Mode.

chanhvinh.com

25.4 Interrupt Sources

The LIN Interface generates four types of interrupt requests.

- LINn transmission interrupt
- LINn successful reception interrupt
- LINn status interrupt
- LINn interrupt

Setting the LIOS bit in the RLN3nLMD register to 0 allows to perform logical OR operation on all of the interrupt sources, outputting the interrupt request from the LINn interrupt.

Setting the LIOS bit in the RLN3nLMD register to 1 allows to output the LINn transmission interrupt, LINn successful reception interrupt, or LINn status interrupt depending on the interrupt request.

Table 25.27 lists the sources for each interrupt.

Table 25.27 Interrupt Sources

		LIOS bit in RLN3nLMD register is 0	LIOS bit in RLN3nLMD register is 1		
		LINn Interrupt	**LINn Transmission Interrupt**	**LINn Successful Reception Interrupt**	**LINn Status Interrupt**
LIN mode	LIN master mode	• Successful frame transmission • Successful frame reception • Successful wake-up transmission • Successful wake-up reception • Successful header transmission • Bit error • Physical bus error • Frame/response timeout error • Framing error • Checksum error • Response preparation error	• Successful frame transmission • Successful wake-up transmission • Successful header transmission	• Successful wake-up reception • Successful wake-up reception	• Bit error • Physical bus error • Frame/response timeout error • Framing error • Checksum error • Response preparation error

Each interrupt request is output when the corresponding bit in the RLN3nLIE register is 1 (interrupt is enabled) and the corresponding flag in the RLN3nLST register is 1.

chanhvinh.com

25.5 Modes

The LIN Interface provides the following three modes, depending upon the specific function to be performed:

- LIN reset mode
- LIN mode (LIN master mode)
- LIN self-test mode

The supply of clocks to the LIN Interface is stopped in LIN reset mode, which reduces power consumption.

Figure 25.2 shows mode transitions. Table 25.28 describes mode transition conditions. Table 25.29 lists operations available in each mode.

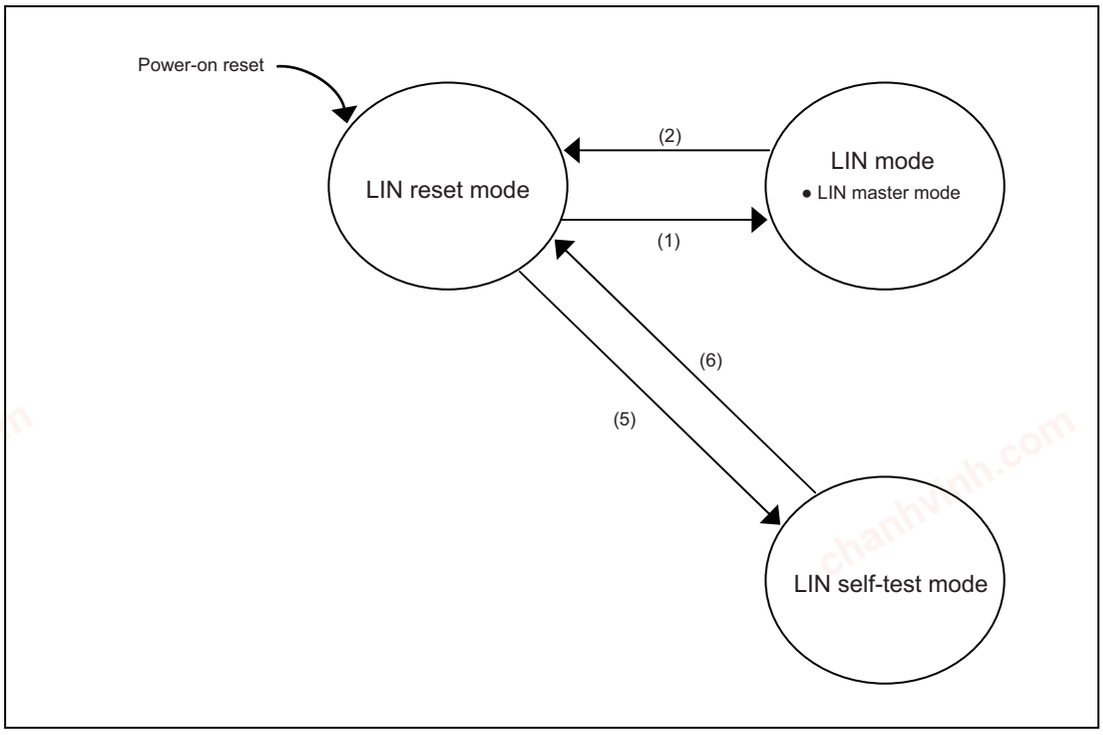


Figure 25.2 Mode Transitions

Table 25.28 Transition Condition of Each Mode

	Mode transition			Transition condition
1	LIN reset mode	→	LIN mode (LIN master mode)	RLN3nLMD.LMD = 00_B and RLN3nLCUC.OM1, OM0 = 01_B or 11_B
2	LIN mode	→	LIN reset mode	RLN3nLCUC.OM0 = 0_B
5	LIN reset mode	→	LIN self-test mode	See Section 25.8, LIN Self-Test Mode.
6	LIN self-test mode	→	LIN reset mode	See Section 25.8, LIN Self-Test Mode.

chanhvinh.com

Table 25.29 Operations Available in Each Mode

LIN mode	
LIN master mode	**LIN self-test mode**
Header transmission Response transmission Response reception Wake-up transmission Wake-up reception Error detection	Self test

Whether a transition has been caused to LIN reset mode or LIN mode can be verified by reading the LMD bits in the RLN3nLMD register or the OMM0 bit in the RLN3nLMST register.

For a description of the LIN self-test mode, see Section 25.8, LIN Self-Test Mode.

chanhvinh.com

25.6 LIN Reset Mode

Setting the OM0 bit in the RLN3nLCUC register to 0 (LIN reset mode) causes a transition to LIN reset mode. The change to LIN reset mode can be verified by determining that the OMM0 bit in the RLN3nLMST register has been set to 0 (LIN reset mode). In this mode, the LIN communication function is halted.

When a DMA channel is activated by a LIN transfer request at the time of a transition to LIN reset mode, stop the channel.

For stopping a DMA channel, see Section 9., Direct Memory Access Controller.

From LIN reset mode, transitions to LIN mode and LIN self-test mode can be made.

When the mode changes to LIN reset mode, the following registers are initialized to their reset values, and as long as LIN reset mode is in effect, they retain their initial values.

- RLN3nLTRC register
- RLN3nLST register
- RLN3nLEST register

The following registers retain their previous values even when a transition to LIN reset mode is made:

- RLN3nLWBR register
- RLN3nLBRP0 register
- RLN3nLBRP1 register
- RLN3nLMD register
- RLN3nLBFC register
- RLN3nLSC register
- RLN3nLWUP register
- RLN3nLIE register
- RLN3nLEDE register
- RLN3nLDFC register
- RLN3nLIDB register
- RLN3nLCBR register
- RLN3nLDBRm register (m = 1 to 8)

chanhvinh.com

25.7 LIN Mode

LIN mode can operate in the LIN master mode.

In LIN master mode, the following operations can be performed: header transmission, response transmission, response reception, wake-up transmission, wake-up reception, and error detection. In LIN reset mode, setting the LMD bits in the RLN3nLMD register to 00_B (LIN master mode) and the OM1 and OM0 bits in the RLN3nLCUC register to either 01_B or 11_B sets LIN master mode, turning the OMM1 and OMM0 bits in the RLN3nLMST register to either 01_B to 11_B.

The LIN mode provides the following two operation modes:

- LIN operation mode
- LIN wake-up mode

Figure 25.3 shows the transition of operation modes. Table 25.30 describes the transition conditions of operation modes.

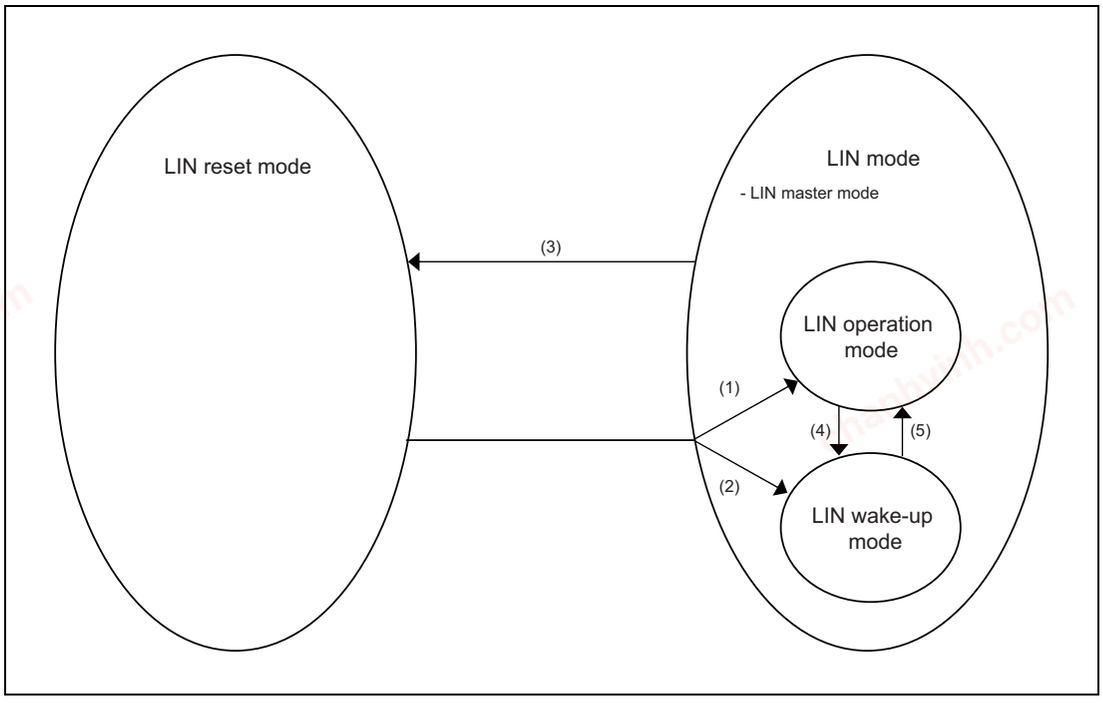


Figure 25.3 Transition of Operation Modes

chanhvinh.com

Table 25.30 Transition Conditions of Operation Modes

	Operation mode transition			Transition condition
(1)	LIN reset mode	→	LIN mode - LIN operation mode	RLN3nLMD.LMD = 00_B and RLN3nLCUC.OM1, OM0 = 11_B
(2)	LIN reset mode	→	LIN mode - LIN wake-up mode	RLN3nLMD.LMD = 00_B and RLN3nLCUC.OM1, OM0 = 01_B
(3)	LIN mode -LIN operation mode -LIN wake-up mode	→	LIN reset mode	RLN3nLCUC.OM0 = 0_B
(4) *1	LIN mode -LIN operation mode	→	LIN mode - LIN wake-up mode	RLN3nLCUC.OM1, OM0 = 01_B
(5) *1	LIN mode -LIN wake-up mode	→	LIN mode - LIN operation mode	RLN3nLCUC.OM1, OM0 = 11_B

Note 1. Transition between LIN operation mode and LIN wake-up mode cannot be made when communication is going on (when the FTS bit in the RLN3nLTRC register is 1).

(1) LIN Operation Mode

In LIN operation mode, frame processing (header transmission, header reception, response transmission, response reception, and error detection) can be performed.

During a transition from LIN reset mode to LIN mode, setting the OM1 and OM0 bits in the RLN3nLCUC register to 11_B changes the mode to LIN operation mode, changing the OMM1 and OMM0 bits in the RLN3nLMST register to 11_B. Communication settings should be performed after the RLN3nLMST register has become 11_B.

(2) LIN Wake-up Mode

In LIN wake-up mode, wake-up signal processing (wake-up transmission, wake-up reception, and error detection) can be performed.

During a transition from LIN reset mode to LIN mode, setting the OM1 and OM0 bits in the RLN3nLCUC register to 01_B changes the mode to LIN wake-up mode, changing the OMM1 and OMM0 bits in the RLN3nLMST register to 01_B. Communication settings should be performed after the RLN3nLMST register has become 01_B.

chanhvinh.com

25.7.1 LIN Master Mode

25.7.1.1 Header Transmission

Figure 25.4 shows the operation of the LIN Interface (LIN master mode) in header transmission.
Table 25.31 provides processing in header transmission.

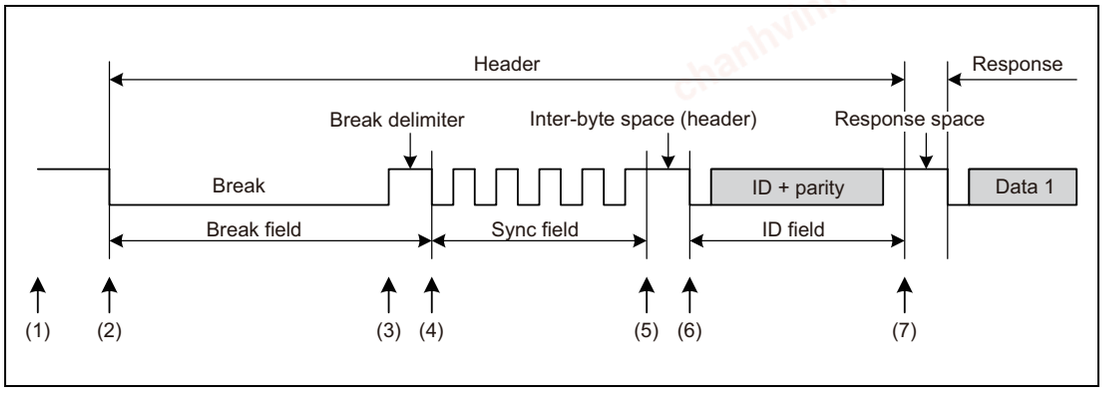


Figure 25.4 Operation in Header Transmission

Table 25.31 Processing in Header Transmission

	Software processing	LIN Interface processing
(1)	• Sets a baud rate • Sets noise filter ON/OFF • Enables interrupt • Enables error detection • Sets frame configuration parameters • Changes the LIN Interface to the LIN master mode: LIN operation mode • Sets information on the frame to be transmitted (ID, parity, data length, response direction, Checksum method, and transmission data)	Waits for the setting of the FTS bit in the RLN3nLTRC register by software (idle)
(2)	Sets the FTS bit in the RLN3nLTRC register to 1 (frame transmission or wake-up transmission/reception started)	Transmits a break.
(3)	Waits for an interrupt request	Transmits a break delimiter.
(4)		Transmits a sync field (55h).
(5)		Transmits an inter-byte space (header).
(6)		Transmits an ID field.
(7)		Sets a successful header transmission flag.

Note: For information about error detection, refer to Section 25.7.6, Error Status.

chanhvinh.com

25.7.1.2 Response Transmission

Figure 25.5 shows the operation of the LIN Interface (LIN master mode) in response transmission. Table 25.32 provides processing in response transmission.

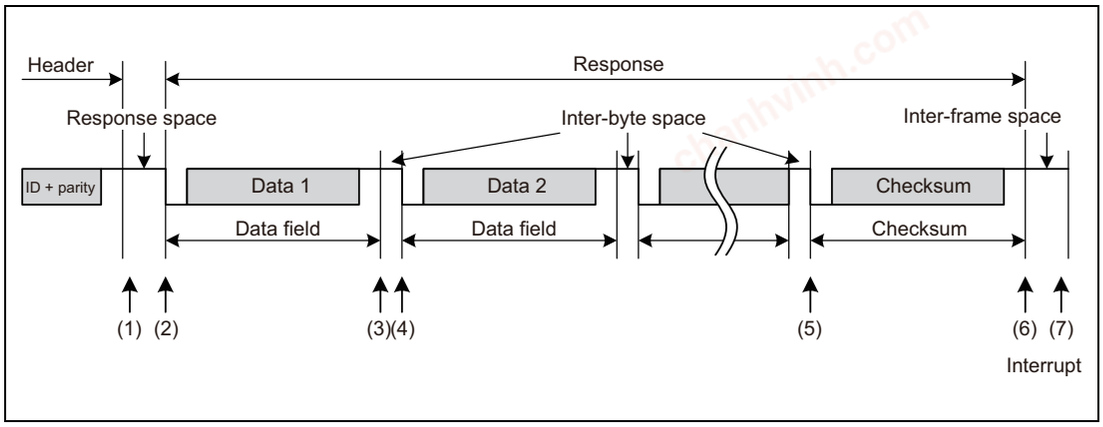


Figure 25.5 Operation in Response Transmission

Table 25.32 Processing in Response Transmission

	Software processing	LIN Interface processing
(1)	(When in frame separate mode) • Sets the RTS bit in the RLN3nLTRC register to 1 (response transmission/ reception started) (When not in frame separate mode) • Waits for an interrupt request	(When in frame separate mode) • Waits for the setting of the RTS bit in the RLN3nLTRC register to 1 by software. • When the bit is set to 1, sends a response space. (When not in frame separate mode) • Sends a response space.
(2)	Waits for an interrupt request	Transmits the data 1.
(3)		Transmits an inter-byte space.
(4)		• Transmits the data 2. • Transmits an inter-byte space • Transmits the data 3. • Transmits an inter-byte space (Repeats the transmission of inter-byte spaces as many times as the data length specified in bits RFDL[3:0] in the RFC register.) : :
(5)		Transmits the checksum.
(6)		• Sets a successful frame/wake-up transmission flag. • Sets the FTS bit in the RLN3nLTRC register to 0 (frame transmission or wake-up transmission/reception stopped) (When in frame separate mode), and the RTS bit in the RLN3nLTRC register to 0 (response transmission/reception stopped).
(7)	• Processing after communication Checks the RLN3nLST register, and clears flags.	Idle

Note: For information about error detection, refer to Section 25.7.6, Error Status.

chanhvinh.com

25.7.1.3 Response Reception

Figure 25.6 shows the operation of the LIN Interface (LIN master mode) on response reception. Table 25.33 provides processing in response reception.

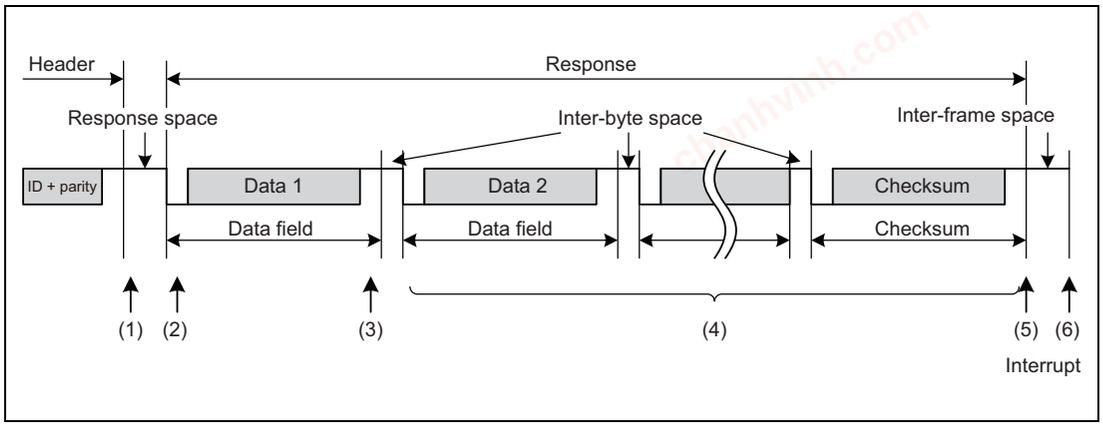


Figure 25.6 Operation in Response Reception

Table 25.33 Processing in Response Reception

	Software processing	LIN Interface processing
(1)	Waits for an interrupt request (no processing).	Waits for detection of a start bit.
(2)	Waits for an interrupt request.	Receives the data 1 when the start bit is detected.
(3)		Sets the successful data 1 reception flag.
(4)		• Receives the data 2 when the start bit is detected. • Receives the data 3 when the start bit is detected. (Repeats the transmission of inter-byte spaces as many times as the data length specified in bits RFDL[3:0] in the RLN3nLDFC register.) : : • Receives the checksum when the start bit is detected.
(5)		• Determines the checksum. • Sets the successful frame/wake-up reception flag. • Sets the RTS bit in the RLN3nLTRC register to 0 (response transmission/reception stopped).
(6)	• Processing after communication Reads the received data. Checks the RLN3nLST register, and clears flags.	Idle

Note: For information about error detection, refer to Section 25.7.6, Error Status.

chanhvinh.com

25.7.2 Data Transmission/Reception

25.7.2.1 Data Transmission

One bit of data is transmitted per 1 Tbit.

The data that is transmitted returns to the reception data input pin via the LIN transceiver. The received data and the transmitted data is compared bit by bit, and the results are stored in the BER flag in the RLN3nLEST register (see Section 25.7.6, Error Status).

In LIN mater mode, 1 Tbit is generated to be 16 fLIN, and thus the sampling point for received data is at the 13th clock cycle (81.25% position).

Figure 25.7 shows an example of data transmission timing.

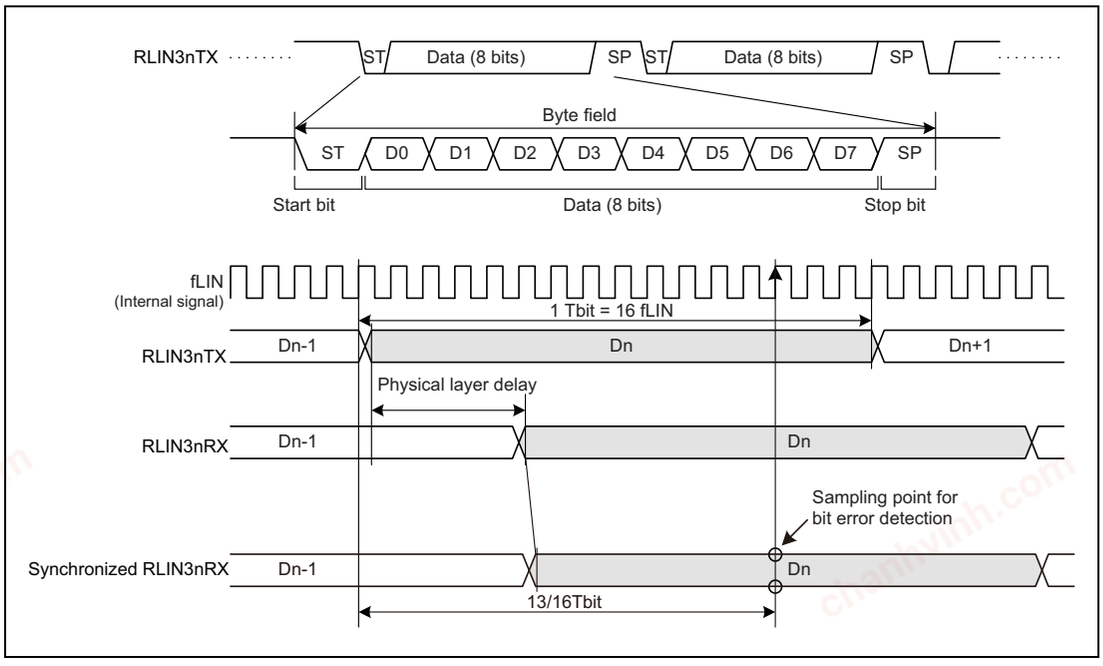


Figure 25.7 Example of Data Transmission Timing (LIN Master Mode)

chanhvinh.com

25.7.2.2 Data Reception

Data reception is performed by using the synchronized RLIN3nRX signal (an internal signal) that is the input from the RLIN3nRX pin synchronized with prescaler clock.

The byte field is synchronized at the falling edge of the start bit for the synchronized RLIN3nRX signal. After the falling edge is detected, sampling is performed again a specified period of time later, and the falling edge is recognized as a start bit if the synchronized RLIN3nRX signal is low level. The falling edge is not recognized as a start bit if the RLIN3nRX signal after the clearing of the resetting is low-level-fixed or if a high level is detected on re-sampling.

After the start bit is detected, the system samples 1 bit per Tbit.

The LIN Interface has a noise filter function with respect to reception data. If the LRDNFS bit in the RLN3nLMD register is 0, the LIN Interface uses a noise filter, and for a sampling value the value determined by a 3-sampling majority rule on prescaler clocks is used. If the LRDNFS bit in the RLN3nLMD register is 1, the LIN Interface does not use a noise filter, and for a sampling value the value of the synchronized RLIN3nRX value at the sampling position is used as is.

Figure 25.8 shows an example of data reception timing.

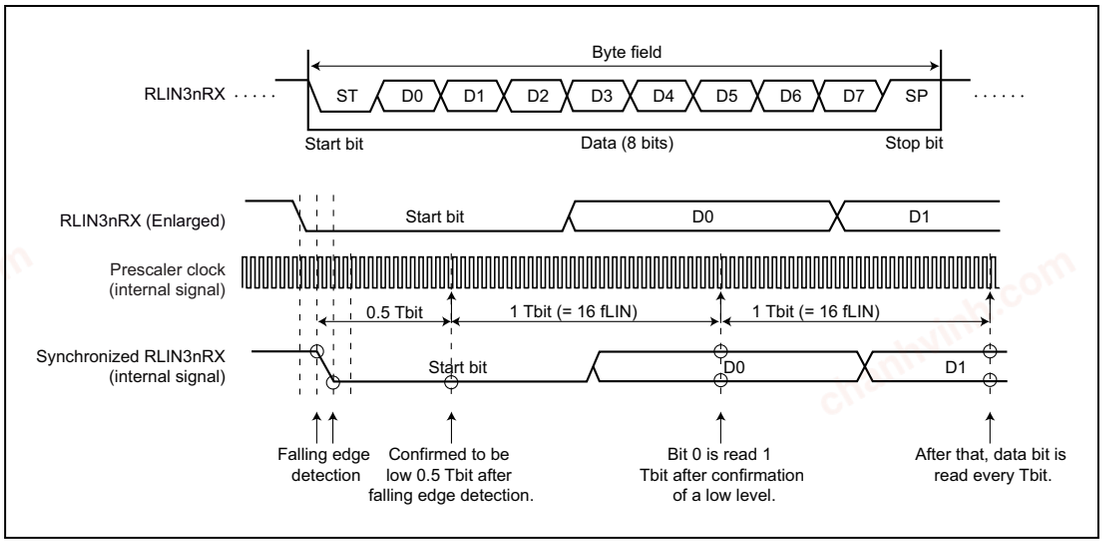


Figure 25.8 Example of Data Reception Timing

chanhvinh.com

25.7.3 Transmission/Reception Data Buffering

This section explains the buffer processing that takes place when the LIN Interface sends or receives data continuously.

25.7.3.1 Transmission of LIN Frames

For an 8-byte transmission, the contents stored in registers RLN3nLDBR1 to RLN3nLDBR8 are sequentially transmitted to data areas 1 to 8 of the LIN frame. In the case of a 4-bytes transmission, the contents stored in registers RLN3nLDBR1 to RLN3nLDBR4 are transmitted to data areas 1 to 4 of the LIN frame, but the contents of registers RLN3nLDBR5 to RLN3nLDBR8 are not transmitted. The transmitted checksum data is stored in the RLN3nLCBR register.

Figure 25.9 depicts the LIN transmission processing and the required buffer.

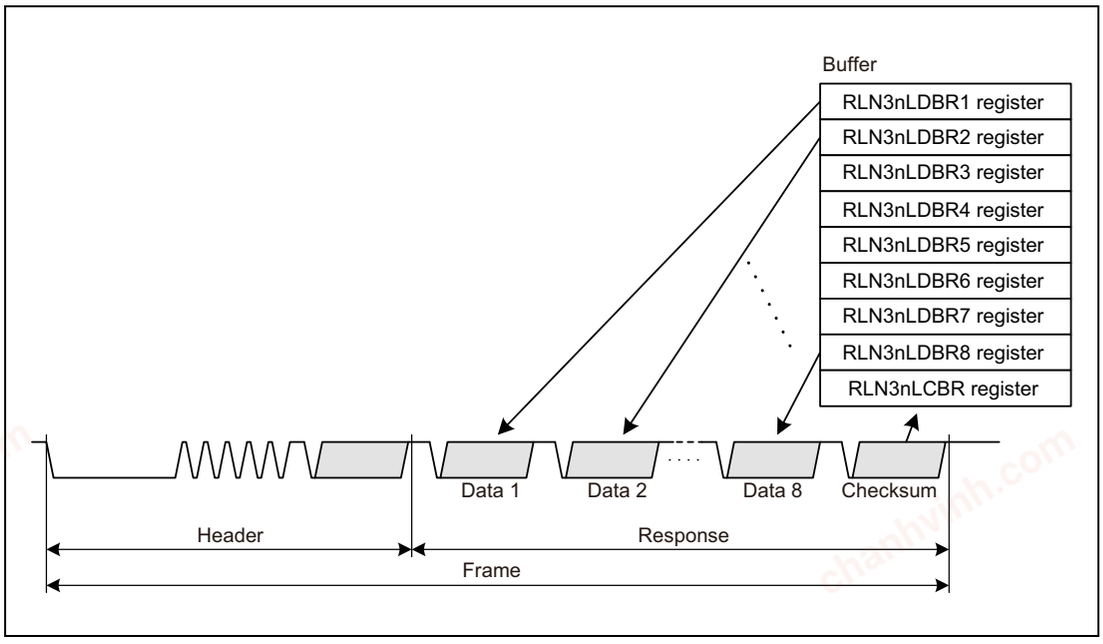


Figure 25.9 LIN Transmission Processing and Required Buffer

(1) Frame Separate Mode

Setting the FSM bit in the RLN3nLDFC register to 1 turns on the frame separate mode.

In frame separate mode, a header and a response are transmitted when prompted by separate transmission start requests.

When the transmission of a header is finished, the HTRC flag in the RLN3nLST register turns 1 (successful header transmission).

Use frame separate mode when sending or receiving response data of 9 bytes or greater in LIN master mode.

chanhvinh.com

25.7.3.2 Reception of LIN Frames

For an 8-byte reception, the contents of data areas 1 to 8 of the LIN frame is stored in registers RLN3nRLN3nLDBR1 to RLN3nLDBR8, respectively, upon receipt of a stop bit. In the case of a 4-byte reception, the contents of data areas 1 to 4 of the LIN frame are stored in registers RLN3nLDBR1 to RLN3nLDBR4, respectively; however, no data is stored in registers RLN3nLDBR5 to RLN3nLDBR8. Also, the received checksum data is stored in the RLN3nLCBR register.

Figure 25.10 depicts the LIN reception processing and the required buffer.

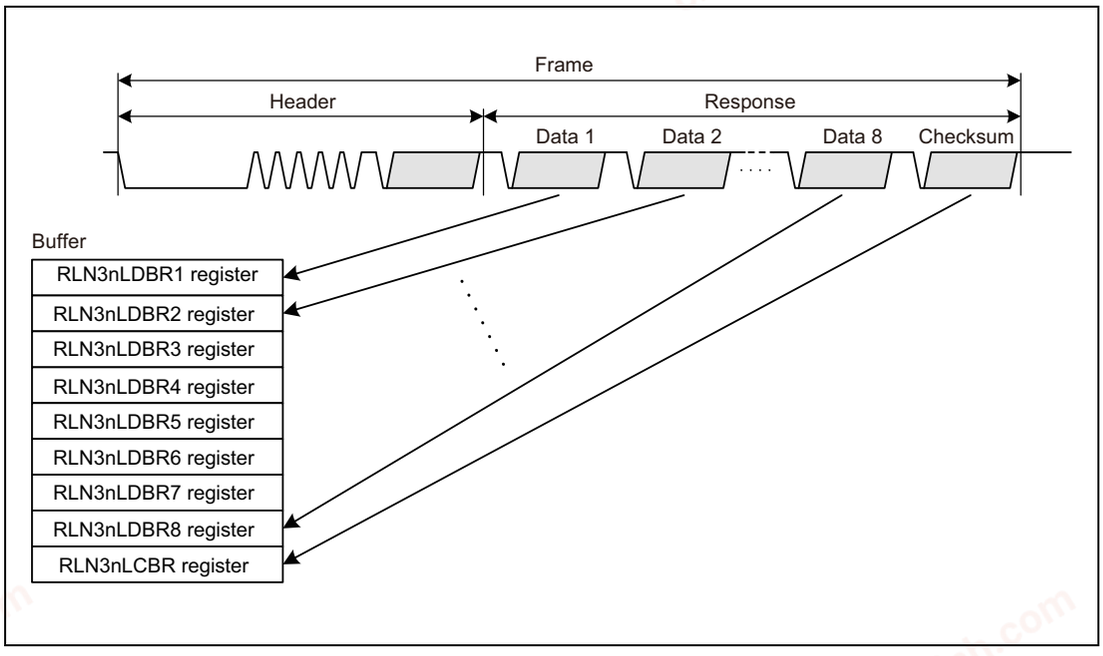


Figure 25.10 LIN Reception Processing and Required Buffer

(1) Reception of Data 1

When the reception of the first byte of data is finished, the D1RC flag in the RLN3nLST register turns 1 (successful data 1 reception).

chanhvinh.com

25.7.3.3 Multi-Byte Response Transmission/Reception Function

Normally in LIN communications, a response is 9 bytes or less including a checksum field; however, responses in 10 bytes or greater can also be sent and received.

In such a case, the bit error, framing error, response preparation error detection, and auto checksum functions are enabled.

If the data length is greater than 8 bytes, the LSS bit in RLN3nLDFC register should be set to 1 (indicating that the next data group to be sent or received is not the final data group) in the first data group (variable in 0 to 8 bytes) before sending or receiving the data group. After the transmission or reception, the user should determine whether the next data group is the final data group. If it is the final data group, the LSS bit should be set to 0 (indicating that the next data group to be sent or received is the final data group, and a checksum should be appended to the final data group.

By changing the RFDL bit in RLN3nLDFC register settings when the RTS bit in RLN3nLTRC register is 0, the user can change the data length for each data group.

When performing multi-byte response transmission/reception in LIN master mode, set the FSM bit in RLN3nLDFC register in the RLN3nLDFC register to 1 (frame separate mode).

chanhvinh.com

25.7.4 Wake-up Transmission/Reception

The wake-up transmission/reception can be used in LIN wake-up mode.

25.7.4.1 Wake-up Transmission

In LIN wake-up mode, setting the RCDS bit in the RLN3nLDFC register to 1 (transmission) and the FTS bit in the RLN3nLTRC register to 1 (header reception or wake-up transmission/reception started) causes a wake-up signal to be output from the output pin. The low level width of the wake-up signal should be set using the WUTL[3:0] bits in the RLN3nLWUP register.

However, if the LWBR0 bit of the RLN3nLWBR register is 1 (LIN2.x use), the LIN system clock (fLIN) becomes low level width at fa regardless of the setting of the LCKS bit of the RLN3nLMD register.

By setting the baud rate to 19200 bps while fa is selected and the WUTL[3:0] bits of the RLN3nLWUP register to 0100_B (5 Tbits), 260 µs low width can be output in LIN wake-up mode regardless of the setting of the LCKS bit of the RLN3nLMD register.

If a wake-up low is output without any bit error, the FTC flag in the RLN3nLST register turns 1 (successful frame response or wake-up transmission); when the FTCIE bit in the RLN3nLIE register is 1 (successful frame response/wakeup transmission interrupt enabled), an interrupt request is generated.

If RLN3nLEDE.BERE is set and a bit error is detected, wake-up transmission is canceled and the BER flag in the RLN3nLEST register is set to 1 (bit error detection).

When RLN3nLEDE.PBERE is set, set RLN3nLEST.PBER flag to 1 (physical bus error detection) at the same time of a bit error.

Figure 25.11 shows the wake-up transmission timing.

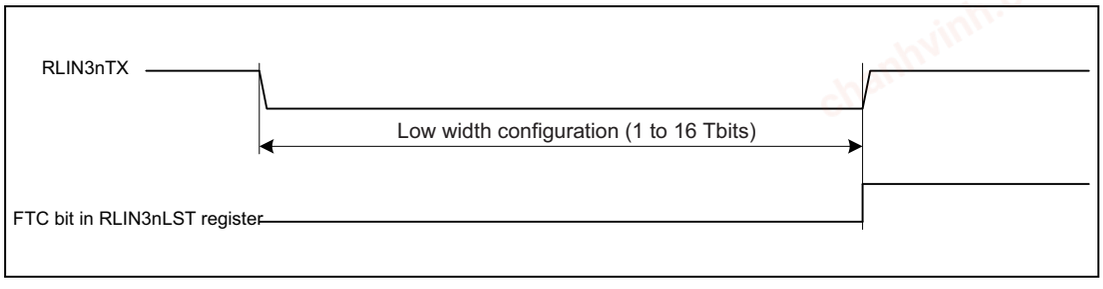


Figure 25.11 Wake-up Transmission Timing

chanhvinh.com

25.7.4.2 Wake-up Reception

The detection of a wake-up involves the use of an input signal low level width count function.

The input signal low level width count function measures the low width of the input signal to the RLIN3nTX pin, using the same sampling point as data reception. The function can measure the input signal low level width of 2.5 Tbits of fLIN or greater.

By setting the LWBR0 bit in the RLN3nLWBR register, operation is executed without changing the baud rate generator setting at a transition between LIN operation mode and LIN wake-up mode.

When LIN Specification Package Revision 1.3 is used, set the LWBR0 bit in the RLN3nLWBR register to 0. When LIN Specification Package Revision 2.x is used, set the LWBR0 bit to 1. Setting the LWBR0 bit to 1 selects the LIN system clock (fLIN) to fa regardless of the setting of the LCKS bit in the RLN3nLMD register. (The LCKS bit is not changed). By setting the baud rate to 19200 bps while fa is selected, the 130 µs or longer low-level width of the input signal to be measured regardless of the setting of the LCKS bit in the RLN3nLMD register.

When using the wake-up reception function, in LIN wake-up mode set the RFT bit in the RLN3nLDFC register to 0 (response reception), and then the FTS bit in the RLN3nLTRC register to 1 (frame transmission (header reception) or wake-up transmission/reception started).

When the low level width to be measured is reached, the FRC flag in the RLN3nLST register turns 1 (successful frame response/wake-up reception). If the FRCIE bit in the RLN3nLIE register is 1 (successful frame response or wake-up reception interrupt enabled), an interrupt request for successful LINn reception is generated.

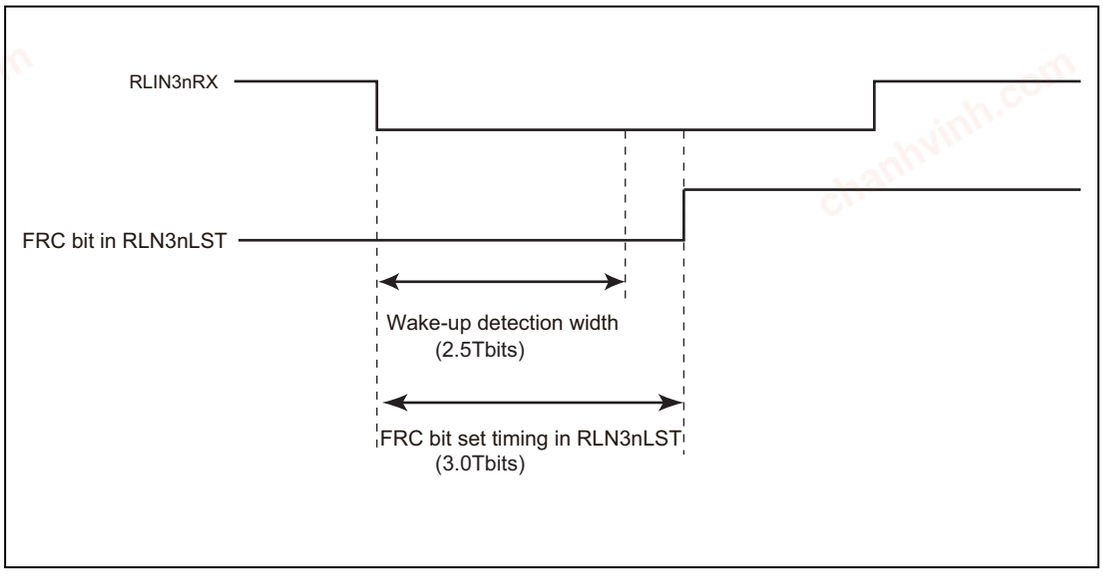


Figure 25.12 Input Signal Low level Count Function

25.7.4.3 Wakeup Collision

If the master node and the slave node transmit wakeup signals simultaneously, a collision will occur on the LIN bus, though a collision of wakeup signals is not detected in the LIN interface.

chanhvinh.com

25.7.5 Status

During LIN mode operation, the LIN Interface can detect seven types of statuses.

The four statuses, successful frame/wake-up transmission, successful frame/wake-up reception, error detection, successful header transmission/header reception, can generate interrupt requests.

Table 25.34 shows the types of statuses available in LIN master mode.

Table 25.34 Types of Statuses in LIN Master Mode

Status	Status set condition	Status clear condition	Operation mode capable of status detection	Corresponding bit	Interrupt
Reset	After the OM0 bit in the RLN3nLCUC register is set to LIN reset mode (OM0 = 0), if the LIN Interface enters LIN reset mode.	After the OM0 bit in the RLN3nLCUC register is set to 0 (LIN reset mode is canceled), if LIN reset mode is canceled.	All modes	OMM0 bit in RLN3nLMST register	—
Operation mode	After the OM1 bit in the RLN3nLCUC register is set to LIN operation mode, if actually the LIN Interface enters LIN operation mode.	After the OM1 bit in the RLN3nLCUC register is set to LIN wake-up mode, if actually the LIN Interface enters LIN wake-up mode.	• LIN operation mode • LIN wake-up mode	OMM1 bit in RLN3nLMST register	—
Frame/wake-up transmission end	When a frame (header transmission + response transmission), a wake-up signal, or a data group is transmitted successfully.	• When another communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software • After transition to LIN reset mode	• LIN operation mode • LIN wake-up mode	FTC flag in RLN3nLST register	√
Frame/wake-up reception end	When a frame (header transmission + response reception), a wake-up signal, or a data group is received successfully.	• When another communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software • After transition to LIN reset mode	• LIN operation mode • LIN wake-up mode	FRC flag in RLN3nLST register	√
Error detection	If any of the RPER flag, CSER flag, FER flag, FTER flag, PBER flag, and BER flags in the RLN3nLEST register turns 1 (error detected).	• When another communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software • After transition to LIN reset mode	• LIN operation mode • LIN wake-up mode	ERR flag in RLN3nLST register	√
Data 1 reception end	The RFT bit in the RLN3nLDFC register is 0 (reception) and the first byte of the response field or the first byte of each data group is received.*2	• When another communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software • After transition to LIN reset mode	LIN operation mode	D1RC flag in RLN3nLST register	—
Header transmission end	When a header field is transmitted successfully.	• When another communication is started • When cleared by software • After transition to LIN reset mode	LIN operation mode	HTRC flag in RLN3nLST register	√

Note 1. In LIN operation mode, the ERR flag in the RLN3nLST register is cleared to 0 by writing 0 to the RPER flag, CSER flag, FER flag, FER flag, FTER flag, PBER flag or BER flags in the RLN3nLEST register.

Note 2. Not detected when the RFDL [3:0] bits in the RLN3nLDFC register are 0000_B (0-byte + checksum).

chanhvinh.com

25.7.6 Error Status

25.7.6.1 LIN Master Mode

(1) Types of Error Statuses

The LIN Interface can detect six types of error statuses in LIN master mode. The condition of these error statuses can be checked by means of the corresponding bits in the RLN3nLEST register.

All error statuses represent interrupt events.

Table 25.35 shows the types of error statuses.

Table 25.35 Types of Error Statuses in LIN Master Mode

Status	Error detection condition	Operation mode capable of error detection	Communication	Enable/ disable detection	Corresponding bit
Bit error	The transmitted data and the data on the LIN bus monitored by the receive pin do not match *1*2	• LIN operation mode • LIN wake-up mode	Cancel	√	BER flag in RLN3nLEST register
Physical bus error	• LIN bus is detected to be high level when sending a break • LIN bus is detected to be low level when sending a break delimiter • LIN bus is detected to be high level when sending a wake-up	• LIN operation mode • LIN wake-up mode	Cancel	√	PBER flag in RLN3nLEST register
Timeout error	A frame or response transmission/reception does not terminate within a given time*3	LIN operation mode	Cancel	√	FTER flag in RLN3nLEST register
Framing error	In response field reception, a stop bit of each data byte is low level	LIN operation mode	Cancel	√	FER flag in RLN3nLEST register
Checksum error	In response field reception, the result of checksum test gives an error	LIN operation mode	—	×	CSER flag in RLN3nLEST register
Response preparation error	One of the following conditions occurs in frame separate mode during a multi-byte response reception: • The first reception data byte is received after completion of header transmission but before a response transmission/reception request is set • The first reception data byte is received after the completion of previous data group reception before a transmission/reception request for another data group is sett	LIN operation mode	Cancel	×	RPER flag in RLN3nLEST register

Note 1. If a bit error is detected, the process is canceled after a stop bit is sent. If a bit error is detected in a non-data area, such as an inter-byte space, the transmission is canceled immediately after that area. If a bit error is detected during the transmission of a wake-up, the transmission of the wake-up is canceled after the error-causing bit is sent.

Note 2. In a multi-byte response transmission, bit errors are detected also between data groups.

Note 3. The timeout time depends on the response field data length (the RFDL [3:0] bits in the RLN3nLDFC register) and the checksum selection (the CSM bit in the RLN3nLDFC register), and this can be calculated according to the following formula.
When the setting of the FSM bit in the RLN3nLDFC register is 1 (i.e., frame separation mode), the timeout time is that for eight bytes until the RTS bit of the RLN3nLTRC register is set. Once the RTS bit is set, the timeout time is re-set to the time based on the response field data length (the RFDL[3:0] bits in the RLN3nLDFC register).
[Frame timeout]
On classic selection (when the CSM bit in RLN3nLDFC is 0):
Timeout time = 49 + (number of data bytes + 1) × 14 [Tbit]

chanhvinh.com

On enhanced selection (when the CSM bit in RLN3nLDFC is 1):
Timeout time = 48 + (number of data bytes + 1) × 14 [Tbit]
The aforementioned timeout time is a time greater than the TFRAME_MAX of LIN Specification Package Revision 1.3 on classic selection, or the TFRAME_MAX of LIN Specification Package Revision 2.x on enhanced selection.
[Response timeout]
Timeout time = (number of data bytes + 1) × 14 [Tbit]
When an error is detected, time-out error detection function stops.

The error status is cleared when the next communication is started (when the FTS bit in the RLN3nLTRC register is set), by software, or at a transition to LIN reset mode.

(2) Target Time Area for LIN Error Detection

Figure 25.13 shows the time domain in which the LIN Interface in master mode performs monitoring for error detection.

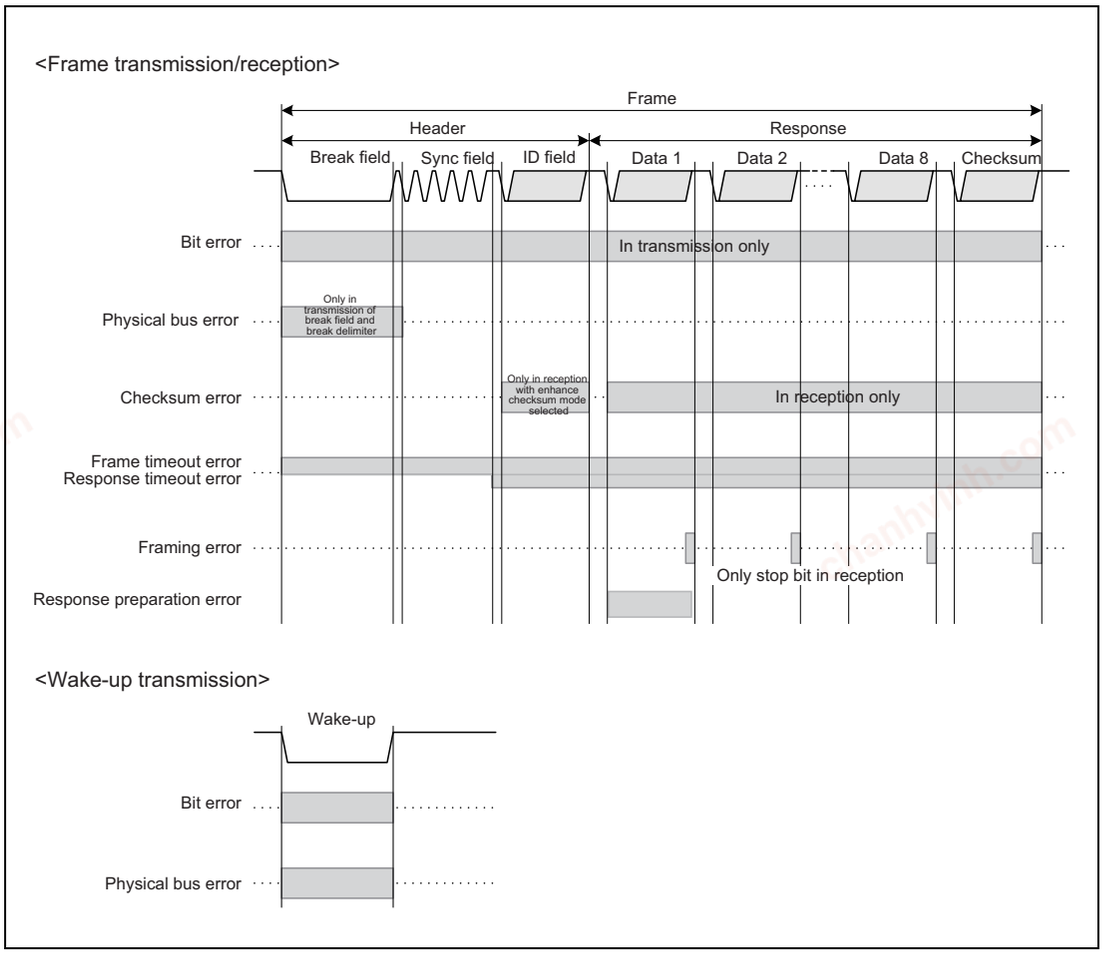


Figure 25.13 Target Time Area for LIN Error Detection (LIN Master Mode)

chanhvinh.com

25.8 LIN Self-Test Mode

When the LIN interface enters the LIN self-test mode, RLIN3nTX and RLIN3nRX are disconnected from external pins and RLIN3nTX and RLIN3nRX are connected in the LIN interface. Therefore, the frame transmitted from RLIN3nTX is looped back to RLIN3nRX.

The LIN self-test mode can perform tests exclusively in LIN mode.

The self-test can be performed in the following two types.

- LIN master self-test mode (transmission): Header transmission and response transmission
- LIN master self-test mode (reception): Header transmission and response reception

In LIN self-test mode, the operate is at the fastest baud rate, regardless of the setting of the baud rate generator,

Regardless of the setting of the baud rate related registers, the baud rate operates at the LIN communication clock source/16 [bps]. (The NSPB bits in the RLN3nLWBR register should be set to 0000_B or 1111_B.)

(The LPRS bits in the RLN3nLWBR register should be set to 000_B.)

In addition, in LIN self-mode, the following functions are not supported.

- LIN wake-up mode
- Frame separate mode
- Multi-byte response transmission/reception
- Frame/response timeout error

chanhvinh.com

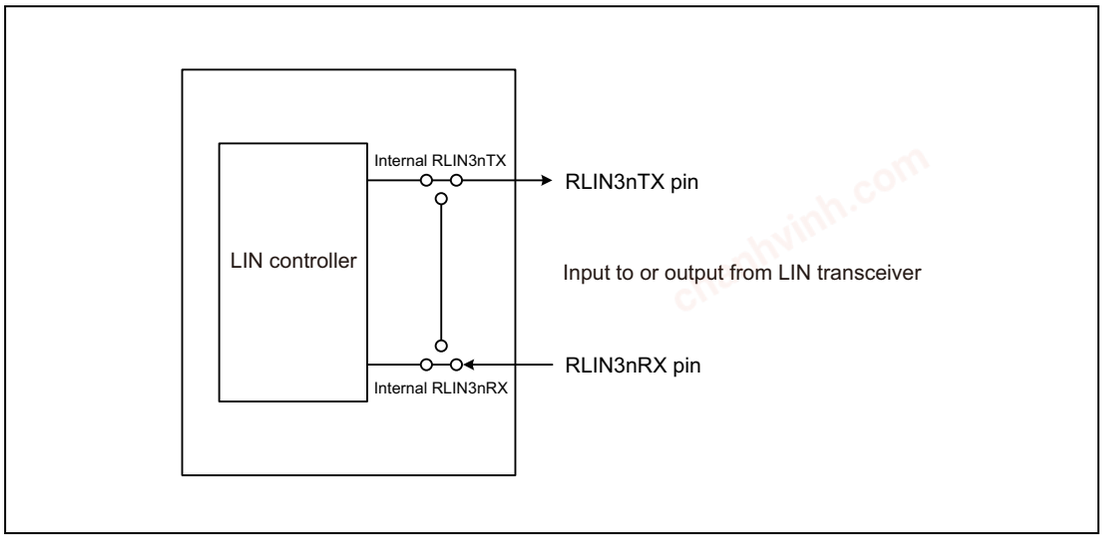


Figure 25.14 Connection in LIN Reset Mode and LIN Mode

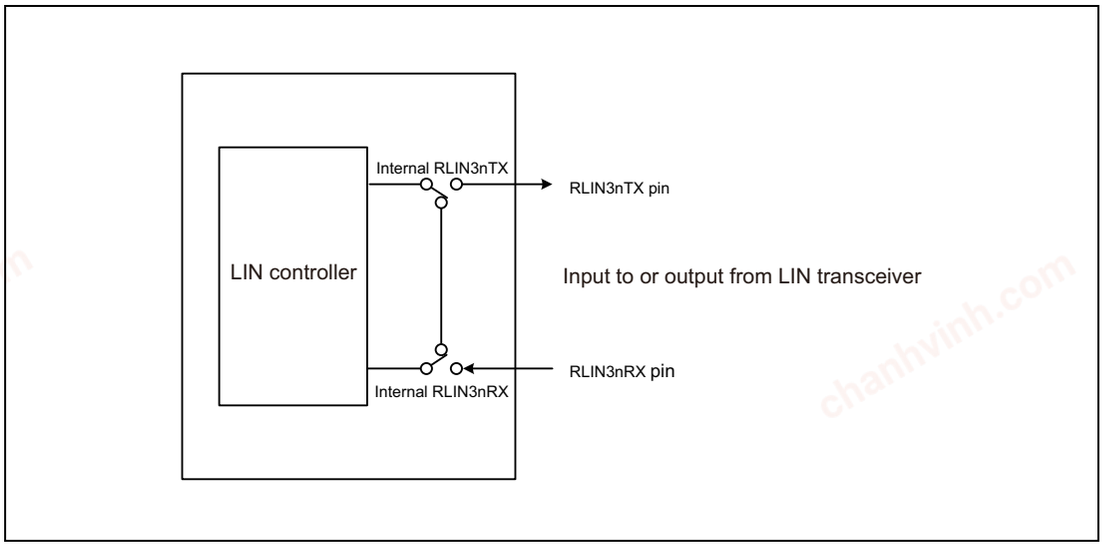


Figure 25.15 Connection in LIN Self-Test Mode

chanhvinh.com

25.8.1 Change to LIN Self-Test Mode

Writing to the RLN3nLSTC register makes a transition to the LIN self-test mode.
When the LSTM bit in the RLN3nLSTC register is set to 1, the shift to the LIN self-test mode is checked.

When changing to LIN self-test mode, be sure to execute a specific sequence. In that sequence, information must be written three times consecutively to the LIN self-test control register, as follows:

- Change to LIN reset mode
  Set the OM0 bit in the RLN3nLCUC register to 0 (LIN reset mode).
  Read the OMM0 bit in the RLN3nLMST register; verify that it is 0 (LIN reset mode).
- Select a LIN mode
  LMD bits in RLN3nLMD = 00_B (LIN master mode)
- 1st write: RLN3nLSTC register = $1010\ 0111_B$ ($A7_H$)
- 2nd write: RLN3nLSTC register = $0101\ 1000_B$ (58_H)
- 3rd write: RLN3nLSTC register = $0000\ 0001_B$ (01_H)
- Verify the transition to LIN self-test mode
  Read the LSTM bit in the RLN3nLSTC register; verify that it is 1 (LIN self-test mode).

If the key of the first write ($A7_H$) is written twice by mistake, the transition to LIN self-test mode is canceled. The above sequence should be retried from the step of first write. In addition, if a write to another LIN-related register is performed during transition to LIN self-test mode (three consecutive write operations to the RLN3nLSTC register), the transition is also canceled.

chanhvinh.com

25.8.2 Transmission in LIN Master Self-Test Mode

To execute a self-test on LIN master transmission, perform the procedure below:

- Set the baud rate, noise filter, and interrupt output related registers.
  RLN3nLWBR register = $0000000x_B$
  RLN3nLBRP0 register = $xxxxxxxx_B$*1
  RLN3nLBRP1 register = $xxxxxxxx_B$*1
  RLN3nLMD register = $00xxxx00_B$*1
- Set the interrupt enable and error enable related registers.
  RLN3nLIE register = $0000xxxx_B$*2
  RLN3nLEDE register = $x000x0xx_B$
- Set the break field and space related registers.
  RLN3nLBFC register = $00xxxxxx_B$
  RLN3nLSC register = $00xx0xxx_B$
- Cancel the LIN reset mode.
  Write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register, and check that the OMM1 and OMM0 bits in the RLN3nLMST register are 11_B.
- Set the transmit frame related registers.
  RLN3nLDFC register = $00x1xxxx_B$
  RLN3nLIDB register = $xxxxxxxx_B$
  RLN3nLDRB1 to RLN3nLDRB8 registers = $xxxxxxxx_B$
- Header transmission → response transmission started
  Set the FTS bit in the RLN3nLTRC register to 1 (frame transmission or wake-up transmission/ reception started).
  The LIN master self-test mode (transmission) is executed. In this mode, interrupts are generated, and status and error status are also updated. The checksum is automatically calculated by the LIN interface. To suspend the LIN master self-test mode (transmission) being executed, write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register for transition to LIN reset mode.
- When the transmission is completed, the reversed value of the looped-back frame data is stored in the RLN3nLIDB, RLN3nLDBRm (m = 1 to 8), and RLN3nLCBR registers (the data is reversed before being stored because the transmitted value should be compared with the looped-back value). Then, the FTS bit in the RLN3nLTRC register is cleared.
- If the transmission fails to complete due to an error, the applicable error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

Note: x: Don't care

Note 1. The following register settings are not reflected to the operation of the LIN self-test mode: the RLN3nLBRP0 register, the RLN3nLBRP1 register, and the LCKS bit in the RLN3nLMD register. Therefore, those settings are not necessary.

Note 2. When the successful header transmission interrupt and the successful frame transmission interrupt are used in the same interrupt processing, if the software processing of the successful header transmission interrupt is not completed before the generation of the successful frame transmission interrupt, the SHIE bit in the RLN3nLIE register should not be set to 1 (successful header transmission interrupt enabled). The time required from the set of the successful header transmission flag to the set of the successful frame/wake-up transmission flag is calculated by the following formula.

chanhvinh.com

10 × (number of data bytes + 1) [Tbit]
1 Tbit = LIN communication clock source × 16

25.8.3 Reception in LIN Master Self-Test Mode

To execute a self-test on LIN master reception, perform the procedure below:

- Set the baud rate, noise filter, and interrupt output related registers.
  RLN3nLWBR register = $0000000x_B$
  RLN3nLBRP0 register = $xxxxxxxx_B$*1
  RLN3nLBRP1 register = $xxxxxxxx_B$*1
  RLN3nLMD register = $00xxxx00_B$*1
- Set the interrupt enable and error enable related registers.
  RLN3nLIE register = $0000xxxx_B$*2
  RLN3nLEDE register = $x000x0xx_B$
- Set the break field and space related registers.
  RLN3nLBFC register = $00xxxxxx_B$
  RLN3nLSC register = $00xx0xxx_B$*1
- Cancel the LIN reset mode.
  Write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register, and check that the OMM1 and OMM0 bits in the RLN3nLMST register are 11_B.
- Set the reception frame related registers.
  RLN3nLDFC register = $00x0xxxx_B$*3
  RLN3nLIDB register = $xxxxxxxx_B$
  RLN3nLDBR1 to RLN3nLDBR8 registers = $xxxxxxxx_B$
  RLN3nCBR register = $xxxxxxxx_B$
  Since the checksum value to be transmitted is not automatically calculated, set the calculation value to the RLN3nLCBR register. If an incorrect checksum is set at this time, the checksum error can be tested.
- Header transmission → response reception started
  Set the FTS bit in the RLN3nLTRC register to 1 (frame transmission or wake-up transmission/ reception started).
  The LIN master self-test mode (reception) is executed. In this mode, interrupts are generated, and status and error status are also updated. To suspend the LIN master self-test mode (reception) being executed, write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register for transition to LIN reset mode.
- When the reception is completed, the reversed value of the looped-back frame data is stored in the RLN3nLIDB, RLN3nLDBRm (m = 1 to 8), and RLN3nLCBR registers (the data is reversed before being stored because the set value should be compared with the looped-back value). Then, the FTS bit in the RLN3nLTRC register is cleared.
- If the reception fails to complete due to an error, the applicable error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

Note: x: Don't care

Note 1. The following register settings are not reflected to the operation of the LIN self-test mode: the RLN3nLBRP0 register, the RLN3nLBRP1 register, the LCKS bit in the RLN3nLMD register, and the IBS bit in the RLN3nLSC register. Therefore, those settings are not necessary.

chanhvinh.com

Note 2. When the successful header transmission interrupt and the successful frame reception interrupt are used in the same interrupt processing, if the software processing of the successful header transmission interrupt is not completed before the generation of the successful frame reception interrupt, the SHIE bit in the RLN3nLIE register should not be set to 1 (successful header transmission interrupt enabled).
The time required from the set of the successful header transmission flag to the set of the successful frame/wake-up reception flag is calculated by the following formula.

10 × (number of data bytes + 1) [Tbit]

1 Tbit = LIN communication clock source × 16

Note 3. When the reception is in self-test mode, be sure to set the response field length (RFDL bits) to at least 1 byte.

25.8.4 Terminating LIN Self-Test Mode

To terminate LIN self-test mode, perform the procedure below:

- Write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register.
  If the OMM1 and OMM0 bits in the RLN3nLMST register are not 11_B, write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register. After confirming that the OMM1 and OMM0 bits in the RLN3nLMST register have turned 11_B, change to LIN reset mode.
- Verify the cancelation of LIN self-test mode.
  Read the LSTM bit in the RLN3nLSTC register; confirm that it is not 0 (not in LIN self-test)
- Verify the transition to LIN reset mode.
  Read the OMM0 bit in the RLN3nLMST register; verify that it is 0 (LIN reset mode).

chanhvinh.com

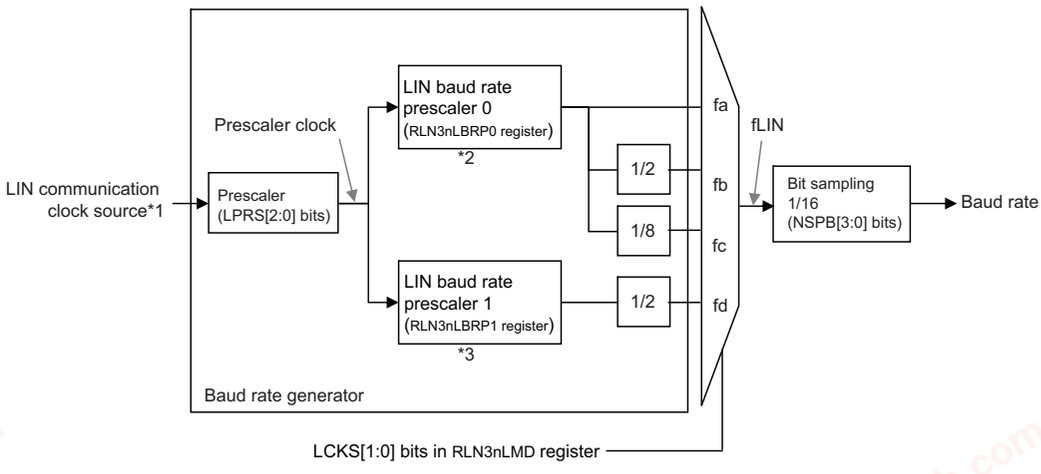
25.9 Baud Rate Generator

The prescaler clock is obtained by frequency-dividing the LIN communication clock source by the prescaler, and the LIN system clock (fLIN) is obtained by frequency-dividing the prescaler clock by the baud rate generator. The clock obtained by frequency-dividing the LIN system clock (fLIN) by the number of samples is the baud rate. The reciprocal of this baud rate is called the bit time (Tbit).

The LIN Interface has two kinds of baud rate generators. The baud rate generators switch over according to the mode used.

25.9.1 LIN Master Mode

Figure 25.16 shows a block diagram of baud rate generation in LIN master mode.



Note 1. For the LIN communication clock source, refer to **section 6, Clock Pulse Generator**.
Note 2. When the value in RLN3nLBRP0 register is N (N = 0 to 255), the clock frequency is divided by N + 1.
Note 3. When the value in RLN3nLBRP1 register is M (M = 0 to 255), the clock frequency is divided by M + 1.

Figure 25.16 Block Diagram of Baud Rate Generation in LIN Master Mode

By setting the RLN3nLBRP0 register so that fa is 307200 Hz (= 19200 × 16), the resulting bit rates are fa = 19200 × 16, fb = 9600 × 16 and fc = 2400 × 16. These bit rates are frequency-divided by 16 in the bit timing generator, enabling bit rates of 19200 bps, 9600 bps and 2400 bps to be generated. Also, by setting the RLN3nLBRP1 register so that fd is 166672 Hz (= 10417 × 16), the resulting bit rate is fd = 10417 × 16. This bit rate is frequency-divided by 16 in the bit timing generator, enabling 10417 bps to be generated.

The equation for calculating the baud rate is given below.

Baud rate of LIN master

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selection clock) ÷ (RLN3nLBRP0 + 1) ÷ 16 [bps] (When fa is selected for fLIN)

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selection clock) ÷ (RLN3nLBRP0 + 1) ÷ 2 ÷ 16 [bps] (When fb is selected for fLIN)

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selection clock) ÷ (RLN3nLBRP0 + 1) ÷ 8 ÷ 16 [bps] (When fc is selected for fLIN)

chanhvinh.com

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selection clock) ÷ (RLN3nLBRP1 + 1) ÷ 2 ÷ 16 [bps] (When fd is selected for fLIN)

25.9.2 Noise Filter

The LIN Interface has a noise filter for reducing erroneous receiving of data due to noise. By setting the LRDNFS bit in the RLN3nLMD register to 0 (to use the noise filter), the noise filter is activated. The noise filter samples the level of the synchronized RLIN3nRX with the prescaler clock, and outputs the sampling value determined by a 3-sampling majority rule. The value of each bit of the receive data is determined based on the noise filter output.

Figure 25.17 shows the configuration of the noise filter, Figure 25.18 an example of a noise filter circuit, and Figure 25.19 the determination of the received data when the noise filter is used.

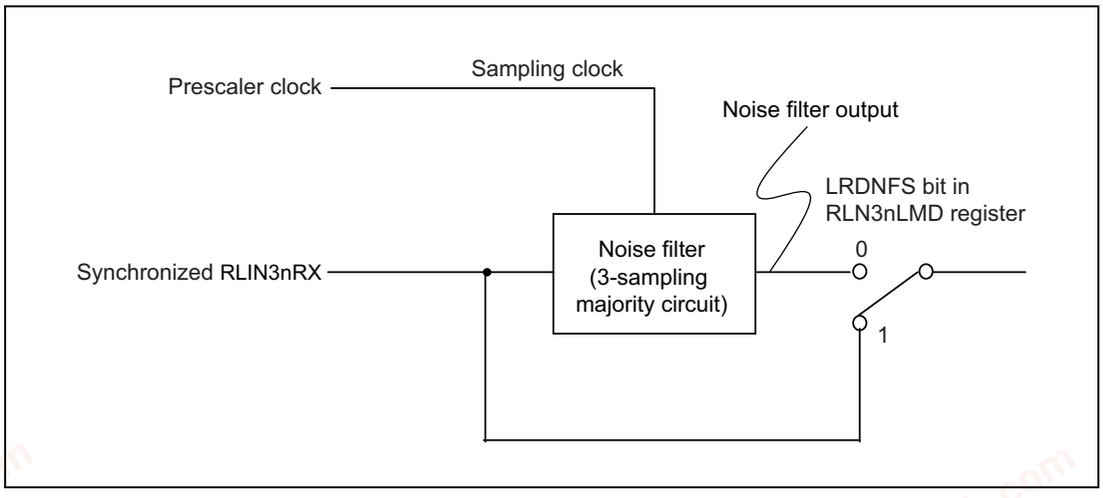


Figure 25.17 Configuration of Noise Filter

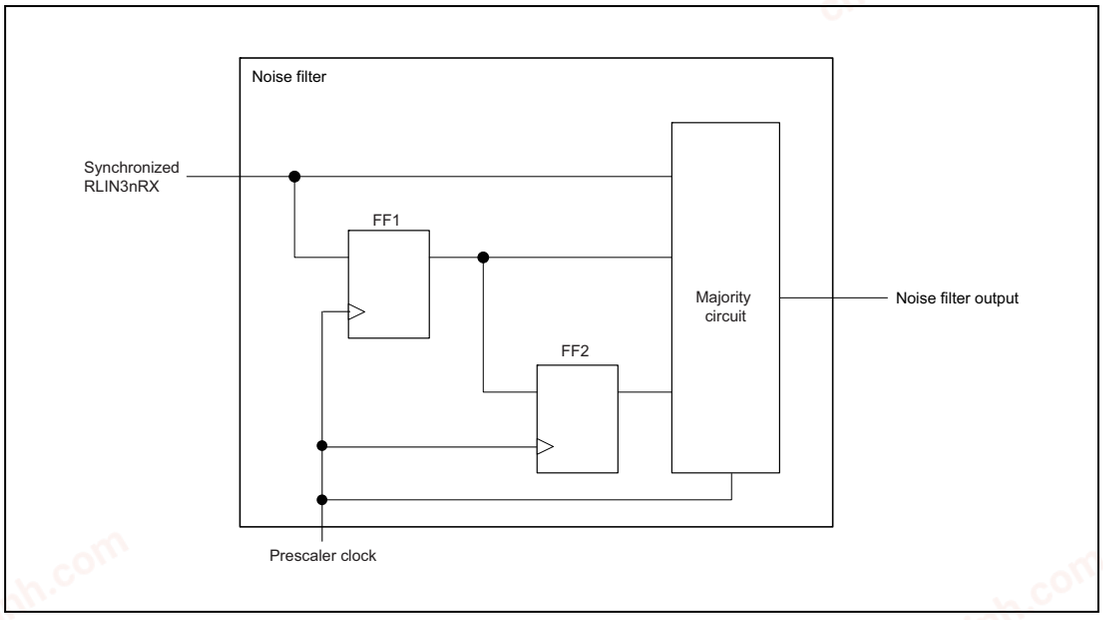


Figure 25.18 Example of Noise Filter Circuit

chanhvinh.com

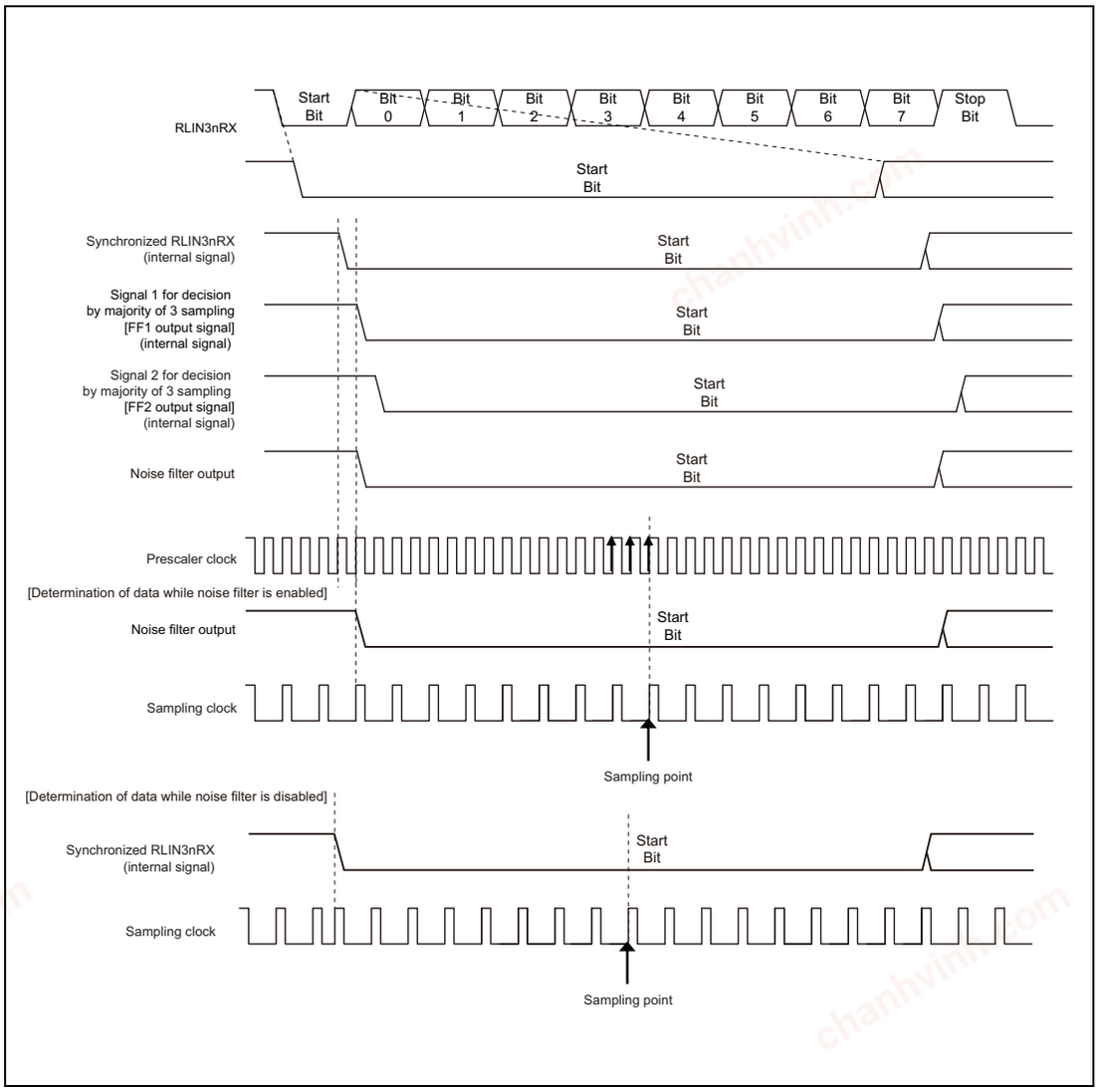


Figure 25.19 Determination of Received Data when Noise Filter is Used

chanhvinh.com

26. Ethernet Controller

This LSI has an on-chip Ethernet controller (ETHER) conforming to the Ethernet or the IEEE802.3 MAC (Media Access Control) layer standard. Connecting a physical-layer LSI (PHY-LSI) complying with this standard enables the ETHER to perform transmission and reception of Ethernet/IEEE802.3 frames. The Ethernet controller in this LSI has one MAC layer interface port, which can be made to perform transmission and reception independently.
The ETHER can transfer the transmitted or received Ethernet frame data to and from the transmit/receive buffer in the memory at high speed using a dedicated direct memory access controller (E-DMAC).

26.1 Features

- MAC (Media Access Control) function
  Constructs/deconstructs data frames (frame format conforming to IEEE802.3, 2000 Edition)
  Supports transfer at 10 and 100 Mbps
  Supports full-duplex mode
  One channel (ETHER0)
  Flow control conforming to IEEE802.3x
  Supports one PHY interface conforming to IEEE802.3
  —MII (Media Independent Interface)
  Upward protocol support (checksum) function
- E-DMAC (Direct Memory Access Controller for Ethernet controller) function
  Data transfer between ETHER and external/internal memory
  One channel
  32-byte burst transfer
  Supports single-frame/single-descriptor operation and single-frame/multi-descriptor (multi-buffer) operation
  Transfer data width: 32 bits
  Transmit/receive FIFO (for transmission: 2 Kbytes, for reception: 4 Kbytes)
  Function for calculating the intelligent checksum value

chanhvinh.com

Figure 26.1 shows the configuration of the ETHER.

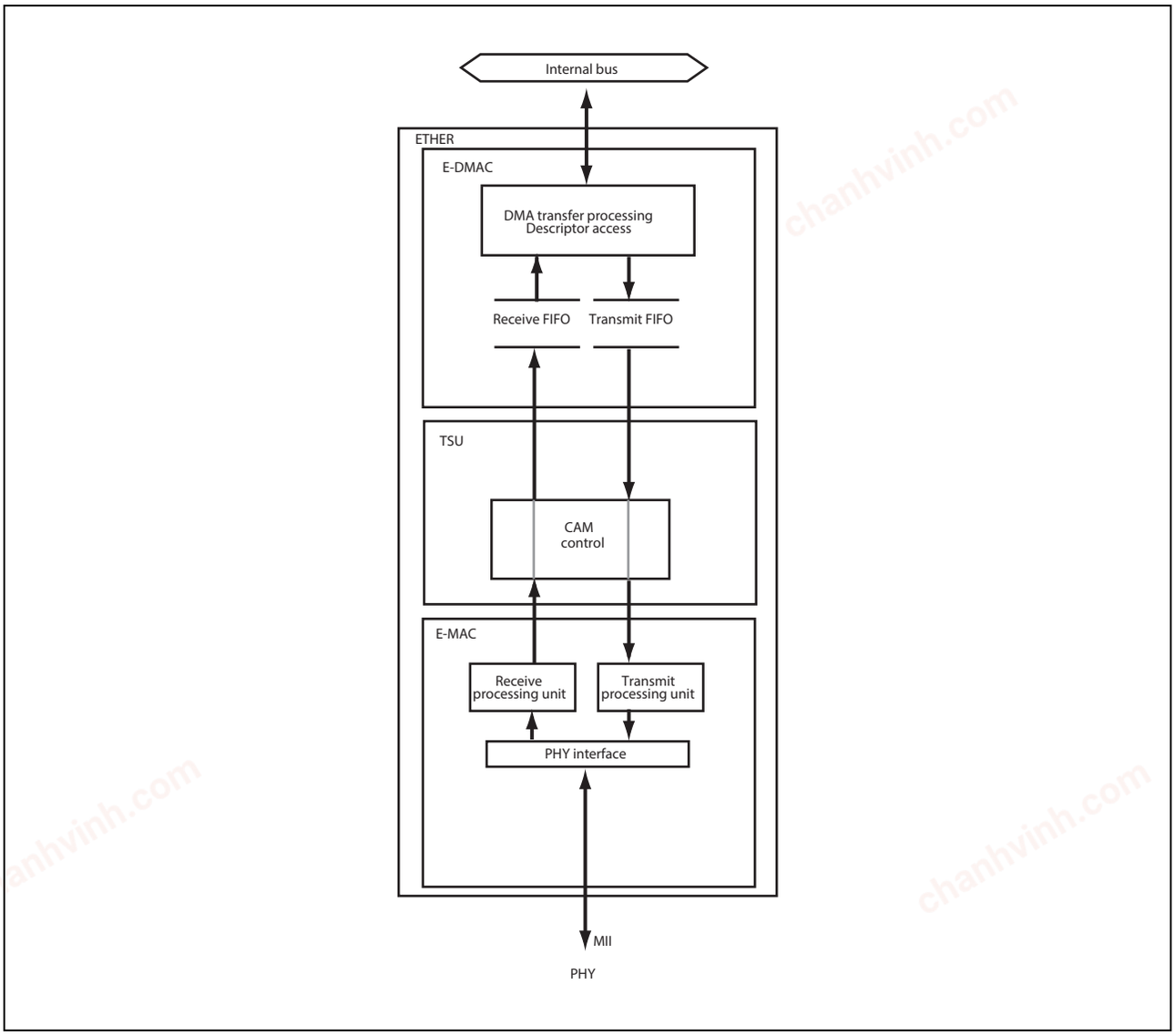


Figure 26.1 Configuration of ETHER

chanhvinh.com

26.2 Input/Output Pins

Table 26.1 lists the pin configuration of the ETHER.

Table 26.1 Pin Configuration

Name	Abbreviation	I/O	Function
Transmit clock	ET_TXCLK*1	Input	ET_TXEN, ET_TXD[3:0] timing reference signal
Transmit enable	ET_TXEN*1	Output	Indicates that transmit data is ready on ET_TXD[3:0]
MII transmit data	ET_TXD[3:0]*1	Output	MII transmit data
Collision detection	ET_COL*1	Input	Collision detection signal
Transmit error	ET_TXER*1	Output	Not asserted in the ETHER
Receive clock	ET_RXCLK*1	Input	ET_RXDV, ET_RXD[3:0], ET_RXER timing reference signal
Receive data valid	ET_RXDV*1	Input	Indicates that valid receive data is on ET_RXD[3:0]
MII receive data	ET_RXD[3:0]*1	Input	MII receive data
Receive error	ET_RXER*1	Input	Identifies error state occurred during data reception
Carrier detection	ET_CRS*1	Input	Carrier detection signal
Management data clock	ET_MDC*1	Output	Reference clock signal for information transfer via ET_MDIO
Management data I/O	ET_MDIO*1	I/O	Bidirectional signal for exchange of management information between STA and PHY

Note 1. MII signal conforming to IEEE802.3u

chanhvinh.com

26.3 Register Descriptions

Table 26.2 shows the configuration of registers of the ETHER.

Table 26.2 Register Configuration

Name	Abbreviation	R/W	Address	Access Size
Software reset register	ARSTR	R/W	H'E820 4800	32
E-MAC mode register	ECMR0	R/W	H'E820 3500	32
E-MAC status register	ECSR0	R/W	H'E820 3510	32
E-MAC interrupt permission register	ECSIPR0	R/W	H'E820 3518	32
PHY interface register	PIR0	R/W	H'E820 3520	32
MAC address high register	MAHR0	R/W	H'E820 35C0	32
MAC address low register	MALR0	R/W	H'E820 35C8	32
Receive frame length register	RFLR0	R/W	H'E820 3508	32
CRC error frame receive counter register	CEFCR0	R/W	H'E820 3740	32
Frame receive error counter register	FRECR0	R/W	H'E820 3748	32
Too-short frame receive counter register	TSFRCR0	R/W	H'E820 3750	32
Too-long frame receive counter register	TLFRCR0	R/W	H'E820 3758	32
Residual-bit frame receive counter register	RFCR0	R/W	H'E820 3760	32
Multicast address frame receive counter register	MAFCR0	R/W	H'E820 3778	32
Automatic PAUSE frame register	APR0	R/W	H'E820 3554	32
Manual PAUSE frame register	MPR0	R/W	H'E820 3558	32
Automatic PAUSE frame retransmit count register	TPAUSER0	R/W	H'E820 3564	32
PAUSE frame transmit counter register	PFTCR0	R	H'E820 355C	32
PAUSE frame receive counter register	PFRCR0	R	H'E820 3560	32
TSU counter reset register	TSU_CTRST	R/W	H'E820 4804	32
CAM entry table specification enable register (common)	TSU_FWSLC	R/W	H'E820 4838	32
VLANtag set register	TSU_VTAG0	R/W	H'E820 4858	32
CAM entry table busy register	TSU_ADSBSY	R	H'E820 4860	32
CAM entry table enable register	TSU_TEN	R/W	H'E820 4864	32
CAM entry table POST 1 register	TSU_POST1	R/W	H'E820 4870	32
CAM entry table POST 2 register	TSU_POST2	R/W	H'E820 4874	32
CAM entry table POST 3 register	TSU_POST3	R/W	H'E820 4878	32
CAM entry table POST 4 register	TSU_POST4	R/W	H'E820 487C	32
CAM entry table 0H register	TSU_ADRH0	R/W	H'E820 4900	32
CAM entry table 1H register	TSU_ADRH1	R/W	H'E820 4908	32
CAM entry table 2H register	TSU_ADRH2	R/W	H'E820 4910	32
CAM entry table 3H register	TSU_ADRH3	R/W	H'E820 4918	32
CAM entry table 4H register	TSU_ADRH4	R/W	H'E820 4920	32
CAM entry table 5H register	TSU_ADRH5	R/W	H'E820 4928	32
CAM entry table 6H register	TSU_ADRH6	R/W	H'E820 4930	32
CAM entry table 7H register	TSU_ADRH7	R/W	H'E820 4938	32
CAM entry table 8H register	TSU_ADRH8	R/W	H'E820 4940	32
CAM entry table 9H register	TSU_ADRH9	R/W	H'E820 4948	32
CAM entry table 10H register	TSU_ADRH10	R/W	H'E820 4950	32
CAM entry table 11H register	TSU_ADRH11	R/W	H'E820 4958	32
CAM entry table 12H register	TSU_ADRH12	R/W	H'E820 4960	32

chanhvinh.com

Table 26.2 Register Configuration

Name	Abbreviation	R/W	Address	Access Size
CAM entry table 13H register	TSU_ADRH13	R/W	H'E820 4968	32
CAM entry table 14H register	TSU_ADRH14	R/W	H'E820 4970	32
CAM entry table 15H register	TSU_ADRH15	R/W	H'E820 4978	32
CAM entry table 16H register	TSU_ADRH16	R/W	H'E820 4980	32
CAM entry table 17H register	TSU_ADRH17	R/W	H'E820 4988	32
CAM entry table 18H register	TSU_ADRH18	R/W	H'E820 4990	32
CAM entry table 19H register	TSU_ADRH19	R/W	H'E820 4998	32
CAM entry table 20H register	TSU_ADRH20	R/W	H'E820 49A0	32
CAM entry table 21H register	TSU_ADRH21	R/W	H'E820 49A8	32
CAM entry table 22H register	TSU_ADRH22	R/W	H'E820 49B0	32
CAM entry table 23H register	TSU_ADRH23	R/W	H'E820 49B8	32
CAM entry table 24H register	TSU_ADRH24	R/W	H'E820 49C0	32
CAM entry table 25H register	TSU_ADRH25	R/W	H'E820 49C8	32
CAM entry table 26H register	TSU_ADRH26	R/W	H'E820 49D0	32
CAM entry table 27H register	TSU_ADRH27	R/W	H'E820 49D8	32
CAM entry table 28H register	TSU_ADRH28	R/W	H'E820 49E0	32
CAM entry table 29H register	TSU_ADRH29	R/W	H'E820 49E8	32
CAM entry table 30H register	TSU_ADRH30	R/W	H'E820 49F0	32
CAM entry table 31H register	TSU_ADRH31	R/W	H'E820 49F8	32
CAM entry table 0L register	TSU_ADRL0	R/W	H'E820 4904	32
CAM entry table 1L register	TSU_ADRL1	R/W	H'E820 490C	32
CAM entry table 2L register	TSU_ADRL2	R/W	H'E820 4914	32
CAM entry table 3L register	TSU_ADRL3	R/W	H'E820 491C	32
CAM entry table 4L register	TSU_ADRL4	R/W	H'E820 4924	32
CAM entry table 5L register	TSU_ADRL5	R/W	H'E820 492C	32
CAM entry table 6L register	TSU_ADRL6	R/W	H'E820 4934	32
CAM entry table 7L register	TSU_ADRL7	R/W	H'E820 493C	32
CAM entry table 8L register	TSU_ADRL8	R/W	H'E820 4944	32
CAM entry table 9L register	TSU_ADRL9	R/W	H'E820 494C	32
CAM entry table 10L register	TSU_ADRL10	R/W	H'E820 4954	32
CAM entry table 11L register	TSU_ADRL11	R/W	H'E820 495C	32
CAM entry table 12L register	TSU_ADRL12	R/W	H'E820 4964	32
CAM entry table 13L register	TSU_ADRL13	R/W	H'E820 496C	32
CAM entry table 14L register	TSU_ADRL14	R/W	H'E820 4974	32
CAM entry table 15L register	TSU_ADRL15	R/W	H'E820 497C	32
CAM entry table 16L register	TSU_ADRL16	R/W	H'E820 4984	32
CAM entry table 17L register	TSU_ADRL17	R/W	H'E820 498C	32
CAM entry table 18L register	TSU_ADRL18	R/W	H'E820 4994	32
CAM entry table 19L register	TSU_ADRL19	R/W	H'E820 499C	32
CAM entry table 20L register	TSU_ADRL20	R/W	H'E820 49A4	32
CAM entry table 21L register	TSU_ADRL21	R/W	H'E820 49AC	32
CAM entry table 22L register	TSU_ADRL22	R/W	H'E820 49B4	32
CAM entry table 23L register	TSU_ADRL23	R/W	H'E820 49BC	32
CAM entry table 24L register	TSU_ADRL24	R/W	H'E820 49C4	32
CAM entry table 25L register	TSU_ADRL25	R/W	H'E820 49CC	32

chanhvinh.com

Table 26.2 Register Configuration

Name	Abbreviation	R/W	Address	Access Size
CAM entry table 26L register	TSU_ADRL26	R/W	H'E820 49D4	32
CAM entry table 27L register	TSU_ADRL27	R/W	H'E820 49DC	32
CAM entry table 28L register	TSU_ADRL28	R/W	H'E820 49E4	32
CAM entry table 29L register	TSU_ADRL29	R/W	H'E820 49EC	32
CAM entry table 30L register	TSU_ADRL30	R/W	H'E820 49F4	32
CAM entry table 31L register	TSU_ADRL31	R/W	H'E820 49FC	32
Transmit frame counter register (normal transmission only)	TXNLCR0	R	H'E820 4880	32
Transmit frame counter register (normal and erroneous transmission)	TXALCR0	R	H'E820 4884	32
Receive frame counter register (normal reception only)	RXNLCR0	R	H'E820 4888	32
Receive frame counter register (normal and erroneous reception)	RXALCR0	R	H'E820 488C	32
E-DMAC start register	EDSR0	W	H'E820 3000	32
E-DMAC mode register	EDMR0	R/W	H'E820 3400	32
E-DMAC transmit request register	EDTRR0	R/W	H'E820 3408	32
E-DMAC receive request register	EDRRR0	R/W	H'E820 3410	32
E-MAC/E-DMAC status register	EESR0	R/W	H'E820 3428	32
E-MAC/E-DMAC status interrupt permission register	EESIPR0	R/W	H'E820 3430	32
Transmit descriptor list start address register	TDLAR0	R/W	H'E820 3010	32
Transmit descriptor fetch address register	TDFAR0	R/W	H'E820 3014	32
Transmit descriptor finished address register	TDFXR0	R/W	H'E820 3018	32
Transmit descriptor final flag register	TDFFR0	R/W	H'E820 301C	32
Receive descriptor list start address register	RDLAR0	R/W	H'E820 3030	32
Receive descriptor fetch address register	RDFAR0	R/W	H'E820 3034	32
Receive descriptor finished address register	RDFXR0	R/W	H'E820 3038	32
Receive descriptor final flag register	RDFFR0	R/W	H'E820 303C	32
Transmit/receive status copy enable register	TRSCER0	R/W	H'E820 3438	32
Receive missed-frame counter register	RMFCR0	R/W	H'E820 3440	32
Transmit FIFO threshold register	TFTR0	R/W	H'E820 3448	32
FIFO depth register	FDR0	R/W	H'E820 3450	32
Receiving method control register	RMCR0	R/W	H'E820 3458	32
Receive data padding insert register	RPADIR0	R/W	H'E820 3460	32
Overflow alert FIFO threshold register	FCFTR0	R/W	H'E820 3468	32
Intelligent checksum mode register	CSMR	R/W	H'E820 34E4	32
Intelligent checksum skipped bytes monitor register	CSSBM	R	H'E820 34E8	32
Intelligent checksum monitor register	CSSMR	R	H'E820 34EC	32

chanhvinh.com

26.3.1 Software Reset Register (ARSTR)

ARSTR resets all blocks (E-MAC, TSU, and E-DMAC) in the ETHER. By writing 1 to the ARST bit in this register, a software reset is issued to all blocks of the ETHER (for 256 cycles of internal bus clock Bφ). The ARST bit is always read as 0. While a software reset is issued, register access to all blocks of the ETHER is prohibited.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARST
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ARST	0	R/W	Software Reset When 1 is written to this bit, a software reset is issued to all blocks of the ETHER (for 256 cycles of internal bus clock Bφ). Writing 0 does not affect this bit. This bit is always read as 0. While a software reset is issued, register access to all blocks of the ETHER is prohibited. The following registers are not initialized by a software reset. TSU_ADRH0 to TSU_ADRH31, TSU_ADRL0 to TSU_ADRL31, TXNLCR0, TXALCR0, RXNLCR0, RXALCR0

chanhvinh.com

26.3.2 E-MAC Mode Register (ECMR)

ECMR is a 32-bit readable/writable register that specifies the operating mode of the ETHER. The settings in this register are normally made in the initialization process following a reset.

The operating mode setting must not be changed while the transmitting and receiving functions are enabled. To switch the operating mode, return the E-MAC and E-DMAC to their initial states by means of the SWRT and SWRR bits in EDMR before making settings again.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TRCCM	—	—	RCSC	—	DPAD	RZPF	ZPF	PFR	RXF	TXF
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	MCT	—	—	—	—	—	—	RE	TE	—	—	—	DM	PRM
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26	TRCCM	0	R/W	Counter Clear Mode Sets the method for clearing the counter register. Refer to the description of each register. 0: Cleared to 0 when the relevant register is written 1: Cleared to 0 when the relevant register is read
25, 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23	RCSC	0	R/W	Checksum Calculation Specifies whether to perform automatic calculation (hardware calculation) of the checksum of the receive frame data unit. 0: Checksum is not automatically calculated 1: Checksum is automatically calculated Note that the checksum calculation of a frame with a VLAN tag is not supported. For details, see section 26.6.1, Checksum Calculation of Ethernet Frames.
22	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
21	DPAD	0	R/W	Data Padding 0: Padding is inserted to data less than 60 bytes so it is transmitted as 60-byte data 1: Padding is not inserted to data less than 60 bytes and it is transmitted without changes
20	RZPF	0	R/W	PAUSE Frame Reception with TIME = 0 0: Reception of a PAUSE frame whose TIME parameter value is 0 is disabled 1: Reception of a PAUSE frame whose TIME parameter value is 0 is enabled
19	ZPF	0	R/W	PAUSE Frame Usage with TIME = 0 Enable 0: Control of a PAUSE frame whose TIME parameter value is 0 is disabled. The next frame is not transmitted until the time specified by the Timer value has elapsed. If a PAUSE frame whose time specified by the Timer value is 0 is received, that PAUSE frame is discarded. 1: Control of a PAUSE frame whose TIME parameter value is 0 is enabled. When the data size in the receive FIFO becomes smaller than the FCFTR setting before the time specified by the Timer value elapses, an automatic PAUSE frame with a Timer value of 0 is transmitted. On receiving a PAUSE frame with a Timer value of 0, the transmission wait state is canceled.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
18	PFR	0	R/W	PAUSE Frame Receive Mode 0: PAUSE frame is not transferred to E-DMAC 1: PAUSE frame is transferred to E-DMAC
17	RXF	0	R/W	Operating Mode for Receiving Port Flow Control 0: PAUSE frame detection is disabled 1: Flow control for the receiving port is enabled
16	TXF	0	R/W	Operating Mode for Transmitting Port Flow Control 0: Flow control for the transmitting port is disabled (Automatic PAUSE frame is not transmitted) 1: Flow control for the transmitting port is enabled (Automatic PAUSE frame is transmitted as required)
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	MCT	0	R/W	Multicast Address Frame Receive Mode 0: Frames other than the multicast address set by the CAM entry table 0 to 31 (H/L) registers are received. However, if the on-chip CAM entry table reference is disabled, all multicast address frames are received. 1: Only the multicast address set by the CAM entry table 0 to 31 (H/L) registers is received.
12 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6	RE	0	R/W	Reception Enable If a switch is made from receiving function enabled (RE = 1) to disabled (RE = 0) while a frame is being received, the receiving function will be enabled until reception of the corresponding frame is completed. 0: Receiving function is disabled 1: Receiving function is enabled
5	TE	0	R/W	Transmission Enable If a switch is made from transmitting function enabled (TE = 1) to disabled (TE = 0) while a frame is being transmitted, the transmitting function will be enabled until transmission of the corresponding frame is completed. 0: Transmitting function is disabled 1: Transmitting function is enabled
4 to 2	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	DM	0	R/W	Full-duplex Transfer Enable 0: Full-duplex transfer function is disabled. 1: Full-duplex transfer function is enabled.
0	PRM	0	R/W	Promiscuous Mode Setting this bit enables all Ethernet frames to be received. All Ethernet frames means all receivable frames, irrespective of differences or enabled/disabled status (destination address, broadcast address, multicast bit, etc.). 0: ETHER performs normal operation 1: ETHER performs promiscuous mode operation

Note: • All bits, except for TE and RE, should be changed while the transmitting function is disabled (TE = 0) and the receiving function is disabled (RE = 0).

chanhvinh.com

26.3.3 E-MAC Status Register (ECSR)

ECSR is a 32-bit readable/writable register that indicates the status in the E-MAC. This status can be notified to the CPU by interrupts. When 1 is written to the PFROI and ICD bits, the corresponding flags can be cleared. Writing 0 does not affect the flag. For bits that generate interrupts, the interrupt can be enabled or disabled by the corresponding bit in ECSIPR.

The interrupts generated due to this status register are indicated in the ECI bit in EESR of the E-DMAC.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	PFROI	—	—	—	ICD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PFROI	0	R/W	PAUSE Frame Retransmit Retry Over Indicates whether the retransmit count for retransmitting a PAUSE frame when flow control is enabled has exceeded the retransmit upper-limit set in the automatic PAUSE frame retransmit count register (TPAUSER). 0: PAUSE frame retransmit count has not exceeded the upper limit 1: PAUSE frame retransmit count has exceeded the upper limit
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ICD	0	R/W	Illegal Carrier Detection Indicates that the PHY-LSI has detected an illegal carrier on the line. If a change in the signal input from the PHY-LSI occurs in a period shorter than the software recognition period, the correct information may not be obtained. Refer to the timing specification for the PHY-LSI used. 0: PHY-LSI has not detected an illegal carrier on the line 1: PHY-LSI has detected an illegal carrier on the line

chanhvinh.com

26.3.4 E-MAC Interrupt Permission Register (ECSIPR)

ECSIPR is a 32-bit readable/writable register that enables or disables the interrupt sources indicated by ECSR. Each bit can disable or enable interrupts corresponding to the bits in ECSR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	PFROIP	PHYIP	LCHNGIP	MPDIP	ICDIP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PFROIP	0	R/W	PAUSE Frame Retransmit Interrupt Enable 0: Interrupt notification by the PFROI bit is disabled 1: Interrupt notification by the PFROI bit is enabled
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ICDIP	0	R/W	Illegal Carrier Detect Interrupt Enable 0: Interrupt notification by the ICD bit is disabled 1: Interrupt notification by the ICD bit is enabled

chanhvinh.com

26.3.5 PHY Interface Register (PIR)

PIR is a 32-bit readable/writable register that provides a means of accessing the PHY-LSI internal registers via the MII.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	MDI	MDO	MMD	MDC
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	—	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	MDI	Undefined	R	MII Management Data-In Indicates the level of the ET_MDIO pin.
2	MDO	0	R/W	MII Management Data-Out Outputs the value set in this bit from the ET_MDIO pin when the MMD bit is 1.
1	MMD	0	R/W	MII Management Mode Specifies the data read/write direction with respect to the MII. 0: Read direction is specified 1: Write direction is specified
0	MDC	0	R/W	MII Management Data Clock Outputs the value set in this bit from the MDC pin and supplies the MII with the management data clock. For the method of accessing the MII registers, see section 26.5.2, Accessing MII Registers.

26.3.6 MAC Address High Register (MAHR)

MAHR is a 32-bit readable/writable register that specifies the upper 32 bits of the 48-bit MAC address. The settings in this register are normally made in the initialization process after a reset. The MAC address setting must not be changed while the transmitting and receiving functions are enabled. Return the E-MAC and E-DMAC to their initial states by means of the SWRT and SWRR bits in EDMR before making settings again.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MA[47:32]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

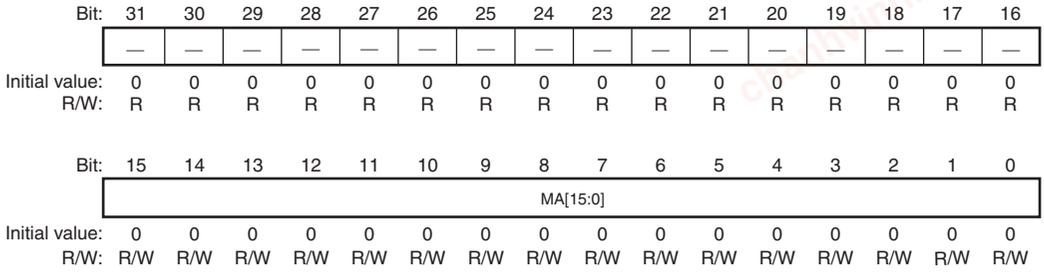
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MA[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MA[47:16]	All 0	R/W	MAC Address Bits 47 to 16 These bits are used to set the upper 32 bits of the MAC address. If the MAC address is 01-23-45-67-89-AB (hexadecimal), set H'01234567 in this register.

chanhvinh.com

26.3.7 MAC Address Low Register (MALR)

MALR is a 32-bit readable/writable register that specifies the lower 16 bits of the 48-bit MAC address. The settings in this register are normally made in the initialization process after a reset. The MAC address setting must not be changed while the transmitting and receiving functions are enabled. Return the E-MAC and E-DMAC to their initial states by means of the SWRT and SWRR bits in EDMR before making settings again.



Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	MA[15:0]	All 0	R/W	MAC Address Bits 15 to 0 These bits are used to set the lower 16 bits of the MAC address. If the MAC address is 01-23-45-67-89-AB (hexadecimal), set H'000089AB in this register.

chanhvinh.com

26.3.8 Receive Frame Length Register (RFLR)

RFLR is a 32-bit readable/writable register that specifies the maximum frame length (in bytes) that can be received by this LSI. The settings in this register must not be changed while the receiving function is enabled.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFL[17:16]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

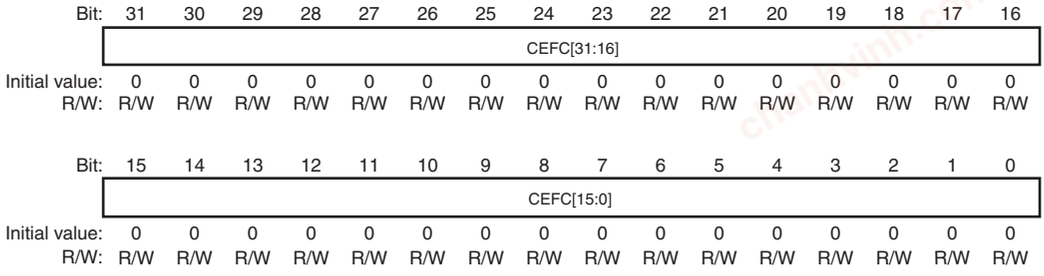
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFL[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17 to 0	RFL[17:0]	All 0	R/W	Receive Frame Length The frame data described here refers to all fields from the destination address up to the CRC data. Frame contents from the destination address up to the data are actually transferred to memory. CRC data is not included in the transfer. When data that exceeds the specified value is received, the part of data that exceeds the specified value is discarded. H'00000 to H'005EE: 1,518 bytes H'005EF: 1,519 bytes H'005F0: 1,520 bytes : : H'007FF: 2,047 bytes H'00800: 2,048 bytes : : H'01000: 4,096 bytes : : H'10000: 65,536 bytes : : H'20000 to H'3FFFF: 131,072 bytes

chanhvinh.com

26.3.9 CRC Error Frame Receive Counter Register (CEFCR)

CEFCR is a 32-bit counter that indicates the number of times a frame with a CRC error was received. When the value in this register reaches H'FFFFFFFF, count-up is halted. This register is cleared to 0 when it is read with the TRCCM bit in ECMR set to 1. When the TRCCM bit in ECMR is 0, writing to this register will clear it regardless of the value written.

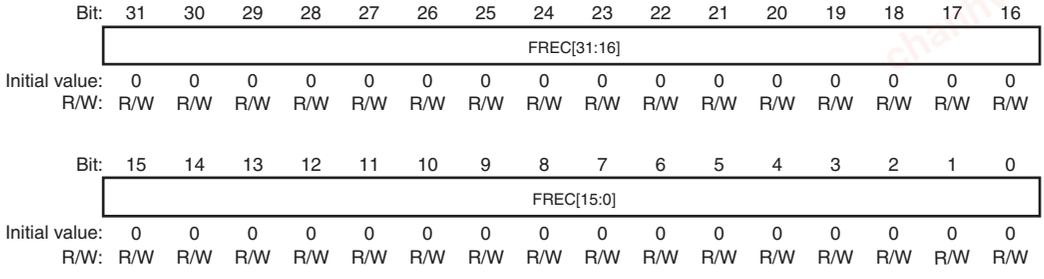


Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CEFC[31:0]	All 0	R/W	CRC Error Frame Count These bits indicate the number of CRC error frames received.

Note: • When count-up and count-clearing of this register value occur simultaneously, count-clearing is performed.

26.3.10 Frame Receive Error Counter Register (FRECR)

FRECR is a 32-bit counter that indicates the number of frames for which a receive error was generated by the RXER pin input from the PHY-LSI. FRECR is incremented each time the RXER pin becomes active. When the value in this register reaches H'FFFFFFFF, count-up is halted. This register is cleared to 0 when it is read with the TRCCM bit in ECMR set to 1. When the TRCCM bit in ECMR is 0, writing to this register will clear it regardless of the value written.



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	FREC[31:0]	All 0	R/W	Frame Receive Error Count These bits indicate the number of errors during frame reception.

Note: • When count-up and count-clearing of this register value occur simultaneously, count-clearing is performed.

chanhvinh.com

26.3.11 Too-Short Frame Receive Counter Register (TSFRCR)

TSFRCR is a 32-bit counter that indicates the number of frames received with a length fewer than 64 bytes. When the value in this register reaches H'FFFFFFFF, count-up is halted. This register is cleared to 0 when it is read with the TRCCM bit in ECMR set to 1. When the TRCCM bit in ECMR is 0, writing to this register will clear it regardless of the value written.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TSFC[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSFC[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TSFC[31:0]	All 0	R/W	Too-Short Frame Receive Count These bits indicate the number of frames received with a length of less than 64 bytes.

Note: • When count-up and count-clearing of this register value occur simultaneously, count-clearing is performed.

26.3.12 Too-Long Frame Receive Counter Register (TLFRCR)

TLFRCR is a 32-bit counter that indicates the number of frames received with a length exceeding the value specified by the receive frame length register (RFLR). When the value in this register reaches H'FFFFFFFF, count-up is halted. This register is not incremented when a frame containing residual bits is received. In this case, the reception of the frame is indicated in the residual-bit frame receive counter register (RFCR). This register is cleared to 0 when it is read with the TRCCM bit in ECMR set to 1. When the TRCCM bit in ECMR is 0, writing to this register will clear it regardless of the value written.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TLFC[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TLFC[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TLFC[31:0]	All 0	R/W	Too-Long Frame Receive Count These bits indicate the number of frames received with a length exceeding the value in RFLR.

Note: • When count-up and count-clearing of this register value occur simultaneously, count-clearing is performed.

chanhvinh.com

26.3.13 Residual-Bit Frame Receive Counter Register (RFCR)

RFCR is a 32-bit counter that indicates the number of frames received containing residual bits (less than an 8-bit unit). When the value in this register reaches H'FFFFFFFF, count-up is halted. This register is cleared to 0 when it is read with the TRCCM bit in ECMR set to 1. When the TRCCM bit in ECMR is 0, writing to this register will clear it regardless of the value written.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFC[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFC[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RFC[31:0]	All 0	R/W	Residual-Bit Frame Receive Count These bits indicate the number of frames received containing residual bits.

Note: • When count-up and count-clearing of this register value occur simultaneously, count-clearing is performed.

26.3.14 Multicast Address Frame Receive Counter Register (MAFCR)

MAFCR is a 32-bit counter that indicates the number of frames received with a specified multicast address. When the value in this register reaches H'FFFFFFFF, count-up is halted. This register is cleared to 0 when it is read with the TRCCM bit in ECMR set to 1. When the TRCCM bit in ECMR is 0, writing to this register will clear it regardless of the value written.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MAFC[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MAFC[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

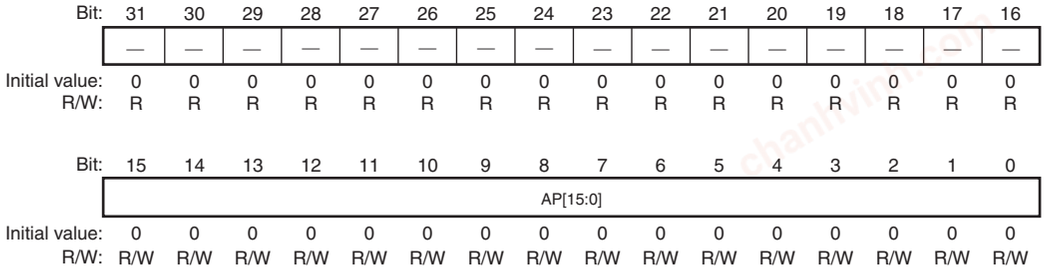
Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MAFC[31:0]	All 0	R/W	Multicast Address Frame Count These bits indicate the number of multicast frames received.

Note: • When count-up and count-clearing of this register value occur simultaneously, count-clearing is performed.

chanhvinh.com

26.3.15 Automatic PAUSE Frame Register (APR)

APR is used to set the TIME parameter value of an automatic PAUSE frame. When an automatic PAUSE frame is transmitted, the value set in this register is used as the TIME parameter of the PAUSE frame.



Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	AP[15:0]	All 0	R/W	Automatic PAUSE These bits set the TIME parameter value of an automatic PAUSE frame. One bit is equivalent to 512 bit-time. When flow control is enabled in transmission (PAUSE frame transmission) (TXF bit in ECMR = 1), set a value other than H'0000 in these bits. H'0000: — H'0001: 512 × 1 bit-time H'0002: 512 × 2 bit-time : : H'FFFF: 512 × 65,535 bit-time Note:The bit-time becomes as follows according to the transfer speed. 100 Mbps: 1 bit-time = 10 ns 10 Mbps: 1 bit-time = 100 ns

chanhvinh.com

26.3.16 Manual PAUSE Frame Register (MPR)

MPR is used to set the TIME parameter value of a manual PAUSE frame. When a manual PAUSE frame is transmitted, the value set in this register is used as the TIME parameter of the PAUSE frame.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MP[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	MP[15:0]	All 0	R/W	Manual PAUSE These bits set the TIME parameter value of a manual PAUSE frame. One bit is equivalent to 512 bit-time. H'0000: — H'0001: 512 × 1 bit-time H'0002: 512 × 2 bit-time : : H'FFFF: 512 × 65,535 bit-time Note:The bit-time becomes as follows according to the transfer speed. 100 Mbps: 1 bit-time = 10 ns 10 Mbps: 1 bit-time = 100 ns

26.3.17 Automatic PAUSE Frame Retransmit Count Register (TPAUSER)

TPAUSER is used to set the upper limit for the number of times to retransmit an automatic PAUSE frame. The settings in this register must not be changed while the transmitting function is enabled.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TPAUSE[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	TPAUSE[15:0]	All 0	R/W	Upper Limit for Automatic PAUSE Frame Retransmission H'0000: Retransmit count is unlimited H'0001: Retransmit count is 1 : : H'FFFF: Retransmit count is 65,535

chanhvinh.com

26.3.18 PAUSE Frame Transmit Counter Register (PFTCR)

PFTCR is a 16-bit counter that indicates the number of times a PAUSE frame is transmitted. This register is cleared to 0 when it is read.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFTXC[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	PFTXC[15:0]	All 0	R	PAUSE Frame Transmit Count These bits indicate the total number of automatic PAUSE frames and manual PAUSE frames transmitted.

Note: • When count-up and count-clearing of this register value occur simultaneously, count-clearing is performed.

26.3.19 PAUSE Frame Receive Counter Register (PFRCR)

PFRCR is a 16-bit counter that indicates the number of times a PAUSE frame is received. This register is cleared to 0 when it is read.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFRXC[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	PFRXC[15:0]	All 0	R	PAUSE Frame Receive Count These bits indicate the number of PAUSE frames received when flow control is enabled in reception (RXF bit in ECMR = 1).

Note: • When count-up and count-clearing of this register value occur simultaneously, count-clearing is performed.

chanhvinh.com

26.3.20 TSU Counter Reset Register (TSU_CTRST)

TSU_CTRST clears the transmit, receive, and relay frame counters to 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CTRST	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	CTRST	0	R/W	TSU Counter Reset When 1 is written to this bit, the values of registers TXNLCR0, TXALCR0, RXNLCR0, and RXALCR0 are cleared to 0. Writing 0 does not affect this bit. This bit is always read as 0.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

26.3.21 CAM Entry Table Specification Enable Register (Common) (TSU_FWSLC)

When the CAM function is in use, you can use the TSU_POST1 to TSU_POST4 registers to specify which of among the CAM entry tables are for reference or reference to all tables. The TSU_FWSLC register enables or disables this action of the settings of the TSU_POST1 to TSU_POST4 registers.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	POSTENU	POSTENL	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	POSTENU	0	R/W	This bit enables the settings for CAM entry tables 0 to 15 in the POST bits of the TSU_POST1 and TSU_POST2 registers. 0: The settings of the POST bits are disabled (CAM entry tables 0 to 15 are referred to on reception of frames). 1: The settings of the POST bits are enabled (whether to refer to each of CAM entry tables is determined by the settings of the corresponding POST bit).
12	POSTENL	0	R/W	This bit enables the settings for CAM entry tables 16 to 31 in the POST bits of the TSU_POST3 and TSU_POST4 registers. 0: The settings of the POST bits are disabled (CAM entry tables 16 to 31 are not referred to on reception of frames). 1: The settings of the POST bits are enabled (whether to refer to each of CAM entry tables is determined by the settings of the corresponding POST bit).
11 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

26.3.22 VLANtag Set Register (TSU_VTAG0)

TSU_VTAG0 enables or disables the frame receive/discard evaluation function based on the VLAN number, and also sets the VLAN number.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VTAG0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	VID0[11:0]											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	VTAG0	0	R/W	Port 0 VLANtag Evaluation Function 0: Disables receive/discard evaluation for frames based on the VLAN number 1: Enables receive/discard evaluation for frames based on the VLAN number
30 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	VID0[11:0]	All 0	R/W	V-LAN ID Setting (VID) These bits set the VLAN number received by receive frames.

chanhvinh.com

26.3.23 CAM Entry Table Busy Register (TSU_ADSBSY)

When CAM entry table registers (TSU_ADRH0 to TSU_ADRH31 and TSU_ADRL0 to TSU_ADRL31) are set by register writing, the ADSBSY bit in this register is set to 1 (when the process of reflecting the contents of the CAM entry table registers in the CAM controller is completed inside the TSU, the ADSBSY bit is automatically restored to 0). Access to TSU_ADRH0 to TSU_ADRH31 and TSU_ADRL0 to TSU_ADRL31 is prohibited, while the ADSBSY bit in this register is set to 1. This register is a read-only status register, which must not be written to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADS BSY
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ADSBSY	0	R	CAM Entry Table Setting Busy When TSU_ADRH0 to TSU_ADRH31 and TSU_ADRL0 to TSU_ADRL31 are set by register writing, this bit is set to 1. When the process of reflecting the contents of the CAM entry table registers in the CAM controller is completed inside the TSU, this bit is automatically restored to 0. Access to TSU_ADRH0 to TSU_ADRH31 and TSU_ADRL0 to TSU_ADRL31 is prohibited, while this bit is set to 1. Writing to this register is also prohibited.

chanhvinh.com

26.3.24 CAM Entry Table Enable Register (TSU_TEN)

TSU_TEN enables or disables the settings of TSU_ADRH0 to TSU_ADRH31 and TSU_ADRL0 to TSU_ADRL31.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TEN0	TEN1	TEN2	TEN3	TEN4	TEN5	TEN6	TEN7	TEN8	TEN9	TEN10	TEN11	TEN12	TEN13	TEN14	TEN15
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TEN16	TEN17	TEN18	TEN19	TEN20	TEN21	TEN22	TEN23	TEN24	TEN25	TEN26	TEN27	TEN28	TEN29	TEN30	TEN31
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	TEN0	0	R/W	CAM Entry Table 0 (TSU_ADRH0 and TSU_ADRL0) Setting 0: Disabled 1: Enabled
30	TEN1	0	R/W	CAM Entry Table 1 (TSU_ADRH1 and TSU_ADRL1) Setting 0: Disabled 1: Enabled
29	TEN2	0	R/W	CAM Entry Table 2 (TSU_ADRH2 and TSU_ADRL2) Setting 0: Disabled 1: Enabled
28	TEN3	0	R/W	CAM Entry Table 3 (TSU_ADRH3 and TSU_ADRL3) Setting 0: Disabled 1: Enabled
27	TEN4	0	R/W	CAM Entry Table 4 (TSU_ADRH4 and TSU_ADRL4) Setting 0: Disabled 1: Enabled
26	TEN5	0	R/W	CAM Entry Table 5 (TSU_ADRH5 and TSU_ADRL5) Setting 0: Disabled 1: Enabled
25	TEN6	0	R/W	CAM Entry Table 6 (TSU_ADRH6 and TSU_ADRL6) Setting 0: Disabled 1: Enabled
24	TEN7	0	R/W	CAM Entry Table 7 (TSU_ADRH7 and TSU_ADRL7) Setting 0: Disabled 1: Enabled
23	TEN8	0	R/W	CAM Entry Table 8 (TSU_ADRH8 and TSU_ADRL8) Setting 0: Disabled 1: Enabled
22	TEN9	0	R/W	CAM Entry Table 9 (TSU_ADRH9 and TSU_ADRL9) Setting 0: Disabled 1: Enabled
21	TEN10	0	R/W	CAM Entry Table 10 (TSU_ADRH10 and TSU_ADRL10) Setting 0: Disabled 1: Enabled
20	TEN11	0	R/W	CAM Entry Table 11 (TSU_ADRH11 and TSU_ADRL11) Setting 0: Disabled 1: Enabled
19	TEN12	0	R/W	CAM Entry Table 12 (TSU_ADRH12 and TSU_ADRL12) Setting 0: Disabled 1: Enabled
18	TEN13	0	R/W	CAM Entry Table 13 (TSU_ADRH13 and TSU_ADRL13) Setting 0: Disabled 1: Enabled

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
17	TEN14	0	R/W	CAM Entry Table 14 (TSU_ADRH14 and TSU_ADRL14) Setting 0: Disabled 1: Enabled
16	TEN15	0	R/W	CAM Entry Table 15 (TSU_ADRH15 and TSU_ADRL15) Setting 0: Disabled 1: Enabled
15	TEN16	0	R/W	CAM Entry Table 16 (TSU_ADRH16 and TSU_ADRL16) Setting 0: Disabled 1: Enabled
14	TEN17	0	R/W	CAM Entry Table 17 (TSU_ADRH17 and TSU_ADRL17) Setting 0: Disabled 1: Enabled
13	TEN18	0	R/W	CAM Entry Table 18 (TSU_ADRH18 and TSU_ADRL18) Setting 0: Disabled 1: Enabled
12	TEN19	0	R/W	CAM Entry Table 19 (TSU_ADRH19 and TSU_ADRL19) Setting 0: Disabled 1: Enabled
11	TEN20	0	R/W	CAM Entry Table 20 (TSU_ADRH20 and TSU_ADRL20) Setting 0: Disabled 1: Enabled
10	TEN21	0	R/W	CAM Entry Table 21 (TSU_ADRH21 and TSU_ADRL21) Setting 0: Disabled 1: Enabled
9	TEN22	0	R/W	CAM Entry Table 22 (TSU_ADRH22 and TSU_ADRL22) Setting 0: Disabled 1: Enabled
8	TEN23	0	R/W	CAM Entry Table 23 (TSU_ADRH23 and TSU_ADRL23) Setting 0: Disabled 1: Enabled
7	TEN24	0	R/W	CAM Entry Table 24 (TSU_ADRH24 and TSU_ADRL24) Setting 0: Disabled 1: Enabled
6	TEN25	0	R/W	CAM Entry Table 25 (TSU_ADRH25 and TSU_ADRL25) Setting 0: Disabled 1: Enabled
5	TEN26	0	R/W	CAM Entry Table 26 (TSU_ADRH26 and TSU_ADRL26) Setting 0: Disabled 1: Enabled
4	TEN27	0	R/W	CAM Entry Table 27 (TSU_ADRH27 and TSU_ADRL27) Setting 0: Disabled 1: Enabled
3	TEN28	0	R/W	CAM Entry Table 28 (TSU_ADRH28 and TSU_ADRL28) Setting 0: Disabled 1: Enabled
2	TEN29	0	R/W	CAM Entry Table 29 (TSU_ADRH29 and TSU_ADRL29) Setting 0: Disabled 1: Enabled
1	TEN30	0	R/W	CAM Entry Table 30 (TSU_ADRH30 and TSU_ADRL30) Setting 0: Disabled 1: Enabled
0	TEN31	0	R/W	CAM Entry Table 31 (TSU_ADRH31 and TSU_ADRL31) Setting 0: Disabled 1: Enabled

chanhvinh.com

26.3.25 CAM Entry Table POST 1 Register (TSU_POST1)

When the CAM function is in use, you can use the TSU_POST1 to TSU_POST4 registers to specify which of the CAM entry tables are for reference. The TSU_POST1 register specifies the conditions for reference to TSU_ADRH0 to TSU_ADRH7 and TSU_ADRL0 to TSU_ADRL7. The setting of this register is effective when the POSTENU bit of the TSU_FWSLC register is set to 1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	POST0	—	—	—	POST1	—	—	—	POST2	—	—	—	POST3	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	POST4	—	—	—	POST5	—	—	—	POST6	—	—	—	POST7	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	POST0	0	R/W	Specifies whether reference is or is not made to CAM entry table 0. Setting this bit to 1 selects the condition below. POST0: CAM entry table 0 is referred to on reception of frames.
30 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27	POST1	0	R/W	Specifies whether reference is or is not made to CAM entry table 1. Setting this bit to 1 selects the condition below. POST1: CAM entry table 1 is referred to on reception of frames.
26 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23	POST2	0	R/W	Specifies whether reference is or is not made to CAM entry table 2. Setting this bit to 1 selects the condition below. POST2: CAM entry table 2 is referred to on reception of frames.
22 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19	POST3	0	R/W	Specifies whether reference is or is not made to CAM entry table 3. Setting this bit to 1 selects the condition below. POST3: CAM entry table 3 is referred to on reception of frames.
18 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15	POST4	0	R/W	Specifies whether reference is or is not made to CAM entry table 4. Setting this bit to 1 selects the condition below. POST4: CAM entry table 4 is referred to on reception of frames.
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	POST5	0	R/W	Specifies whether reference is or is not made to CAM entry table 5. Setting this bit to 1 selects the condition below. POST5: CAM entry table 5 is referred to on reception of frames.
10 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	POST6	0	R/W	Specifies whether reference is or is not made to CAM entry table 6. Setting this bit to 1 selects the condition below. POST6: CAM entry table 6 is referred to on reception of frames.
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	POST7	0	R/W	Specifies whether reference is or is not made to CAM entry table 7. Setting this bit to 1 selects the condition below. POST7: CAM entry table 7 is referred to on reception of frames.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

26.3.26 CAM Entry Table POST 2 Register (TSU_POST2)

When the CAM function is in use, you can use the TSU_POST1 to TSU_POST4 registers to specify which of the CAM entry tables are for reference. The TSU_POST2 register specifies the conditions for reference to TSU_ADRH8 to TSU_ADRH15 and TSU_ADRL8 to TSU_ADRL15. The setting of this register is effective when the POSTENU bit of the TSU_FWSLC register is set to 1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	POST8	—	—	—	POST9	—	—	—	POST10	—	—	—	POST11	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	POST12	—	—	—	POST13	—	—	—	POST14	—	—	—	POST15	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	POST8	0	R/W	Specifies whether reference is or is not made to CAM entry table 8. Setting this bit to 1 selects the condition below. POST8: CAM entry table 8 is referred to on reception of frames.
30 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27	POST9	0	R/W	Specifies whether reference is or is not made to CAM entry table 9. Setting this bit to 1 selects the condition below. POST9: CAM entry table 9 is referred to on reception of frames.
26 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23	POST10	0	R/W	Specifies whether reference is or is not made to CAM entry table 10. Setting this bit to 1 selects the condition below. POST10: CAM entry table 10 is referred to on reception of frames.
22 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19	POST11	0	R/W	Specifies whether reference is or is not made to CAM entry table 11. Setting this bit to 1 selects the condition below. POST11: CAM entry table 11 is referred to on reception of frames.
18 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15	POST12	0	R/W	Specifies whether reference is or is not made to CAM entry table 12. Setting this bit to 1 selects the condition below. POST12: CAM entry table 12 is referred to on reception of frames.
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	POST13	0	R/W	Specifies whether reference is or is not made to CAM entry table 13. Setting this bit to 1 selects the condition below. POST13: CAM entry table 13 is referred to on reception of frames.
10 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
7	POST14	0	R/W	Specifies whether reference is or is not made to CAM entry table 14. Setting this bit to 1 selects the condition below. POST14: CAM entry table 14 is referred to on reception of frames.
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	POST15	0	R/W	Specifies whether reference is or is not made to CAM entry table 15. Setting this bit to 1 selects the condition below. POST15: CAM entry table 15 is referred to on reception of frames.
2 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

26.3.27 CAM Entry Table POST 3 Register (TSU_POST3)

When the CAM function is in use, you can use the TSU_POST1 to TSU_POST4 registers to specify which of the CAM entry tables are for reference. The TSU_POST3 register specifies the conditions for reference to TSU_ADRH16 to TSU_ADRH23 and TSU_ADRL16 to TSU_ADRL23. The setting of this register is effective when the POSTENL bit of the TSU_FWSLC register is set to 1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	POST16	—	—	—	POST17	—	—	—	POST18	—	—	—	POST19	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	POST20	—	—	—	POST21	—	—	—	POST22	—	—	—	POST23	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	POST16	0	R/W	Specifies whether reference is or is not made to CAM entry table 16. Setting this bit to 1 selects the condition below. POST16: CAM entry table 16 is referred to on reception of frames.
30 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27	POST17	0	R/W	Specifies whether reference is or is not made to CAM entry table 17. Setting this bit to 1 selects the condition below. POST17: CAM entry table 17 is referred to on reception of frames.
26 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23	POST18	0	R/W	Specifies whether reference is or is not made to CAM entry table 18. Setting this bit to 1 selects the condition below. POST18: CAM entry table 18 is referred to on reception of frames.
22 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19	POST19	0	R/W	Specifies whether reference is or is not made to CAM entry table 19. Setting this bit to 1 selects the condition below. POST19: CAM entry table 19 is referred to on reception of frames.
18 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15	POST20	0	R/W	Specifies whether reference is or is not made to CAM entry table 20. Setting this bit to 1 selects the condition below. POST20: CAM entry table 20 is referred to on reception of frames.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	POST21	0	R/W	Specifies whether reference is or is not made to CAM entry table 21. Setting this bit to 1 selects the condition below. POST21: CAM entry table 21 is referred to on reception of frames.
10 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	POST22	0	R/W	Specifies whether reference is or is not made to CAM entry table 22. Setting this bit to 1 selects the condition below. POST22: CAM entry table 22 is referred to on reception of frames.
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	POST23	0	R/W	Specifies whether reference is or is not made to CAM entry table 23. Setting this bit to 1 selects the condition below. POST23: CAM entry table 23 is referred to on reception of frames.
2 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

26.3.28 CAM Entry Table POST 4 Register (TSU_POST4)

When the CAM function is in use, you can use the TSU_POST1 to TSU_POST4 registers to specify which of the CAM entry tables are for reference. The TSU_POST4 register specifies the conditions for reference to TSU_ADRH24 to TSU_ADRH31 and TSU_ADRL24 to TSU_ADRL31. The setting of this register is effective when the POSTENL bit of the TSU_FWSLC register is set to 1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	POST24	—	—	—	POST25	—	—	—	POST26	—	—	—	POST27	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	POST28	—	—	—	POST29	—	—	—	POST30	—	—	—	POST31	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	POST24	0	R/W	Specifies whether reference is or is not made to CAM entry table 24. Setting this bit to 1 selects the condition below. POST24: CAM entry table 24 is referred to on reception of frames.
30 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27	POST25	0	R/W	Specifies whether reference is or is not made to CAM entry table 25. Setting this bit to 1 selects the condition below. POST25: CAM entry table 25 is referred to on reception of frames.
26 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23	POST26	0	R/W	Specifies whether reference is or is not made to CAM entry table 26. Setting this bit to 1 selects the condition below. POST26: CAM entry table 26 is referred to on reception of frames.
22 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
19	POST27	0	R/W	Specifies whether reference is or is not made to CAM entry table 27. Setting this bit to 1 selects the condition below. POST27: CAM entry table 27 is referred to on reception of frames.
18 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15	POST28	0	R/W	Specifies whether reference is or is not made to CAM entry table 28. Setting this bit to 1 selects the condition below. POST28: CAM entry table 28 is referred to on reception of frames.
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	POST29	0	R/W	Specifies whether reference is or is not made to CAM entry table 29. Setting this bit to 1 selects the condition below. POST29: CAM entry table 29 is referred to on reception of frames.
10 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	POST30	0	R/W	Specifies whether reference is or is not made to CAM entry table 30. Setting this bit to 1 selects the condition below. POST30: CAM entry table 30 is referred to on reception of frames.
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	POST31	0	R/W	Specifies whether reference is or is not made to CAM entry table 31. Setting this bit to 1 selects the condition below. POST31: CAM entry table 31 is referred to on reception of frames.
2 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

26.3.29 CAM Entry Table 0H to 31H Registers (TSU_ADRH0 to TSU_ADRH31)

TSU_ADRH0 to TSU_ADRH31 are entry tables referred to by the CAM in reception and relay. Each of these registers sets the upper 32 bits of the 48-bit MAC address. Maximum 32 entries of MAC addresses can be registered.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADRHn[31:16] (n = 0 to 31)															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADRHn[15:0] (n = 0 to 31)															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	ADRHn[31:0] (n: 0 to 31)	All 0	R/W	MAC Address Bits These bits set the upper 32 bits of the MAC address. When the MAC address is 01-23-45-67-89-AB (displayed in hexadecimal), set H'01234567 in this register.

Note: • Set the CAM entry tables following the procedure below.
1. Check that the ADSBSY bit in TSU_ADSBSY is cleared to 0.
2. Set the upper 32 bits of the MAC addresses by TSU_ADRH0 to TSU_ADRH31.
3. Set the lower 16 bits of the MAC addresses by TSU_ADRL0 to TSU_ADRL31.

26.3.30 CAM Entry Table 0L to 31L Registers (TSU_ADRL0 to TSU_ADRL31)

TSU_ADRL0 to TSU_ADRL31 are entry tables referred to by the CAM in reception and relay. Each of these registers sets the lower 16 bits of the 48-bit MAC address. Maximum 32 entries of MAC addresses can be registered.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADRLn[15:0] (n = 0 to 31)															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	ADRLn[15:0] (n: 0 to 31)	All 0	R/W	MAC Address Bits These bits set the lower 16 bits of the MAC address. When the MAC address is 01-23-45-67-89-AB (displayed in hexadecimal), set H'000089AB in this register.

Note: • Set the CAM entry tables following the procedure below.
1. Check that the ADSBSY bit in TSU_ADSBSY is cleared to 0.
2. Set the upper 32 bits of the MAC addresses by TSU_ADRH0 to TSU_ADRH31.
3. Set the lower 16 bits of the MAC addresses by TSU_ADRL0 to TSU_ADRL31.

chanhvinh.com

26.3.31 Transmit Frame Counter Register (Normal Transmission Only) (TXNLCR0)

TXNLCR0 is a 32-bit counter indicating the number of frames successfully transmitted in the E-MAC. When the value in this register reaches H'FFFFFFFF, count-up is halted. The counter is cleared to 0 by reading from this register. This register cannot be written to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NTC0[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NTC0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	NTC0[31:0]	All 0	R	Transmit Frame Counter Bits These bits indicate the number of frames successfully transmitted.

Note: • When count-up and count-clearing of this register value occur simultaneously, count-clearing is performed.

26.3.32 Transmit Frame Counter Register (Normal and Erroneous Transmission) (TXALCR0)

TXALCR0 is a 32-bit counter indicating the number of frames transmitted in the E-MAC, including the number of frames erroneously transmitted. When the value in this register reaches H'FFFFFFFF, count-up is halted. The counter is cleared to 0 by reading from this register. This register cannot be written to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TC0[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TC0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TC0[31:0]	All 0	R	Transmit Frame Counter Bits These bits indicate the number of frames successfully transmitted and erroneously transmitted.

Note: • When count-up and count-clearing of this register value occur simultaneously, count-clearing is performed.

chanhvinh.com

26.3.33 Receive Frame Counter Register (Normal Reception Only) (RXNLCR0)

RXNLCR0 is a 32-bit counter indicating the number of frames successfully received in the E-MAC. When the value in this register reaches H'FFFFFFFF, count-up is halted. The counter is cleared to 0 by reading from this register. This register cannot be written to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NRC0[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NRC0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	NRC0[31:0]	All 0	R	Receive Frame Counter Bits These bits indicate the number of frames successfully received.

Note: • When count-up and count-clearing of this register value occur simultaneously, count-clearing is performed.

26.3.34 Receive Frame Counter Register (Normal and Erroneous Reception) (RXALCR0)

RXALCR0 is a 32-bit counter indicating the number of frames received in the E-MAC, including the number of frames erroneously received. When the value in this register reaches H'FFFFFFFF, count-up is halted. The counter is cleared to 0 by reading from this register. This register cannot be written to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RC0[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RC0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RC0[31:0]	All 0	R	Receive Frame Counter Bits These bits indicate the number of frames successfully received and erroneously received.

Note: • When count-up and count-clearing of this register value occur simultaneously, count-clearing is performed.

chanhvinh.com

26.3.35 E-DMAC Start Register (EDSR)

EDSR specifies activation of the transmitting unit and receiving unit of the E-DMAC. This register can only be written to, and the read values are invalid.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ENT	ENR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	ENT	0	W	E-DMAC Transmitting Unit Start 0: Stops the E-DMAC transmitting unit 1: Starts the E-DMAC transmitting unit
0	ENR	0	W	E-DMAC Receiving Unit Start 0: Stops the E-DMAC receiving unit 1: Starts the E-DMAC receiving unit

chanhvinh.com

26.3.36 E-DMAC Mode Register (EDMR)

EDMR is a 32-bit readable/writable register that specifies E-DMAC resetting and the transmit/receive descriptor length. This register is to be set before the transmitting or receiving function is enabled (before the TR bit in EDTRR or the RR bit in EDRRR is set to 1). However, the SWRR and SWRT bits can be written to even after the transmitting or receiving function is enabled. If a software reset is executed with this register during data transmission, abnormal data may be transmitted on the line. Execute a software reset with this register before specifying the transmit/receive descriptor length or modifying the settings of TDLAR, RDLAR, and so forth, the setting of ECMR (E-MAC mode register), and the settings of registers related to the E-DMAC and E-MAC operation.

To execute a software reset with this register, 1 must be written to both the SWRT and SWRR bits simultaneously. Writing 1 to the SWRT and SWRR bits initializes the E-MAC registers and E-DMAC registers, except for TDLAR, RDLAR, and RMFCR of the E-DMAC. The TSU registers (registers whose names are prefixed with TSU_) are not initialized. The SWRT and SWRR bits in EDMR0 initializes the registers related to the E-DMAC and E-MAC. Note that during the period a software reset is issued (for 64 cycles of the internal bus clock Bck), accesses to all Ethernet-related registers are prohibited.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	DE	DL[1:0]		—	—	SWRT	SWRR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6	DE	0	R/W	Transmit/Receive Frame Endian Sets the endian mode for DMA transfer of frame data between the transmit/receive FIFO and transmit/receive buffer. 0: Big endian (longword access) 1: Little endian (longword access)
5, 4	DL[1:0]	00	R/W	Transmit/Receive Descriptor Length These bits specify the descriptor length. (See section 26.4.1, Descriptors and Descriptor List.) 00: 16 bytes 01: 32 bytes 10: 64 bytes 11: Setting prohibited
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	SWRT	0	R/W	Software Reset of Transmit FIFO Controller [Writing] 0: Disabled 1: Software reset started [Reading] 0: Software reset not executed (or completed) 1: Software reset being executed
0	SWRR	0	R/W	Software Reset of Receive FIFO Controller [Writing] 0: Disabled 1: Software reset started [Reading] 0: Software reset not executed (or completed) 1: Software reset being executed

chanhvinh.com

26.3.37 E-DMAC Transmit Request Register (EDTRR)

EDTRR is a 32-bit readable/writable register that issues transmit directives to the E-DMAC. After writing 11 to bits TR[1:0] in this register, the E-DMAC reads the transmit descriptor at the address specified by TDLAR. If the TACT bit of this transmit descriptor is set to 1 (valid), transmit DMA transfer by the E-DMAC starts. When DMA transfer based on the first transmit descriptor is completed, the E-DMAC reads the next transmit descriptor. If the TACT bit of that transmit descriptor is set to 1 (valid), the E-DMAC continues transmit DMA operation. If the TACT bit of a transmit descriptor is cleared to 0 (invalid), the E-DMAC clears bits TR[1:0] and stops transmit DMA operation.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TR[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	TR[1:0]	00	R/W	Transmit Request 00, 01, 10: Transmission-halted state If 00, 01, or 10 is written to these bits, the E-DMAC stops DMA transfer of the currently processed transmit descriptor, reads the next transmit descriptor, and then clears these bits. (Write-back is completed for the valid transmit descriptors that have been detected up till then.) The E-DMAC clears these bits when transmit descriptor empty occurs, or transmission of a transmit descriptor has completed. (Write-back is completed for the valid transmit descriptors that have been detected up till then.) 11: Transmit DMA operation by E-DMAC After writing 11 to these bits, the E-DMAC starts reading a transmit descriptor.

chanhvinh.com

26.3.38 E-DMAC Receive Request Register (EDRRR)

EDRRR is a 32-bit readable/writable register that issues receive directives to the E-DMAC. After writing 1 to the RR bit in this register, the E-DMAC reads the receive descriptor at the address specified by RDLAR. If the RACT bit of this receive descriptor is set to 1 (valid), and the receive FIFO holds a receive frame, the E-DMAC starts receive DMA transfer. When DMA transfer based on the first receive descriptor is completed, the E-DMAC reads the next receive descriptor. If the RACT bit of that receive descriptor is set to 1 (valid), the E-DMAC continues receive DMA operation. However, if the receive FIFO holds no receive data, the E-DMAC places receive DMA operation in the standby state. If the RACT bit of the receive descriptor is cleared to 0 (invalid), the E-DMAC clears the RR bit and stops receive DMAC operation.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	RR	0	R/W	Receive Request 0: Receiving function is disabled*1 If 0 is written to this bit, the E-DMAC stops receive operation after DMA transfer of one frame has completed and then clears this bit. The E-DMAC clears this bit when receive descriptor empty occurs. 1: Receive descriptor is read, and the E-DMAC is ready to receive.

Note 1. If the receiving function is disabled during frame reception, write-back is not performed successfully to the receive descriptor. Following pointers to read a receive descriptor become abnormal and the E-DMAC cannot operate successfully. In this case, to make E-DMAC reception enabled again, execute a software reset by the SWRT and SWRR bits in EDMR0. To disable the E-DMAC receiving function without executing a software reset, specify the RE bit in ECMR0. Next, after the E-DMAC has completed the reception and write-back to the receive descriptor has been confirmed, disable the receiving function using this register.

chanhvinh.com

26.3.39 Transmit Descriptor List Start Address Register (TDLAR)

TDLAR is a 32-bit readable/writable register that specifies the start address of the transmit descriptor list. Descriptors have a boundary configuration in accordance with the descriptor length indicated by the DL bits in EDMR. This register must not be modified during transmission. Modifications to this register should only be made in the transmission-halted state specified by bits TR[1:0] (= 00) in the E-DMAC transmit request register (EDTRR).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TDLA[31:15]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDLA[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TDLA[31:0]	All 0	R/W	Transmit Descriptor Start Address The lower bits are set according to the specified descriptor length. 16-byte boundary: TDLA[3:0] = 0000 32-byte boundary: TDLA[4:0] = 00000 64-byte boundary: TDLA[5:0] = 000000

26.3.40 Receive Descriptor List Start Address Register (RDLAR)

RDLAR is a 32-bit readable/writable register that specifies the start address of the receive descriptor list. Descriptors have a boundary configuration in accordance with the descriptor length indicated by the DL bits in EDMR. This register must not be modified during reception. Modifications to this register should only be made while reception is disabled by the RR bit (= 0) in the E-DMAC receive request register (EDRRR).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDLA[31:15]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDLA[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RDLA[31:0]	All 0	R/W	Receive Descriptor Start Address The lower bits are set according to the specified descriptor length. 16-byte boundary: RDLA[3:0] = 0000 32-byte boundary: RDLA[4:0] = 00000 64-byte boundary: RDLA[5:0] = 000000

chanhvinh.com

26.3.41 E-MAC/E-DMAC Status Register (EESR)

EESR is a 32-bit readable/writable register that shows communications status information on the E-DMAC in combination with the E-MAC. The information in this register is reported in the form of interrupt sources. Individual bits are cleared by writing 1 (however, bit 22 (ECI) is a read-only bit that is not cleared by writing 1) and are not affected by writing 0. Each interrupt source can also be masked by means of the corresponding bit in the E-MAC/E-DMAC status interrupt permission register (EESIPR).

The interrupt generated by this status register is ETHERI. For interrupt priorities, see section 7.4, Interrupt Sources.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TWB[1:0]		TC[1]	TUC	ROC	TABT	RABT	RFCOF	—	ECI	TC[0]	TDE	TFUF	FR	RDE	RFOF
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RMAF	—	—	RRF	RTLF	RTSF	PRE	CERF
Initial value:	0	0	0	0	0	Undefined	Undefined	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	TWB[1:0]	00	R/W	Write-Back Complete Indicates that write-back from the E-DMAC to the corresponding descriptor after frame transmission has completed. This operation is enabled only when the TWBI bit in the transmit descriptor that includes the end of the transmit frame is set to 1. 00: Write-back has not completed, or no transmission directive 11: Write-back has completed Others: Setting disabled
29	TC[1]	0	R/W	Frame Transmission Complete Indicates, in combination with the TC[0] bit, that all the data specified by the transmit descriptor has been transmitted from the E-MAC. This bit is set to 1 on assuming the completion of transmission. This is when transmission of one frame is completed and the transmit descriptor valid bit (TACT) of the next transmit descriptor not being set in single-frame/single-descriptor operation or when the last data of a frame has been transmitted and the transmit descriptor valid bit (TACT) of the next descriptor not being set in multi-buffer frame processing based on single-frame/multi-descriptor operation. After frame transmission has completed, the E-DMAC writes the transmission status back to the relevant descriptor. TC[1:0] 00: Transmission has not completed, or no transmission directive 11: Transmission has completed Others: Setting disabled
28	TUC	0	R/W	Transmit Underflow Frame Write-Back Complete 0: Write-back has not completed for the frame causing transmit underflow 1: Write-back has completed for the frame causing transmit underflow
27	ROC	0	R/W	Receive Overflow Frame Write-Back Complete 0: Write-back has not completed for the frame causing receive overflow 1: Write-back has completed for the frame causing receive overflow
26	TABT	0	R/W	Transmit Abort Detect Indicates that the E-MAC aborts transmitting a frame because of failures during frame transmission. 0: Frame transmission has not been aborted or no transmission directive 1: Frame transmission has been aborted
25	RABT	0	R/W	Receive Abort Detect Indicates that the E-MAC aborts receiving a frame because of failures during frame reception. 0: Frame reception has not been aborted or no reception directive 1: Frame reception has been aborted

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
24	RFCOF	0	R/W	Receive Frame Counter Overflow Indicates that the frame counter in the receive FIFO has overflowed. 0: Receive frame counter has not overflowed 1: Receive frame counter has overflowed
23	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
22	ECI	0	R	E-MAC Status Register Source This bit is a read-only bit. When the source of an ECSR interrupt is cleared, this bit is also cleared. 0: E-MAC status interrupt source has not been detected 1: E-MAC status interrupt source has been detected
21	TC[0]	0	R/W	Frame Transmission Complete Indicates, in combination with the TC[1] bit, that all the data specified by the transmit descriptor has been transmitted from the E-MAC. For details, see the description of the TC[1] bit.
20	TDE	0	R/W	Transmit Descriptor Empty Indicates that the transmit descriptor valid bit (TACT) of a transmit descriptor read by the E-DMAC is not set if the previous descriptor does not represent the end of a frame in multi-buffer frame processing based on single-frame/ multi-descriptor operation. As a result, an incomplete frame may be sent. 0: Transmit descriptor active bit TACT = 1 detected 1: Transmit descriptor active bit TACT = 0 detected When transmit descriptor empty (TDE = 1) occurs, execute a software reset and initiate transmission. In this case, transmission starts from the address that is stored in the transmit descriptor list start address register (TDLAR).
19	TFUF	0	R/W	Transmit FIFO Underflow Indicates that an underflow has occurred in the transmit FIFO during frame transmission. Incomplete data is sent onto the line. 0: Underflow has not occurred 1: Underflow has occurred
18	FR	0	R/W	Frame Reception Indicates that a frame has been received and the receive descriptor has been updated. This bit is set to 1 each time a frame is received. 0: Frame has not been received 1: Frame has been received
17	RDE	0	R/W	Receive Descriptor Empty Indicates that the RACT bit of a receive descriptor read by the E-DMAC for receive DMA operation is cleared to 0 (invalid). When receive descriptor empty (RDE = 1) occurs, reception can be resumed by setting the RACT bit (cleared to 0) of the receive descriptor to 1 and then writing 1 to the RR bit in EDRRR. 0: Receive descriptor active bit RACT = 1 detected 1: Receive descriptor active bit RACT = 0 detected
16	RFOF	0	R/W	Receive FIFO Overflow Indicates that the receive FIFO has overflowed during frame reception. 0: Overflow has not occurred 1: Overflow has occurred
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10, 9	—	Undefined	R	Reserved The write value should always be 0.
8	—	0	R	Reserved The write value should always be 0.
7	RMAF	0	R/W	Receive Multicast Address Frame 0: Multicast address frame has not been received 1: Multicast address frame has been received
6, 5	—	All 0	R	Reserved The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
4	RRF	0	R/W	Receive Residual-Bit Frame 0: Residual-bit frame has not been received 1: Residual-bit frame has been received
3	RTLF	0	R/W	Receive Too-Long Frame Indicates that a frame whose byte size exceeds the upper limit for the receive frame length set by RFLR has been received. 0: Too-long frame has not been received 1: Too-long frame has been received
2	RTSF	0	R/W	Receive Too-Short Frame Indicates that a frame of fewer than 64 bytes has been received. 0: Too-short frame has not been received 1: Too-short frame has been received
1	PRE	0	R/W	PHY-LSI Receive Error 0: PHY-LSI receive error has not been detected 1: PHY-LSI receive error has been detected
0	CERF	0	R/W	CRC Error on Received Frame 0: CRC error has not been detected 1: CRC error has been detected

chanhvinh.com

26.3.42 E-MAC/E-DMAC Status Interrupt Permission Register (EESIPR)

EESIPR is a 32-bit readable/writable register that enables interrupts corresponding to individual bits in the E-MAC/E-DMAC status register (EESR). An interrupt is enabled by writing 1 to the corresponding bit.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TWB1IP	TWB0IP	TC1IP	TUCIP	ROCIP	TABTIP	RABTIP	RFCOFIP	—	ECIIP	TC0IP	TDEIP	TFUFIP	FRIP	RDEIP	RFOFIP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RMAFIP	—	—	RRFIP	RTLFIP	RTSFIP	PREIP	CERFIP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	TWB1IP	0	R/W	Write-Back Complete Interrupt Enable 0: Write-back complete interrupt is disabled 1: Write-back complete interrupt is enabled
30	TWB0IP	0	R/W	Write-Back Complete Interrupt Enable 0: Write-back complete interrupt is disabled 1: Write-back complete interrupt is enabled
29	TC1IP	0	R/W	Frame Transmission Complete Interrupt Enable 0: Frame transmission complete interrupt is disabled 1: Frame transmission complete interrupt is enabled
28	TUCIP	0	R/W	Transmit Underflow Frame Write-Back Complete Interrupt Enable 0: Transmit underflow frame write-back complete interrupt is disabled 1: Transmit underflow frame write-back complete interrupt is enabled
27	ROCIP	0	R/W	Receive Overflow Frame Write-Back Complete Interrupt Enable 0: Receive overflow frame write-back complete interrupt is disabled 1: Receive overflow frame write-back complete interrupt is enabled
26	TABTIP	0	R/W	Transmit Abort Detect Interrupt Enable 0: Transmit abort detect interrupt is disabled 1: Transmit abort detect interrupt is enabled
25	RABTIP	0	R/W	Receive Abort Detect Interrupt Enable 0: Receive abort detect interrupt is disabled 1: Receive abort detect interrupt is enabled
24	RFCOFIP	0	R/W	Receive Frame Counter Overflow Interrupt Enable 0: Receive frame counter overflow interrupt is disabled 1: Receive frame counter overflow interrupt is enabled
23	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
22	ECIIP	0	R/W	E-MAC Status Register Source Interrupt Enable 0: E-MAC status interrupt is disabled 1: E-MAC status interrupt is enabled
21	TC0IP	0	R/W	Frame Transmission Complete Interrupt Enable 0: Frame transmission complete interrupt is disabled 1: Frame transmission complete interrupt is enabled
20	TDEIP	0	R/W	Transmit Descriptor Empty Interrupt Enable 0: Transmit descriptor empty interrupt is disabled 1: Transmit descriptor empty interrupt is enabled
19	TFUFIP	0	R/W	Transmit FIFO Underflow Interrupt Enable 0: Underflow interrupt is disabled 1: Underflow interrupt is enabled
18	FRIP	0	R/W	Frame Reception Interrupt Enable 0: Frame reception interrupt is disabled 1: Frame reception interrupt is enabled

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
17	RDEIP	0	R/W	Receive Descriptor Empty Interrupt Enable 0: Receive descriptor empty interrupt is disabled 1: Receive descriptor empty interrupt is enabled
16	RFOFIP	0	R/W	Receive FIFO Overflow Interrupt Enable 0: Overflow interrupt is disabled 1: Overflow interrupt is enabled
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	RMAFIP	0	R/W	Receive Multicast Address Frame Interrupt Enable 0: Receive multicast address frame interrupt is disabled 1: Receive multicast address frame interrupt is enabled
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	RRFIP	0	R/W	Receive Residual-Bit Frame Interrupt Enable 0: Receive residual-bit frame interrupt is disabled 1: Receive residual-bit frame interrupt is enabled
3	RTLFIP	0	R/W	Receive Too-Long Frame Interrupt Enable 0: Receive too-long frame interrupt is disabled 1: Receive too-long frame interrupt is enabled
2	RTSFIP	0	R/W	Receive Too-Short Frame Interrupt Enable 0: Receive too-short frame interrupt is disabled 1: Receive too-short frame interrupt is enabled
1	PREIP	0	R/W	PHY-LSI Receive Error Interrupt Enable 0: PHY-LSI receive error interrupt is disabled 1: PHY-LSI receive error interrupt is enabled
0	CERFIP	0	R/W	CRC Error on Received Frame Interrupt Enable 0: CRC error interrupt is disabled 1: CRC error interrupt is enabled

chanhvinh.com

26.3.43 Transmit/Receive Status Copy Enable Register (TRSCER)

TRSCER specifies whether the information for the transmit and receive state reported by bits 26, 25, and 10 to 0 in the E-MAC/E-DMAC status register (EESR) is to be reflected in the TFE or RFE bit of the corresponding descriptor. The bits in this register correspond to bits 26, 25, and 10 to 0 in EESR. When a bit is cleared to 0, the transmit status (bits 26 and 10 to 8 in EESR) is reflected in the TFE bit of the transmit descriptor, and the receive status (bits 25 and 7 to 0 in EESR) is reflected in the RFE bit of the receive descriptor. In this case, the state of a status bit set to 1 is reflected as the TFE or RFE bit set to 1. When a bit is set to 1, the occurrence of the corresponding source is not reflected in the descriptor. After this LSI is reset, all bits are cleared to 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TABT CE	RABT CE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RMAF CE	—	—	RRF CE	RTLF CE	RTSF CE	PRE CE	CERF CE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17	TABTCE	0	R/W	TABT Bit Copy Directive 0: Reflects the TABT bit status in the TFE bit of the transmit descriptor 1: Occurrence of the corresponding source is not reflected in the TFE bit of the transmit descriptor
16	RABTCE	0	R/W	RABT Bit Copy Directive 0: Reflects the RABT bit status in the RFE bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFE bit of the receive descriptor
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	RMAFCE	0	R/W	RMAF Bit Copy Directive 0: Reflects the RMAF bit status in the RFE bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFE bit of the receive descriptor
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	RRFCE	0	R/W	RRF Bit Copy Directive 0: Reflects the RRF bit status in the RFE bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFE bit of the receive descriptor
3	RTLFCE	0	R/W	RTLF Bit Copy Directive 0: Reflects the RTLF bit status in the RFE bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFE bit of the receive descriptor
2	RTSFCE	0	R/W	RTSF Bit Copy Directive 0: Reflects the RTSF bit status in the RFE bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFE bit of the receive descriptor
1	PRECE	0	R/W	PRE Bit Copy Directive 0: Reflects the PRE bit status in the RFE bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFE bit of the receive descriptor

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
0	CERFCE	0	R/W	CERF Bit Copy Directive 0: Reflects the CERF bit status in the RFE bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFE bit of the receive descriptor

26.3.44 Receive Missed-Frame Counter Register (RMFCR)

RMFCR is a 16-bit counter that indicates the number of frames that could not be saved in the receive buffer and so were discarded during reception. When the receive FIFO overflows, the receive frames in the FIFO are discarded. The number of frames discarded at this time is counted. When the value in this register reaches H'0000FFFF, count-up is halted. Clear the counter by writing H'00000000 in this register. Note that a value other than H'00000000 must not be written to this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MFC[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	MFC[15:0]	All 0	R/W	Missed-Frame Counter These bits indicate the number of frames that are discarded and not transferred to the receive buffer during reception.

chanhvinh.com

26.3.45 Transmit FIFO Threshold Register (TFTR)

TFTR is a 32-bit readable/writable register that specifies the transmit FIFO threshold at which the first transmission is started. The actual threshold is 4 times the set value. The E-MAC starts transmission when the amount of data in the transmit FIFO exceeds the number of bytes specified by this register, when the transmit FIFO is full, or when one frame of data write is performed. This register must not be written to during transmission (bits TR[1:0] in EDTRR = 11).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TFT[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	TFT[10:0]	All 0	R/W	Transmit FIFO Threshold A value in 32-byte units and smaller than the FIFO size specified by FDR must be set as the transmit FIFO threshold. H'000: Store and forward modes H'008: 32 bytes H'010: 64 bytes H'018: 128 bytes : : H'07F: 508 bytes H'080: 512 bytes : : H'0FF: 1,020 bytes H'100: 1,024 bytes : : H'1FF: 2,044 bytes H'200: 2,048 bytes

Note: • When starting transmission before one frame of data write has completed, take care no underflow occurs.

chanhvinh.com

26.3.46 FIFO Depth Register (FDR)

FDR is a 32-bit readable/writable register that specifies the sizes of the transmit and receive FIFOs.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TFD[2:0]			—	—	—	RFD[4:0]				
Initial value:	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 8	TFD[2:0]	All 1	R/W	Transmit FIFO Size Specifies 256 bytes to 2 Kbytes in 256-byte units as the size of the transmit FIFO whose maximum size is 2 Kbytes. The setting must not be changed after transmission/reception has started. H'00: 256 bytes H'01: 512 bytes : : H'07: 2048 bytes
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4 to 0	RFD[4:0]	All 1	R/W	Receive FIFO Size Specifies 256 bytes to 4 Kbytes in 256-byte units as the size of the receive FIFO whose maximum size is 4 Kbytes. The setting must not be changed after transmission/reception has started. H'00: 256 bytes H'01: 512 bytes : : H'0F: 4096 bytes

chanhvinh.com

26.3.47 Receiving Method Control Register (RMCR)

RMCR is a 32-bit readable/writable register that specifies the control method for the RE bit in ECMR while a frame is received. This register must be set during the receiving-halted state.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RNC
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	RNC	0	R/W	Receive Enable Control Sets whether to continue frame reception. 0: Upon completion of reception of one frame, the E-DMAC writes the receive status to the descriptor and clears the RR bit in EDRRR to 0. 1: Upon completion of reception of one frame, the E-DMAC writes (writes back) the receive status to the descriptor. In addition, the E-DMAC reads the next descriptor and prepares for reception of the next frame.

26.3.48 Receive Descriptor Fetch Address Register (RDFAR)

RDFAR stores the descriptor start address that is required when the E-DMAC fetches descriptor information from the receive descriptor. Which receive descriptor information is used for processing by the E-DMAC can be recognized by monitoring addresses displayed in this register. The address from which the E-DMAC is actually fetching a descriptor may be different from the value read from this register. In the initial setting, set the address of the receive descriptor at which receive processing is to be started.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDFA[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDFA[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RDFA[31:0]	All 0	R/W	Receive Descriptor Fetch Address Writing to these bits during the reception is prohibited.

chanhvinh.com

26.3.49 Receive Descriptor Finished Address Register (RDFXR)

RDFXR stores the start address of the receive descriptor for which the E-DMAC has just completed the write-back processing. Up to which receive descriptor has been processed by the E-DMAC can be recognized by monitoring addresses displayed in this register. In the initial setting, set the address of the descriptor immediately before the descriptor that is pointed to by the address in RDFAR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDFX[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDFX[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RDFX[31:0]	All 0	R/W	Receive Descriptor Finished Address Writing to these bits during the reception is prohibited.

26.3.50 Receive Descriptor Final Flag Register (RDFFR)

RDFFR indicates whether the receive descriptor for which the E-DMAC has just completed the write-back processing and whose start address is stored in RDFXR is at the end of the receive descriptor queue (descriptor list).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RDLF
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	RDLF	0	R/W	Receive Descriptor Queue Last Flag Indicates whether the receive descriptor for which the E-DMAC has just completed the write-back processing and whose start address is stored in RDFXR is at the end of the receive descriptor queue (descriptor list). 0: Not the last descriptor in the receive descriptor queue 1: Last descriptor in the receive descriptor queue

chanhvinh.com

26.3.51 Transmit Descriptor Fetch Address Register (TDFAR)

TDFAR stores the descriptor start address that is required when the E-DMAC fetches descriptor information from the transmit descriptor. Which transmit descriptor information is used for processing by the E-DMAC can be recognized by monitoring addresses displayed in this register. The address from which the E-DMAC is actually fetching a descriptor may be different from the value read from this register. In the initial setting, set the address of the transmit descriptor at which transmit processing is to be started.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TDFA[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDFA[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TDFA[31:0]	All 0	R/W	Transmit Descriptor Fetch Address Writing to these bits during transmission is prohibited.

26.3.52 Transmit Descriptor Finished Address Register (TDFXR)

TDFXR stores the start address of the transmit descriptor for which the E-DMAC has just completed the write-back processing. Up to which transmit descriptor has been processed by the E-DMAC can be recognized by monitoring addresses displayed in this register. In the initial setting, set the address of the transmit descriptor immediately before the descriptor that is pointed to by the address in TDFAR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TDFX[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDFX[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TDFX[31:0]	All 0	R/W	Transmit Descriptor Finished Address Writing to these bits during transmission is prohibited.

chanhvinh.com

26.3.53 Transmit Descriptor Final Flag Register (TDFFR)

TDFFR indicates whether the transmit descriptor for which the E-DMAC has just completed the write-back processing and whose start address is stored in TDFXR is at the end of the transmit descriptor queue (descriptor list).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TDLF
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TDLF	0	R/W	Transmit Descriptor Queue Last Flag Indicates whether the transmit descriptor for which the E-DMAC has just completed the write-back processing and whose start address is stored in TDFXR is at the end of the transmit descriptor queue (descriptor list). 0: Not the last descriptor in the transmit descriptor queue 1: Last descriptor in the transmit descriptor queue

chanhvinh.com

26.3.54 Overflow Alert FIFO Threshold Register (FCFTR)

FCFTR is a 32-bit readable/writable register that sets the flow control of the E-MAC. The threshold can be set by the size of the receive FIFO data (bits RFD[7:0]) and the number of receive frames (bits RFF[4:0]).

If the same receive FIFO size as set by the FIFO depth register (FDR) is set when flow control is turned on according to the RFD setting condition, flow control is turned on with (FIFO data size – 64) bytes. For instance, when the RFD bits in FDR is 7 and the RFD bits in this register is 7, flow control is turned on when (2,048 – 64) bytes of data is stored in the receive FIFO. The value set in the RFD bits in this register should be equal to or less than that set in the RFD bits in FDR. Flow control is turned on when either of the setting conditions of bits RFF[4:0] and bits RFD[7:0] is satisfied. Flow control is turned off when neither of the conditions is satisfied (release).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	RFF[4:0]				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

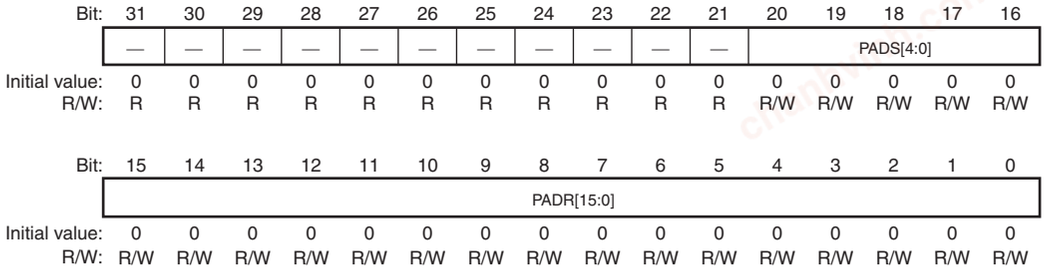
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RFD[7:0]							
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20 to 16	RFF[4:0]	H'17	R/W	Receive FIFO Overflow Alert Signal Output Threshold H'00: When one receive frame has been stored in the receive FIFO H'01: When two receive frames have been stored in the receive FIFO : : H'16: When 23 receive frames have been stored in the receive FIFO H'17: When 24 receive frames have been stored in the receive FIFO
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	RFD[7:0]	H'FF	R/W	Receive FIFO Overflow Alert Signal Output Threshold H'00: When (256 – 32) bytes of data is stored in the receive FIFO H'01: When (512 – 32) bytes of data is stored in the receive FIFO : : H'06: When (1,792 – 32) bytes of data is stored in the receive FIFO H'07: When (2,048 – 64) bytes of data is stored in the receive FIFO

chanhvinh.com

26.3.55 Receive Data Padding Insert Register (RPADIR)

RPADIR is a 32-bit readable/writable register that inserts padding in receive data. To change the settings of this register, execute a software reset by means of the SWRT and SWRR bits in the E-DMAC mode register (EDMR) before making settings again.



Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20 to 16	PADS[4:0]	H'00	R/W	Padding Size H'00: No padding insertion H'01: 1-byte insertion : : H'1F: 31-byte insertion
15 to 0	PADR[15:0]	H'0000	R/W	Padding Slot H'0000: Inserts specified size of padding at the first byte H'0001: Inserts specified size of padding at the second byte : : H'FFFF: Inserts specified size of padding at the 64K byte

chanhvinh.com

26.3.56 Intelligent Checksum Mode Register (CSMR)

CSMR is a readable 32-bit register that specifies the intelligent checksum operation mode. This register must be set while reception is halted.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CSEBL	CSMD	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	SB[5:0]					
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	CSELB	1	R	Intelligent Checksum Calculation Operation Setting 0: The result of checksum calculation is not written back to the receive descriptor. 1: The result of checksum calculation is written back to the receive descriptor.
30	CSMD	1	R/W	Intelligent Checksum Calculation Mode Setting 0: After having skipped the number of bytes specified in SB[5:0], counting from the beginning of the MAC-layer packet, the checksum is calculated for all subsequent data 1: MAC- or IP-layer packets are detected and checksums are calculated for upper-layer protocol packets such as TCP or UDP.
29 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5 to 0	SB[5:0]*1	011010	R/W	Intelligent Checksum Calculation Skip Bytes These bits specify the number of bytes to be skipped for checksum calculation, counting from the beginning of the data received in the E-DMAC. When padding bytes are to be added, specify the checksum start position to cover the amount or extent of padding. H'00: 0 bytes (meaning checksum calculation is performed from the beginning of the packet.) H'02: 2 bytes : : H'1A: 26 bytes H'3E: 62 bytes

Note 1. These bits should only be set when CSEL is 1 and CSMD is 0; otherwise, set to H'00.

chanhvinh.com

26.3.57 Intelligent Checksum Skipped Bytes Monitor Register (CSSBM)

CSSBM is a 32-bit readable register that holds the number of bytes that have been skipped in received packets being handled by the E-DMAC. The number of skipped bytes can be monitored through this register. The amount of data received in E-DMAC may not match the number of skipped bytes.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	SBM[5:0]					
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5 to 0	SBM[5:0]	000000	R	Number of Skipped Bytes These bits are read-only. Writing is prohibited. These bits are initialized when the beginning of a packet for reception is detected.

Note: • This register is only valid when CSEL is 1 and CSMD is 0.

26.3.58 Intelligent Checksum Monitor Register (CSSMR)

CSSMR is a 32-bit register that holds the checksum value of received packets being handled by the E-DMAC. The checksum value can be monitored through this register. The amount of data received in E-DMAC may not match the number of skipped bytes.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CS[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5 to 0	CS[15:0]	000000	R	Intelligent Checksum Value These bits are read-only. Writing is prohibited. These bits are initialized when the beginning of a packet for reception is detected.

Note: • This register is only valid when CSEL = 1 and CSMD = 0.

chanhvinh.com

26.4 Operation

The ETHER consists of the following three function units:

- DMA transfer controller (E-DMAC): DMA transfer between the transmit/receive buffer in the memory and the transmit/receive FIFO
- MAC controller (E-MAC): Transmission/reception processing between the transmit/receive FIFO and the MII
- Transfer Switching Unit (TSU): CAM processing

Using its direct memory access (DMA) function, the E-DMAC performs DMA transfer of frame data between a user-specified Ethernet frame transmission/reception data storage destination (accessible memory space: transmit buffer/ receive buffer) and the transmit/receive FIFO in the E-DMAC. The user cannot read and write data from and to the transmit/receive FIFO directly via the CPU.

To enable the E-DMAC to perform DMA transfer, information (data) including a transmit/receive data storage address and so forth, referred to as a descriptor, is required. The E-DMAC reads transmit data from the transmit buffer or writes receive data to the receive buffer according to the descriptor information. By arranging multiple descriptors as a descriptor row (list) (to be placed in a readable/writable memory space), multiple Ethernet frames can be transmitted or received continuously.

The E-MAC constructs an Ethernet frame using the data written to the transmit FIFO and transmits the frame to the MII. It also performs a CRC check of an Ethernet frame received from the MII and deconstructs the frame to write to the receive FIFO. The E-MAC supports the MII format for interface to the PHI-LSI connected externally to this LSI.

The TSU, which is placed between the E-DMAC and E-MAC, references the CAM entry table to select one of the following tasks according to the Ethernet frame destination address (DA) input to the E-MAC.

- Receives data and writes to the receive FIFO.
- Discards data.

chanhvinh.com

Figure 26.2 shows the frame data path and an overview of each setting.

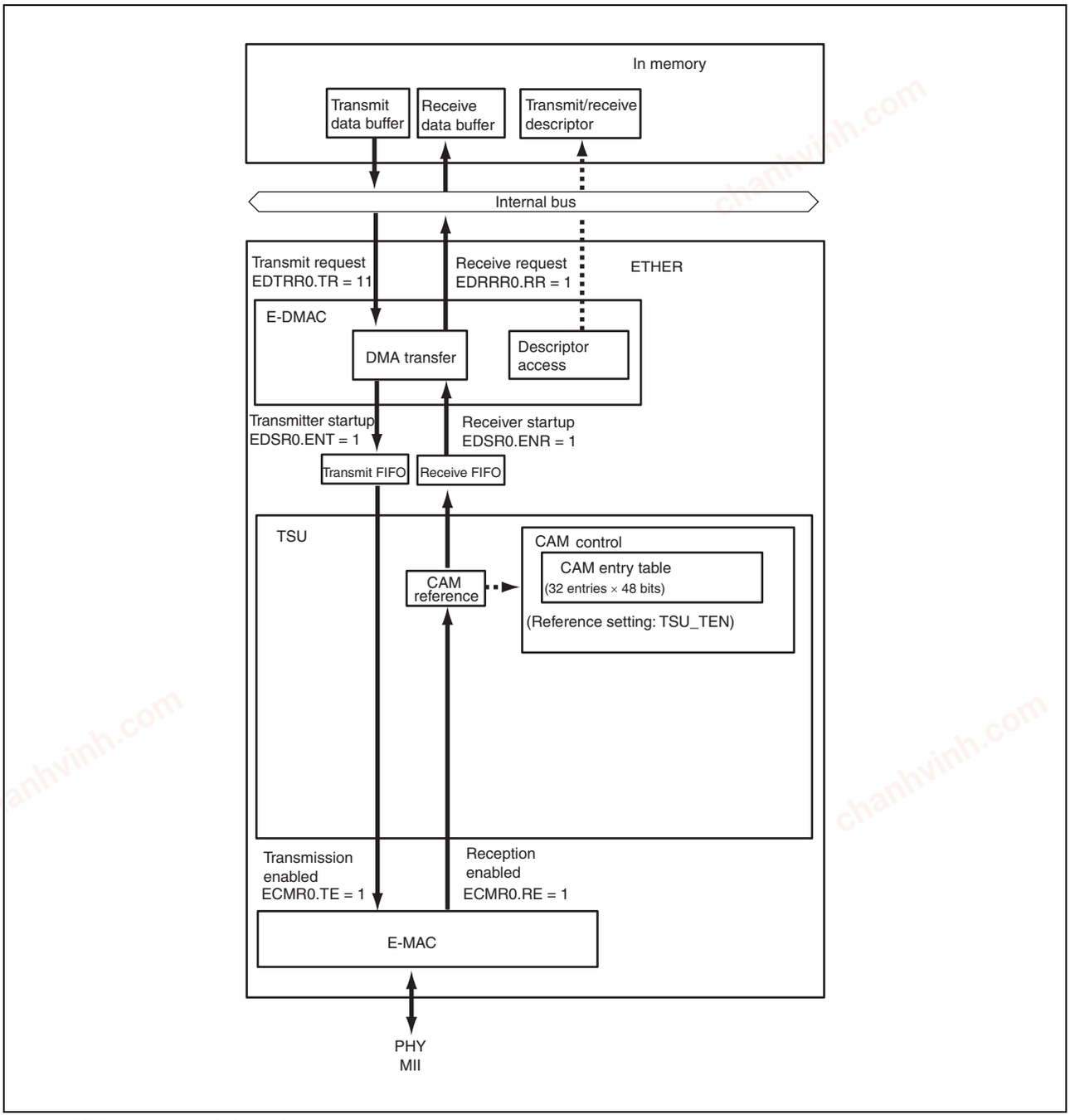


Figure 26.2 ETHER Data Path and Various Settings

chanhvinh.com

26.4.1 Descriptors and Descriptor List

The E-DMAC performs DMA transfer according to the information (data), referred to as a descriptor, written in memory space. There are two types of descriptors: transmit descriptors and receive descriptors. Before a DMA transfer, DMA transfer information including a transmit/receive frame data storage address must be set by software.
The E-DMAC automatically starts reading a transmit/receive descriptor when the TR bits in EDTRR are set to 11 or the RR bit in EDRRR is set to 1, and performs DMA transfer of frame data between the transmit/receive buffer and transmit/receive FIFO according to the information stored in the descriptor. After completion of Ethernet frame transmission/reception, the E-DMAC disables the descriptor valid/invalid bit and reflects the result of transmission/reception in the status bits.
Descriptors are placed in a readable/writable memory space. The address of the start descriptor (descriptor to be read first by the E-DMAC) is set in TDLAR/RDLAR. When multiple descriptors are prepared as a descriptor row (descriptor list), the descriptors are placed in continuous addresses (memory) according to the descriptor length set in the DL0 and DL1 bits in EDMR.

(1) Transmit Descriptor

Figure 26.3 shows the configuration of a transmit descriptor and the relationship with a transmit buffer.
The data of a transmit descriptor consists of TD0, TD1, TD2, and padding data in groups of 32 bits from top to end. The length of padding data is determined according to the descriptor length specified by the DL0 and DL1 bits in EDMR. TD0 indicates whether the transmit descriptor is valid or invalid, and information about the descriptor configuration and status. TD1 indicates the length of data in a transmit buffer to be transferred (TDL) as specified by the descriptor. TD2 indicates the start address of a transmit buffer that holds data to be transferred (TBA).
Depending on the descriptor specification, one transmit descriptor can specify all transmit data of one frame (single-frame/single-buffer) or multiple descriptors can specify the transmit data of one frame (single-frame/multi-buffer). As an example of single-frame/multi-buffer operation, the data portion that is used in a fixed manner in each Ethernet frame transmission can be referenced by multiple descriptors. For example, multiple descriptors can share the destination address and transmit source address in an Ethernet frame, and the remaining data can be stored in each separate buffer.

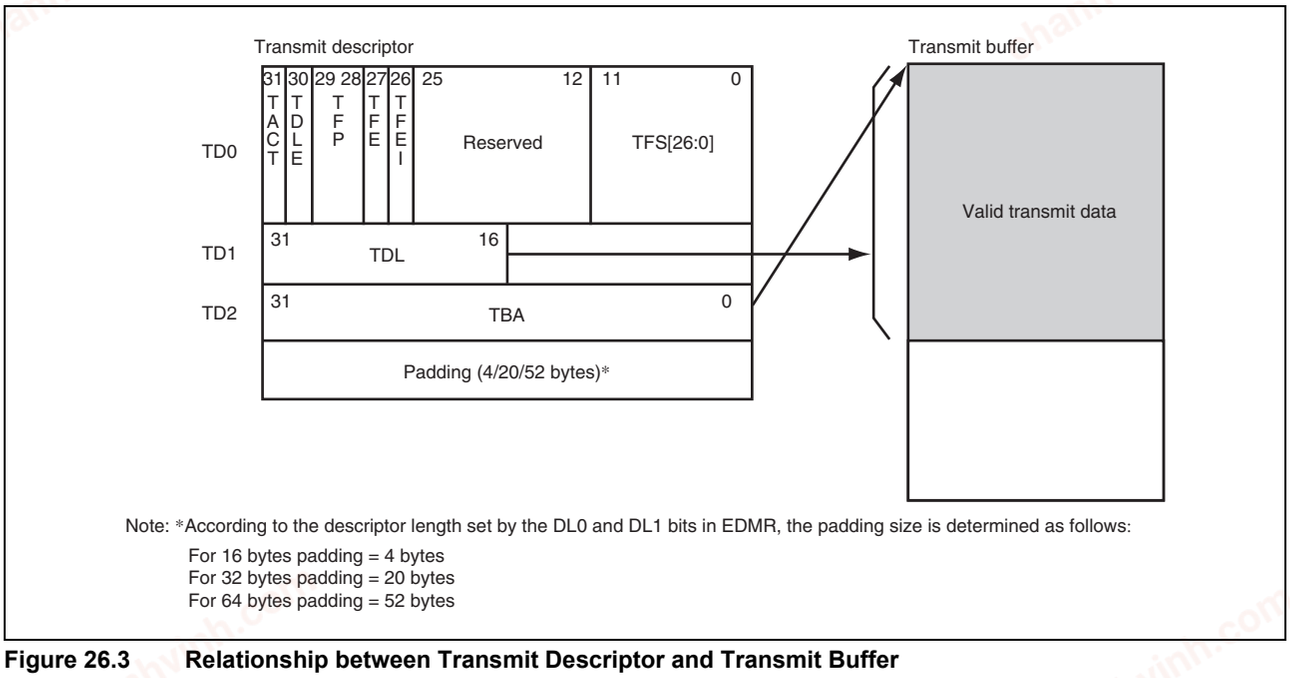


Figure 26.3 Relationship between Transmit Descriptor and Transmit Buffer

chanhvinh.com

(a) Transmit Descriptor 0 (TD0)

Before the TR bits in EDTRR are set to 11, the user sets whether the bits of the descriptor are valid or invalid bit and sets other descriptor configuration. After Ethernet frame transmission, the E-DMAC disables the valid/invalid bits of the descriptor and writes status information. This operation is referred to as write-back.

When using TD0, the user should write desired values to bits 31 to 28 and 26 according to the descriptor configuration. Bits 27 and 25 to 0 should be cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
31	TACT	0	R/W	Transmit Descriptor Valid/Invalid Indicates whether the corresponding descriptor is valid or invalid. To make this bit valid, store transmit data in a transmit buffer (user-specified transmit data storage destination) beforehand, then write 1 to this bit. The E-DMAC clears this bit to 0 after data transfer. 0: Indicates that this transmit descriptor is invalid Indicates the initial setting state, the state after 0 is written, or (in case the user writes 1 to this bit) that this bit is cleared to 0 because the E-DMAC data transfer processing is completed. If this state is recognized when the E-DMAC reads a descriptor, the E-DMAC clears the TR bit in EDTRR to 0, and halts transfer operation related to transmission by the E-DMAC. 1: Indicates that this transmit descriptor is valid After the user writes 1 to this bit, this bit indicates that data is not transferred yet or data is being transferred. When there is a descriptor row (descriptor list) consisting of multiple continuous descriptors, the E-DMAC can continue operation when this bit of the next descriptor is valid.
30	TDLE	0	R/W	Transmit Descriptor List End Indicates whether the corresponding descriptor is the last descriptor of the descriptor row (descriptor list). 0: Not last descriptor After transfer of the corresponding descriptor, the E-DMAC reads the next one in the list of continuous descriptors. 1: Last descriptor After transfer of the corresponding descriptor, the E-DMAC reads the descriptor placed at the address indicated by TDLAR.
29, 28	TFP[1:0]	00	R/W	Transmit Frame Position These bits indicate whether information of this descriptor represents information about the start, middle, or end of the transmit frame. 00: The information of the descriptor represents information about the middle of the frame. 01: The information of the descriptor represents information about the end of the frame. 10: The information of the descriptor represents information about the start of the frame. 11: The information of the descriptor represents all information about the frame (single-frame/single-descriptor (single-buffer)). Reference When one frame is divided for use, the method of specifying this bit for a descriptor row according to the number of divisions is described below. • For single-frame/single-descriptor operation First descriptor: TFP[1:0] = 11 • For single-frame/two-descriptor operation First descriptor: TFP[1:0] = 10 Second descriptor: TFP[1:0] = 01 • For single-frame/three-descriptor operation First descriptor: TFP[1:0] = 10 Second descriptor: TFP[1:0] = 00 Third descriptor: TFP[1:0] = 01 When the number of divisions is large, a descriptor row is configured by adding intermediate descriptors with TFP[1:0] = 00.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
27	TFE	0	R/W	Transmit Frame Error Occurrence Indicates that an error occurred in the transmit frame. 0: The TFS11 to TFS0 bits are all 0 1: One of the TFS11 to TFS0 bits is 1 The TFS8 to TFS0 bits can be masked for each factor by using TRSCER. The TFS11 to TFS9 bits cannot be masked. This bit is set by the E-DMAC write-back operation.
26	TWBI	0	R/W	Write-Back Completion Interrupt Notification 0: Does not notify of a write-back completion interrupt 1: After a write-back operation to this descriptor is complete, this bit sets the TWB1 and TWB0 bits in EESR to 11 and notifies the CPU of a write-back completion interrupt. This bit is valid only for the descriptor including the end of transmit frame (TFP = 01 or 11). This bit is cleared to 0 by the E-DMAC write-back operation.
25 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	TFS [11:0]	All 0	R/W	Transmit Frame Status These bits indicate the status of the corresponding frame. A bit below, which is set by the E-DMAC write-back operation, indicates the occurrence of the corresponding event when set to 1. • TFS[11:10]: Reserved (The write value should always be 0.) • TFS[9]: Transmit FIFO underflow (Corresponding to the TUC bit in EESR) • TFS[8]: Detection of transmission abort (Corresponding to the TABT bit in EESR) • TFS[7:0]: Reserved (The write value should always be 0.)

(b) Transmit Descriptor 1 (TD1)

TD1 indicates the data length of the transmit buffer used by the corresponding descriptor.
The user should set TD1 before the start of a read by the E-DMAC.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	TDL [15:0]	All 0	R/W	Transmit Buffer Data Length (in bytes) These bits indicate the data length of the corresponding transmit buffer in bytes. The specifiable data lengths are from a minimum of 1 (H'0001) byte to a maximum of 64 K - 32 (H'FFE0) bytes.
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

(c) Transmit Descriptor 2 (TD2)

TD2 indicates the start address of the corresponding 32-bit width transmit buffer. An address value should be specified in a 16-byte boundary.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TBA [31:0]	All 0	R/W	Transmit Buffer Start Address These bits set the start address of the corresponding transmit buffer in a 16-byte boundary.

If descriptors are set below, the E-DMAC does not return to normal operation until a system reset is performed.

- TFP (transmit frame position) is not logically correct
  Example: The TFP bits are set to 11 in a descriptor (descriptor A) and the TFP bits are set to 01 in the next descriptor (descriptor B). This specification means that there is no descriptor indicating the start of the transmit frame specified by descriptor B.
- TBL (transmit buffer length) is set to 0

When one transmit frame is divided into three parts or more with transmit descriptors, the E-DMAC performs the following write-back operation:

- A write-back operation is performed for a transmit descriptor including information for the start of the transmit frame (TFP = 10 or 11) and for a transmit descriptor including information for the end of the frame (TFP = 01 or 11).
- A write-back operation is not performed for a transmit descriptor for the middle of the frame (TFP = 00).

However, TFE (transmit frame error occurrence) or TFS (transmit frame status) is written only to a transmit descriptor including information for the end of the frame (TFP = 01 or 11) by a write-back operation.
Before changing a transmit descriptor with the software, make sure that a write-back operation has been performed (TACT = 0) for the transmit descriptor including information for the end of the frame (TFP = 01 or 11) to avoid overwriting (re-setting) an unprocessed transmit descriptor.

chanhvinh.com

(2) Receive Descriptor

Figure 26.4 shows the relationship between a receive descriptor and receive buffer.

The data of a receive descriptor consists of RD0, RD1, RD2, and padding data in groups of 32 bits from top to end. The length of padding data is determined according to the descriptor length specified by the DL0 and DL1 bits in EDMR. RD0 indicates whether the receive descriptor is valid or invalid, and information about descriptor configuration and status. RD1 indicates the length of data that can be received in the receive buffer specified by the descriptor (RBL) and the length of the received frame data (RDL). RD2 indicates the start address of the receive buffer for storing receive data (RBA).

Depending on the descriptor specification, one receive descriptor can specify the storing of all receive data of one frame in a receive buffer (single-frame/single-buffer). All receive frames can be stored in a single buffer if RBL of each descriptor is set to more than 1514 bytes (maximum Ethernet frame length).

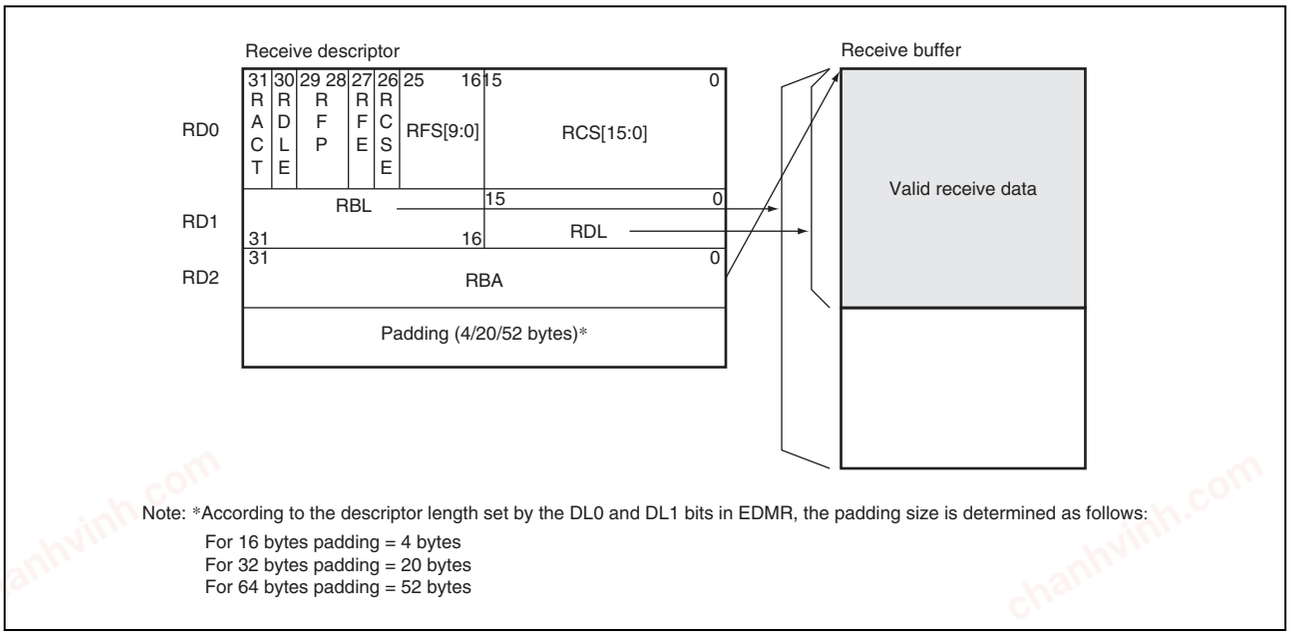


Figure 26.4 Relationship between Receive Descriptor and Receive Buffer

chanhvinh.com

(a) Receive Descriptor 0 (RD0)

The user sets whether the bits of the descriptor are valid or invalid and whether the descriptor represents the end of the descriptor list in RD0 before the RR bit in EDRRR is set to 1 and the start of a read by the E-DMAC. After receive DMA transfer of an Ethernet frame by the E-DMAC, the E-DMAC disables the valid/invalid bits of the descriptor and writes status information. This operation is referred to as write-back.

When using RD0, the user should write desired values to bits 31 and 30 according to the descriptor configuration. Bits 29 to 0 should be cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
31	RACT	0	R/W	Receive Descriptor Valid/Invalid Indicates whether this descriptor is valid or invalid. To make this bit valid, prepare a receive buffer (user-specified receive data storage destination) beforehand, then write 1 to this bit. The E-DMAC clears this bit to 0 after data transfer. 0: Indicates that this receive descriptor is invalid Indicates the initial setting state, the state after 0 is written to, or (in case the user writes 1 to this bit) that this bit is cleared to 0 because the E-DMAC data transfer processing is completed If this state is recognized when the E-DMAC reads a descriptor, the E-DMAC clears the RR bit in EDRRR to 0, and halts transfer operation related to reception by the E-DMAC 1: Indicates that this receive descriptor is valid Indicates that data is not transferred yet after the user writes 1 to this bit, or that data is being transferred When there is a descriptor row (descriptor list) consisting of multiple continuous descriptors, the E-DMAC can continue operation when this bit of the next descriptor is valid
30	RDLE	0	R/W	Receive Descriptor List End Indicates whether this descriptor is the last descriptor of the descriptor row (descriptor list). 0: Not last descriptor After transfer of this descriptor, the E-DMAC reads the next one in the list of continuous descriptors 1: Last descriptor After transfer of this descriptor, the E-DMAC reads the descriptor placed at the address indicated by RDLAR
29, 28	RFP[1:0]	00	R/W	Receive Frame Position 1, 0 The E-DMAC indicates by write-back operation whether information of the corresponding descriptor represents information about the start, middle, or end of the receive frame. 00: The information of the descriptor represents information about the middle of the frame 01: The information of the descriptor represents information about the end of the frame 10: The information of the descriptor represents information about the start of the frame 11: The information of the descriptor represents all information about the frame (single-frame/single-descriptor (single-buffer)) Note: The relationship between a frame after reception of one frame and a descriptor is described below. When the receive buffer data length is less than the receive frame length, these bits are set to a value other than 11 after reception. When they are set so, reception processing should be started after a software reset.
27	RFE	0	R/W	Receive Frame Error Occurrence Indicates that an error occurred in the receive frame. 0: RFS9 to RFS0 are all 0 1: One of RFS9 to RFS0 is 1 Each of RFS8 to RFS0 can be masked by using TRSCER. RFS9 cannot be masked. This bit is set by the E-DMAC write-back operation.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
26	RCSE	0	R/W	Receive Packet Checksum Value Evaluation by Intelligent Checksum When CSEBL = 1 and CSMD = 1, the value of this bit is set as shown in Table 26.3, according to the receive packet and receive data. The information of this bit is invalid when operation is performed with a setting other than above.
25 to 16	RFS[9:0]	All 0	R/W	Receive Frame Status These bits indicate the error status during frame reception. RFS9: Receive FIFO overflow (corresponding to the RFOF bit in EESR) RFS8: Reserved (write value should be 0) RFS7: Multicast address frame received (corresponding to the RMAF bit in EESR) RFS[6:5]: Reserved (write value should be 0) RFS4: Residual-bit frame receive error (corresponding to the RRF bit in EESR) RFS3: Long frame receive error (corresponding to the RTLF bit in EESR) RFS2: Short frame receive error (corresponding to the RTSF bit in EESR) RFS1: PHY-LSI receive error (corresponding to the PRE bit in EESR) RFS0: CRC error on receive frame (corresponding to the CERF bit in EESR)
15 to 0	RCS[15:0]	All 0	R/W	Receive Packet Checksum Value in Intelligent Checksum

Table 26.3 RCSE State Determined by Receive Packet Type and Receive Data

Frame Type		Normal Data		Abnormal Data	
IP Version	Option and Extension Header	RCS[15:0]	RCSE	RCS[15:0]	RCSE
IPv4	None	H'FFFF H'0000	0	Undefined	1
	Fragment	Undefined	Undefined	Undefined	Undefined
	Option	H'FFFF H'0000	0	Undefined	1
IPv6	None	H'FFFF H'0000	0	Undefined	1
	Hop-by-hop	H'FFFF H'0000	0	Undefined	1
	Routing	H'FFFF H'0000	0	Undefined	1
	Destination options	H'FFFF H'0000	0	Undefined	1
	AH	H'FFFF H'0000	0	Undefined	1
	Fragment	Undefined	Undefined	Undefined	Undefined
	ESP	H'0000	1	H'0000	1
	MobileIPv6	H'0000	1	H'0000	1
	Others	H'0000	1	H'0000	1
Other than IPv4 or IPv6		H'0000	0	H'0000	0

chanhvinh.com

(b) Receive Descriptor 1 (RD1)

In RD1, the user specifies the data length of a receive buffer usable by the corresponding descriptor. After reception of a frame, RD1 indicates the length of a frame received by the E-DMAC.
The user should set RD1 before the start of a read by the E-DMAC.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	RBL [15:0]	All 0	R/W	Receive Buffer Data Length (in bytes, to be specified with a 32-byte boundary) These bits set the length of data that can be received by the corresponding receive buffer with an integral multiple of 32 bytes. The specifiable data lengths are from a minimum of 32 (H'0020) bytes to a maximum of 64 K - 32 (H'FFE0) bytes. Set the data length so that a receive frame can be stored in a single buffer. When the checksum function is disabled, any received frame can be stored in a single buffer if the setting is for 1514 bytes (the maximum length of an Ethernet frame) or more. When the checksum function is enabled, any received frame can be stored in a single buffer if the setting is for 1516 bytes (the maximum length of an Ethernet frame + checksum data) or more.
15 to 0	RDL [15:0]	All 0	R	Receive Data Length These bits indicate the data length of a receive frame stored in the receive buffer. Receive data transferred to the receive buffer does not include CRC data (4 bytes) placed at the end of a frame. Accordingly, these bits indicate the number of bytes (valid data bytes), excluding the CRC code, as the data length of the received frame. When the checksum function is enabled, they indicate the number of bytes including the checksum value (2 bytes) as the data length. In single-frame/multi-buffer (descriptor) operation, only the receive data length of the last descriptor is valid. The receive data length of an intermediate descriptor has no meaning. The maximum frame length that can be received is: When padding function is invalid: 64 Kbytes between 1 byte (H'FFFF) When padding function is valid: 64 Kbytes between 32 bytes (H'FFE0)

chanhvinh.com

(c) Receive Descriptor 2 (RD2)

RD2 indicates the start address of the corresponding receive buffer. Set the start address of a receive buffer with a 32-byte boundary.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RBA [31:0]	All 0	R/W	Receive Buffer Start Address These bits set the start address of the corresponding receive buffer with a 32-byte boundary.

The E-DMAC performs DMA transfer for a receive frame from the address specified by RBA (receive buffer address) to the receive buffer in 32-byte units. RBL (receive buffer length) must be set to be an integral multiple of 32 bytes.
If data to be transferred is less than 32 bytes, invalid data will be written to.
[Example]
When the receive frame length is 170 bytes and the required receive buffer capacity is 192 bytes (32 bytes × 6), the sixth DMA-transfer causes invalid data to be written to the receive buffer (In the 32-byte DMA data, the former 10 bytes are valid and the latter 22 bytes are invalid).
Padding of the value 0 can be inserted into only one position in the receive frame by setting RPADIR. The padding size can be selected from 1 byte to 31 bytes in byte units. When padding is inserted into a receive frame, a receive buffer area equal to the total of "receive frame length and padding size" is required. RPADIR setting is valid for all receive frames.
RFE (receive frame error occurrence), PV (padding insertion), RFS (receive frame status) and RFS (receive frame status) are only set in the receive descriptor including information for the end of the frame (TFP = 01 or 11) by a write-back operation.
Before re-setting a receive descriptor with the software, completion of a write-back operation for the receive descriptor (RACT = 0) must be confirmed to avoid rewriting to (and re-setting) an unprocessed receive descriptor.

chanhvinh.com

(3) Descriptor and Transmit/Receive Buffer

(a) Transmission

Each transmit descriptor specifies one transmit buffer. The E-DMAC transfers a transmit frame stored in a transmit buffer specified by a transmit descriptor to the transmit FIFO. Multiple transmit frames stored in transmit buffers specified by multiple descriptors can be connected into one transmit frame and transferred to the transmit FIFO. Figure 26.5 shows the relationship between the transmit descriptors and transmit buffers.

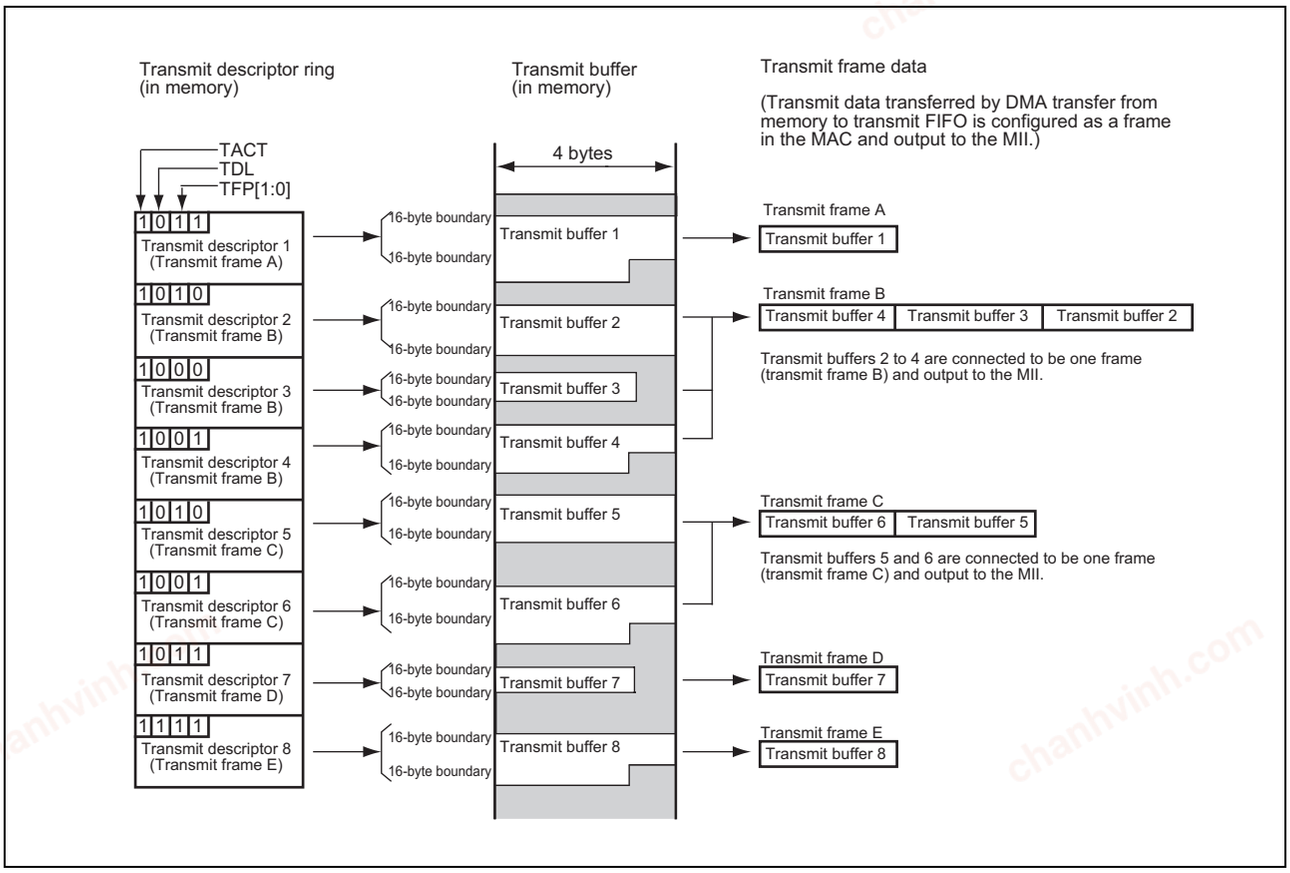


Figure 26.5 Relationship between Transmit Descriptor and Transmit Buffer

chanhvinh.com

(b) Reception

Each receive descriptor specifies one receive buffer. The E-DMAC receives a receive frame from the receive FIFO and stores it in a receive buffer specified by a receive descriptor.

Figure 26.6 shows the relationship between the receive descriptors and receive buffers.

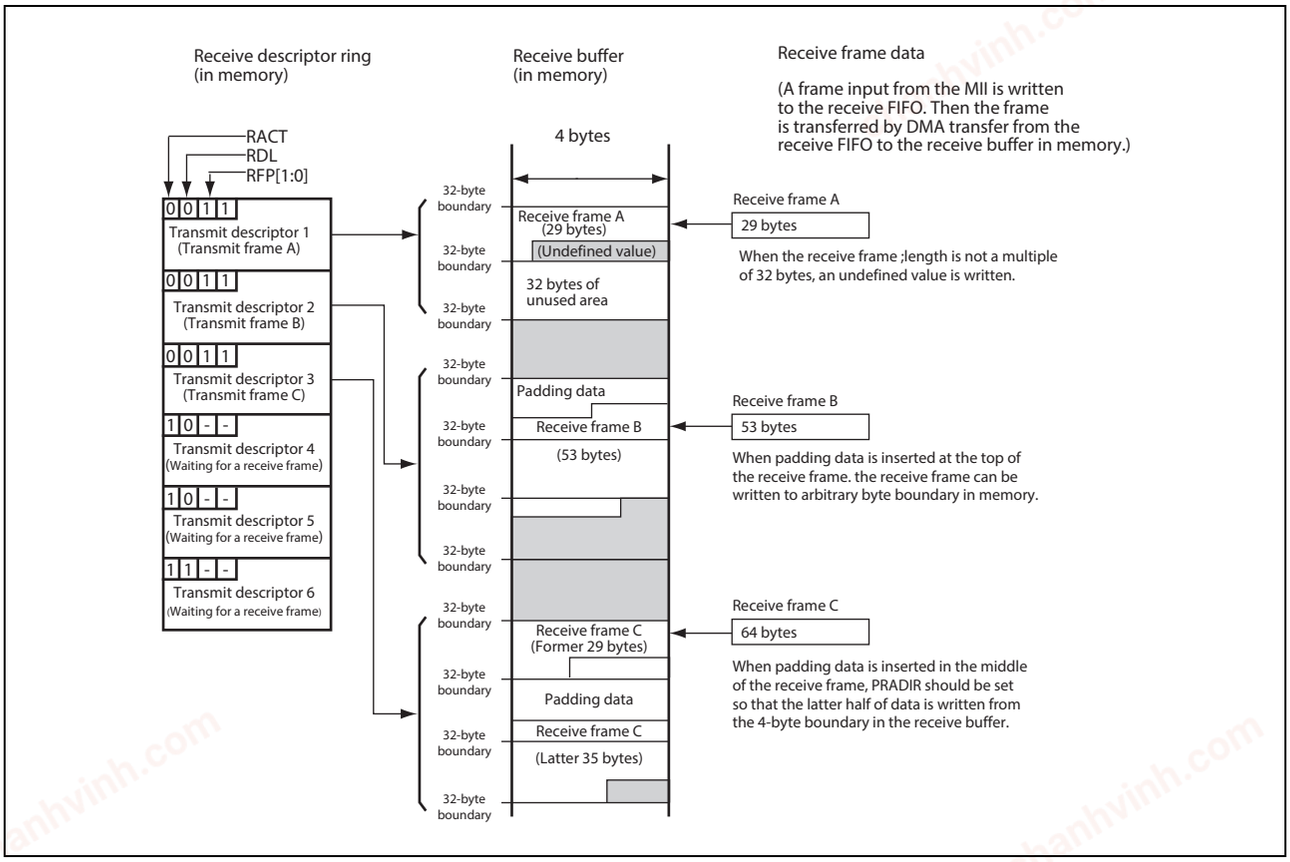


Figure 26.6 Relationship between Receive Descriptor and Receive Buffer

chanhvinh.com

(4) Descriptor Pointer

The E-DMAC controls the transmit and receive descriptor addresses in memory and the processing priority by using the following registers.

1. Registers related to a transmit descriptor

- TDLAR: Address of the start descriptor in a list of transmit descriptors.
- TDFAR: Address of the transmit descriptor to be processed
- TDFXR: Address of the transmit descriptor that finished processing (set by a write-back operation) last
- TDFFR (DL bit): Indicates whether the TDLE value of the transmit descriptor specified by TDFXR is 1 or not.

2. Registers related to receive descriptor:

- RDLAR: Address of the start descriptor in a list of receive descriptors.
- RDFAR: Address of the receive descriptor to be processed
- RDFXR: Address of the receive descriptor that finished processing (set by a write-back operation) last
- RDFFR (DL bit): Indicates whether the RDLE value of the receive descriptor specified by RDFXR is 1 or not.

Transmit descriptors and receive descriptors have a ring structure. When the TDLE (RDLE) value of the processed transmit (receive) descriptor is 0, the next descriptor will be processed. The next descriptor is the transmit (receive) descriptor at the address obtained by adding the processed transmit (receive) descriptor address to the descriptor length specified by the DL bits in EDMR. When the TDLE (RDLE) value of the processed transmit (receive) descriptor is 1, the transmit descriptor indicated by TDLAR (RDLAR) will be processed next. Figure 26.7 shows the relationship between the transmit/receive descriptor ring and read pointer.

The transmit descriptor list must be large enough to point to five or more transmit frames. If four or less transmit frames are pointed to in a list, E-DMAC operation is not guaranteed. Accordingly, do not set that all the transmit descriptors in a ring are used by four or less descriptors. The receive descriptor list does not have this restriction. For example, one receive frame can use all receive descriptors in a list.

In the initial setting, the start address of a descriptor list must be set to TDLAR (RDLAR) and TDFAR (RDFAR), and the end descriptor address of the descriptor list to TDFXR (RDFXR) by the software.

The E-DMAC updates TDFAR (RDFAR), TDFXR (RDFXR), and the DL bit in TDFFR (DL bit in RDFFR) each time a descriptor is processed.

chanhvinh.com

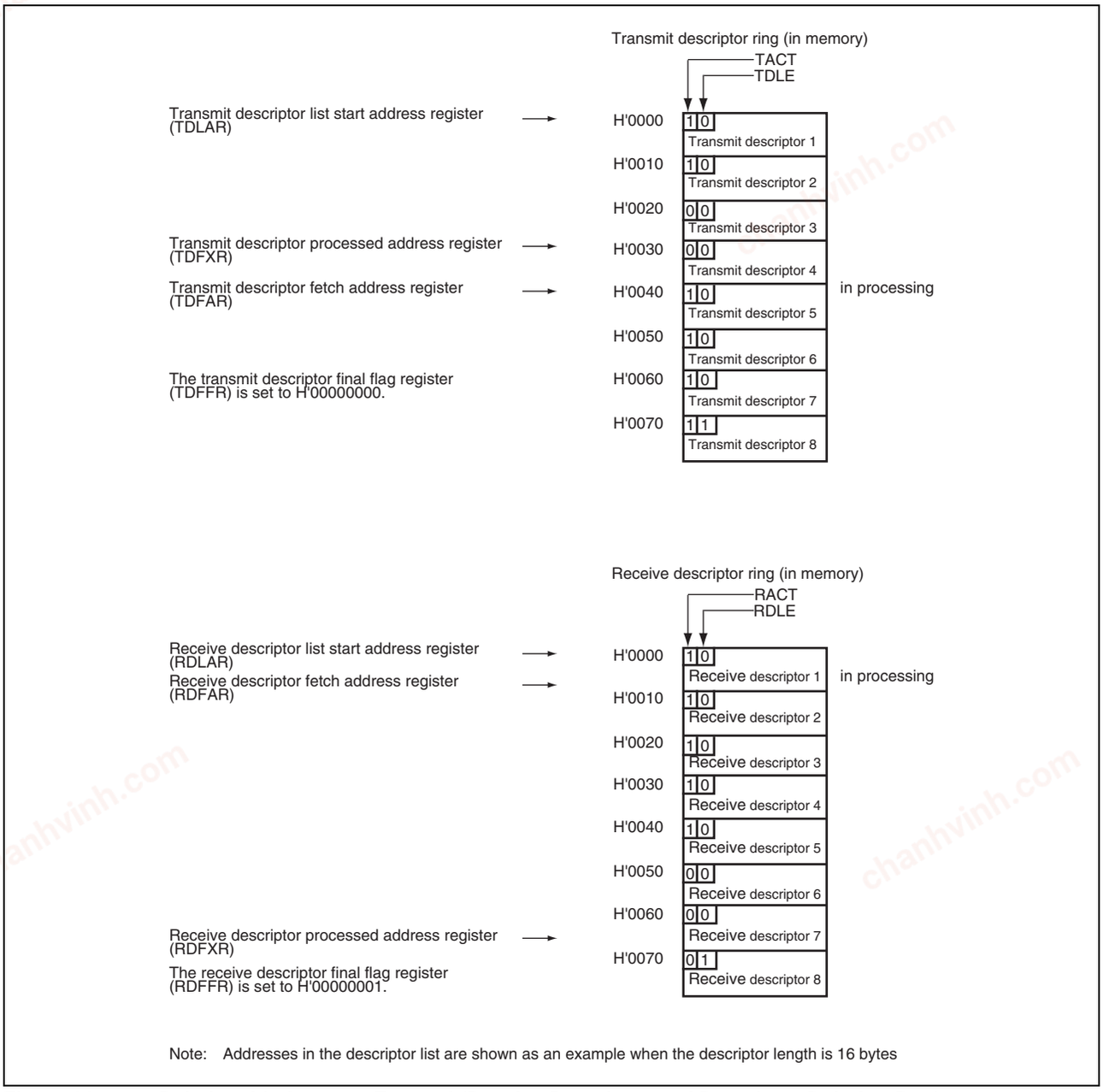


Figure 26.7 Relationship between Transmit/Receive Descriptor and Descriptor Pointing Registers

chanhvinh.com

26.4.2 Transmission

(1) Transmission Procedure and Processing Flow

When 11 is written to the TR bits in EDTRR with the TE bit in ECMR set to 1 and there is empty space of 32 bytes or more in the transmit FIFO, the E-DMAC reads the descriptor following the previously used descriptor from the transmit descriptor list (or the descriptor indicated by TDLAR at the initial startup).
If the TACT bit of the read descriptor is set to 1 (valid), the E-DMAC sequentially reads transmit frame data from the transmit buffer start address specified by TD2 and transfers the data to the transmit FIFO. The E-DMAC configures a transmit frame and starts transmission to the MII. After DMA transfer of data equivalent to the buffer length specified in the descriptor, the following processing is carried out according to the TFP value.

- TFP = 10 (start of a frame)
  Descriptor write-back (writing 0 to the TACT bit) is performed after completion of DMA transfer.
- TFP = 01 or 11 (end of a frame)
  Descriptor write-back (writing 0 to the TACT bit and writing status) is performed after completion of frame transmission.
- TFP = 00 (frame continued)
  Descriptor write-back is not performed. The TACT bit retains the value 1.

As long as the TACT bit of a read descriptor is set to 1 (valid), the reading of E-DMAC descriptors and the transmission of frames continue.
When a descriptor with the TACT bit cleared to 0 (invalid) is read, the E-DMAC performs the following processing and completes transmit processing.

- Clears the TR bits in EDTRR to 00.
- Writes the TC bits in EESR to 11 and generates an interrupt to the CPU.

The E-DMAC can store up to four frames of data in the transmit FIFO.
When the following conditions are satisfied, the E-MAC transmit processing section reads transmit data from the transmit FIFO to configure a frame and transmits the frame to the MII.

- The amount of data in the transmit FIFO exceeds the number of bytes specified by TFTR.
- One or more frame of data is stored in the transmit FIFO.
- The transmit FIFO has no space (full of transmit wait data for the MII).

chanhvinh.com

Figure 26.8 shows an example of transmission flow.

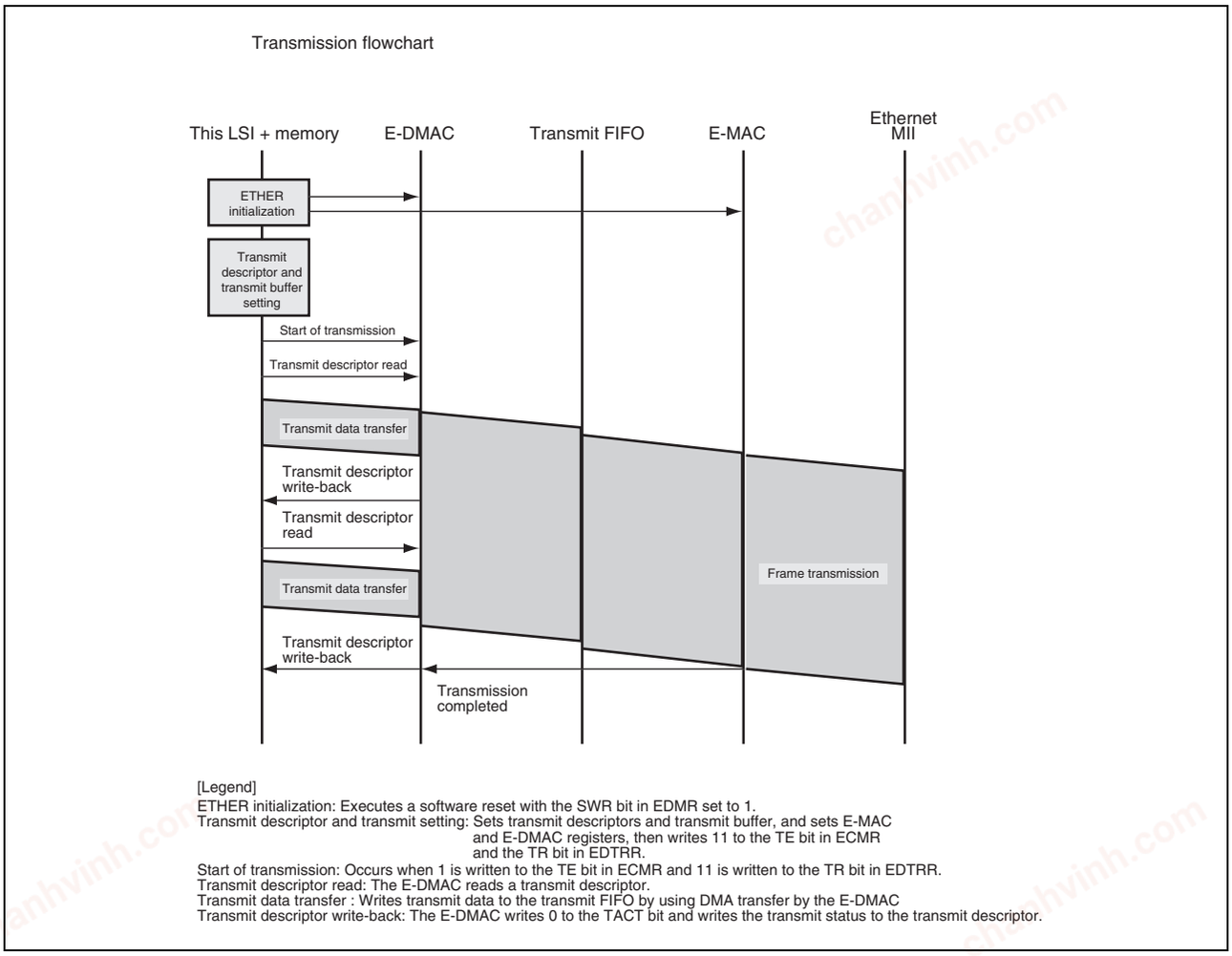


Figure 26.8 Sample Transmission Flowchart (Single-Frame/Two-Description)

Figure 26.9 shows the status change of the E-MAC transmitter.

1. When the TE bit in ECMR is set, the transmitter enters the transmit idle state.
2. The preamble is sent as soon as a transmit request is issued by the E-DMAC.
3. The transmitter sends the SFD, data, and CRC sequentially. At the end of transmission, the transmit E-DMAC generates a transmission complete interrupt (TC).
4. After waiting for the frame interval time, the transmitter enters the idle state, and if there is more transmit data, continues transmitting.

chanhvinh.com

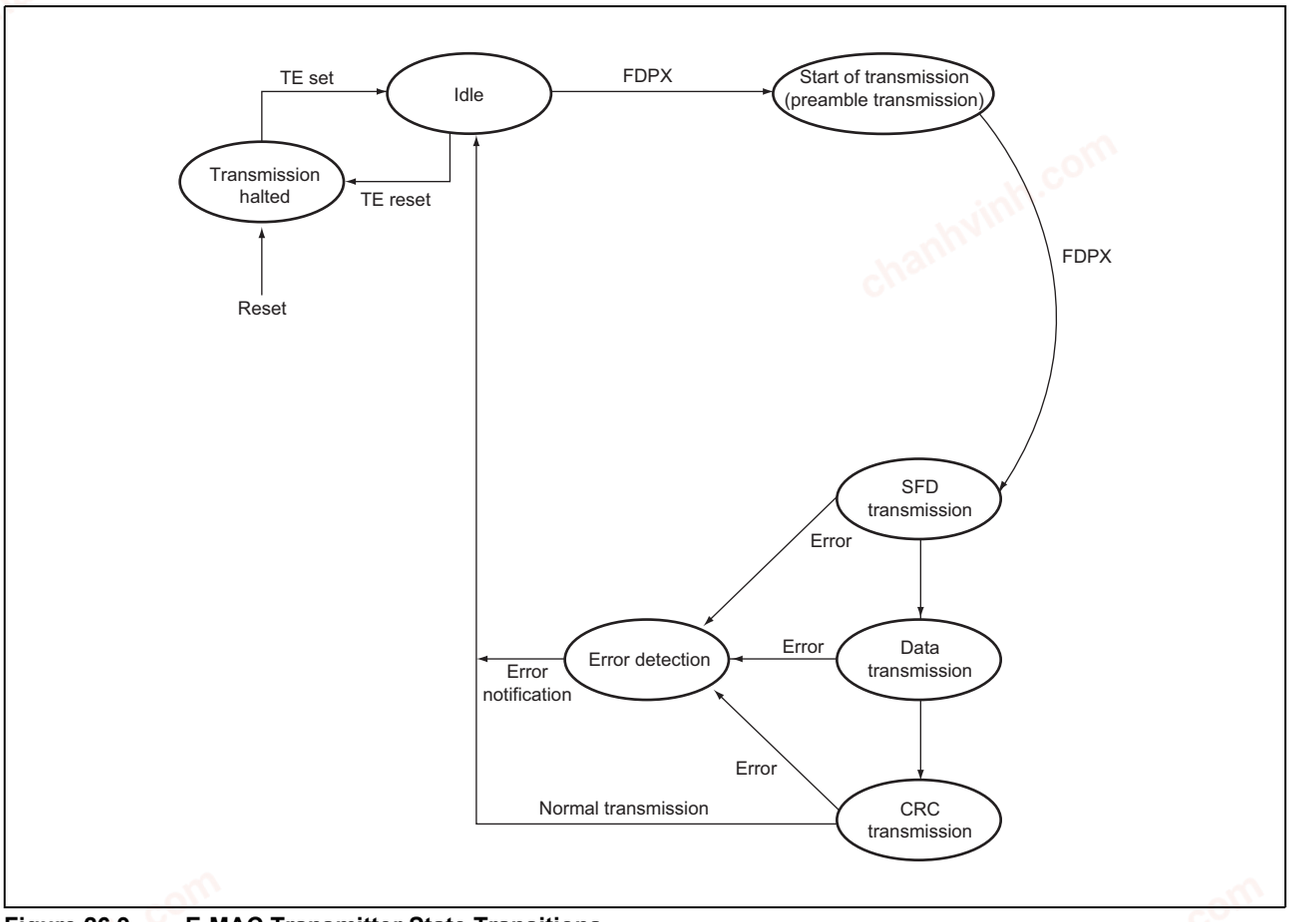


Figure 26.9 E-MAC Transmitter State Transitions

chanhvinh.com

(2) Transmission Error Processing

(a) Transmission Abort

If a transmission error is detected during frame transmission from the transmit FIFO to the MII, transmission of the frame data is aborted. At this time, if DMA transfer of the appropriate frame from the transmit buffer to the transmit FIFO has not been completed, the DMA transfer is also aborted.
Following a write-back operation to the transmit descriptor related to the transmit frame aborted by a transmission error, 1 is written to the TABT bit in EESR and an interrupt is issued to the CPU. The subsequent transmit descriptors will be processed normally.

(b) Transmit FIFO Underflow

If the transmit FIFO is empty (transmit FIFO underflow) during frame transmission from the transmit FIFO to the MII, the E-MAC forcibly aborts transmission of the frame to the MII. At this time, the frame that the E-MAC receives from the E-DMAC is cut off halfway. Then, the E-MAC performs the following operation:

- Writes the TFUF bit in EESR to 1 and generates an interrupt to the CPU.
- Performs a write-back operation to the transmit descriptor corresponding to the transmit frame.
- Following the write-back operation, writes the TUC bit in EESR and generates an interrupt to the CPU.

The subsequent transmit descriptors operate normally.
The E-MAC waits to start frame transmission from the transmit FIFO to the MII until the data that was stored in the transmit FIFO exceeds the number of the bytes specified by TFTR. Through the effective use of TFTR, the transmit FIFO underflow counts can be controlled.

(c) Transmit Descriptor Empty

When the TFP bits of the descriptor previously processed are set to 00 or 10 and the TACT bit of the read transmit descriptor is set to 0 (invalid), a transmit descriptor empty state is determined and 1 is written to the TDE bit in EESR, and then an interrupt is issued to the CPU.
When a transmit descriptor state is empty, start transmission processing after a software reset.

chanhvinh.com

26.4.3 Reception

(1) Reception Procedure and Processing Flow

The E-MAC receiver separates the frame from the MII into preamble, SFD, data and CRC, and transfers the fields from DA (destination address) to the data to the receive FIFO. Up to 24 frames can be written in the receive FIFO. Figure 26.10 shows the status change of the E-MAC receiver.

1. When the RE bit in ECMR is set to 1, the receiver enters the receive idle state.
2. When an SFD (start frame delimiter) is detected after a receive packet preamble, the receiver starts receive processing. A frame with an invalid pattern is discarded.
3. In normal mode, if the destination of the frame address is this LSI, the receiver starts data reception when broadcast or multicast transmission is specified. In promiscuous mode, data reception starts regardless of the frame type.
4. Following data reception from the MII, the receiver carries out a CRC check. The result is indicated as a status bit in the descriptor after the frame data has been written to the receive FIFO. Reports an error status in the case of an abnormality.

After one frame has been received, if the RE bit in ECMR is set to 1, the receiver prepares to receive the next frame.

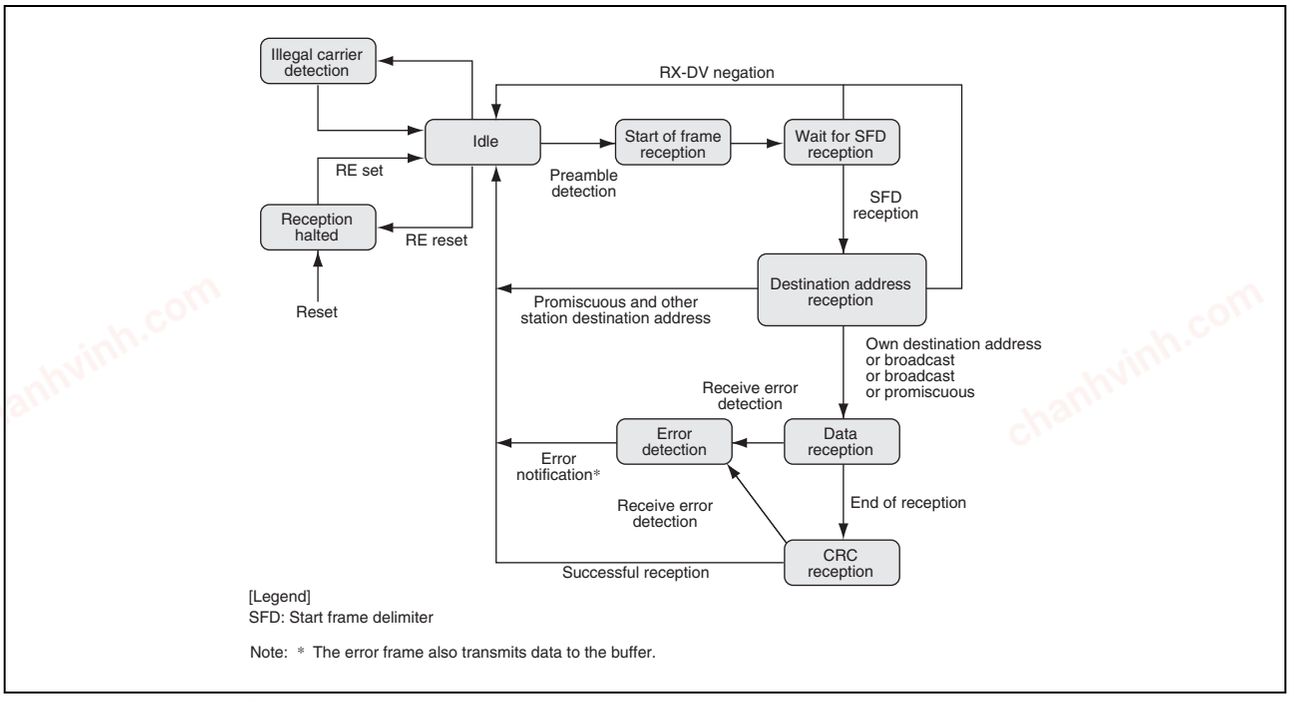


Figure 26.10 E-MAC Receiver State Transitions

CAM evaluation can be referenced during frame processing in reception (for details on the CAM function, refer to section 26.4.4, CAM Function).

When 1 is written to the RR bit in EDRRR while the RE bit in ECMR is set to 1, the E-DMAC reads the descriptor following the previously used descriptor from the receive descriptor list (or the descriptor indicated by RDLAR at the initial startup) then enters the receive wait state. If 32 bytes or more of data or the last byte of the receive frame is stored in the receive FIFO, the E-DMAC transfers receive FIFO data to the receive buffer specified by RD2 according to the receive descriptor with the RACT bit set to 1 (valid).

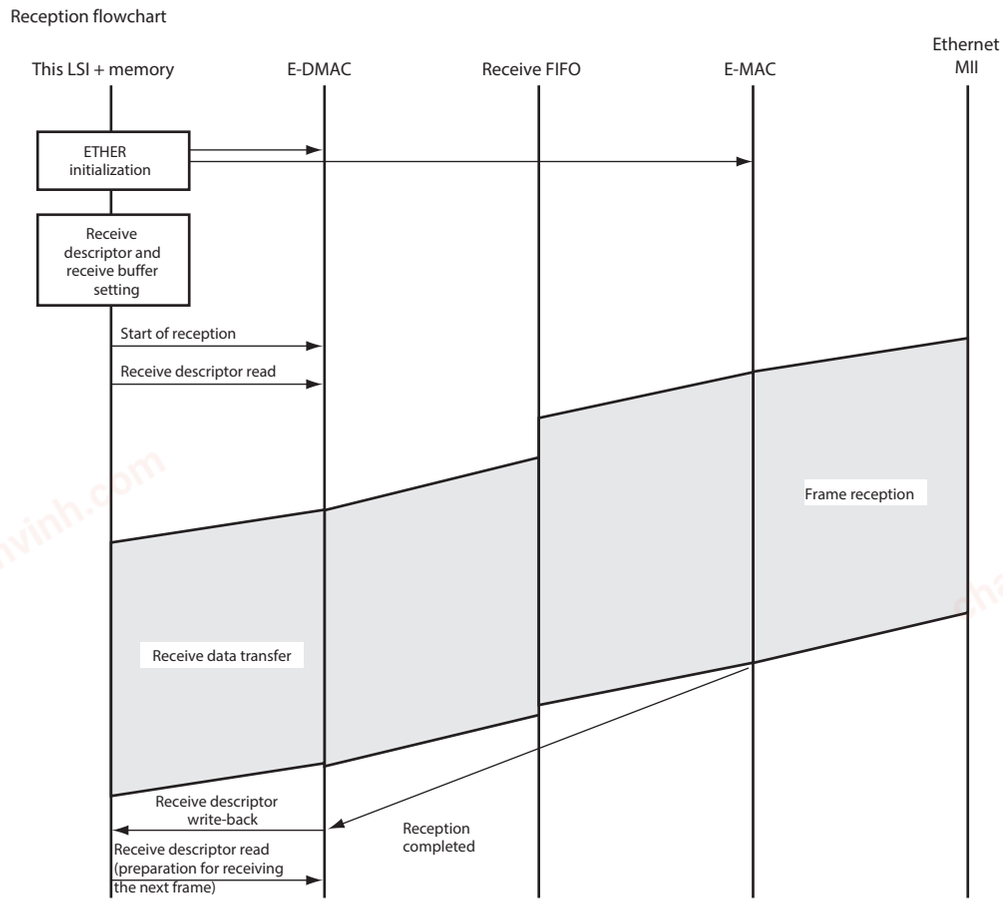
If the data length of a received frame is longer than the buffer length specified by RD1, the E-DMAC performs a write-back operation to the descriptor (set RFP to 10 or 00) when the buffer is full, then reads the next descriptor. The E-DMAC then continues to transfer data to the receive buffer specified by the new RD2.

When the following conditions are satisfied, a write-back operation is performed for the descriptor (RFP = 11 or 01), 11 is written to the FR bits in EESR, and an interrupt is issued to the CPU.

chanhvinh.com

- The receive buffer has been full during DMA transfer.
- DMA transfer to the receive buffer of the last byte of the receive frame has been completed.

After the reception processing of the frame, the next descriptor reading standby state begins. At this time, if 32 bytes or more of data or the last byte of the receive frame is stored in the receive FIFO, the next receive descriptor process is performed continuously.
When the TACT bit of the read receive descriptor is 0 (invalid), the receive descriptor empty state is determined and the RDE bit in EESR is written to 1, and then an interrupt is issued to the CPU.
To receive frames continuously, set the RNC bit in RMCR to 1. The initial value is 0.
Figure 26.11 shows an example of reception flow.



[Legend]

ETHER initialization:	Executes a software reset with the SWR bit in EDMR set to 1.
Receive descriptor and receive buffer setting:	Sets receive descriptors and receive buffers, and sets E-MAC and E-DMAC registers, then writes 1 to the RE bit in ECMR and the RR bit in EDRRR.
Start of Reception:	Occurs when 1 is written to the RE bit in ECMR and the RR bit in EDRRR.
Receive descriptor read:	The E-DMAC reads a receive descriptor.
Receive data transfer:	Writes receive data from the receive FIFO to the receive buffer by using DMA transfer by the E-DMAC.
Receive descriptor write-back:	The E-DMAC writes 0 to the RACT bit and writes the receive status to the receive descriptor

Figure 26.11 Sample Reception Flowchart (Single-Frame/Single-Descriptor)

chanhvinh.com

(2) Reception Error Processing

(a) Reception Error

When a reception error occurs, the FR and RABT bits in EESR are set to 1 and an interrupt is issued to the CPU after a write-back operation for the receive descriptor related to the reception error frame.
If a reception error occurs when the length of the frame received from the MII is less than 32 bytes, DMA transfer to the receive buffer for the frame is not performed. At this time, the receive frame is discarded in the E-DMAC (flush function). However, if padding is inserted in the receive frame by RPADIR, the flush function is performed when the frame length including the padding bytes is less than 32 bytes.

(b) Receive FIFO Overflow

In any of the following cases, the E-MAC cannot receive frames from the MII because it has no space to store receive frames, and all the receive frames that have been transferred to the E-MAC will be discarded in the E-MAC (receive FIFO overflow).

- Receive FIFO is full of data waiting for DMA transfer (the receive FIFO has no space).
- The number of receive frames waiting for DMA transfer is 24 in total (the receive frame information managing area has no empty space; up to 24 frames can be managed).

If an overflow occurs due to the former case, the RFOF bit in EESR is set to 1 and an interrupt is generated to the CPU. If an overflow occurs due to the latter case, the RFCOF bit in EESR is set to 1 and an interrupt is generated to the CPU. Each time a receive frame is discarded due to an overflow, RMFCR is incremented. However, RMFCR is not incremented for a receive frame that is cut off due to insufficient receive FIFO space. If a receive frame is cut off due to insufficient receive FIFO space (the frame is partially stored in the receive FIFO), the E-DMAC performs the following operation:

- Performs DMA transfers for the cut-off frame stored in the receive FIFO to the receive buffer.
- After the DMA transfer, performs a write-back operation on the receive descriptor.
- After the write-back operation, sets the ROC bit in EESR to 1 and generates an interrupt to the CPU.

When the receive FIFO is full of data waiting for DMA transfer, frame reception from the MII can be resumed if DMA transfer is performed from the receive FIFO to the receive buffer and 32 bytes or more of empty space is generated in the receive FIFO. When the number of receive frames waiting for DMA transfer is 24 in total, frame reception from the MII can be resumed if one or more frame has been DMA transferred from the receive FIFO to the receive buffer. For restarting frame reception from the MII, when the E-DMAC resumes frame reception from the MII, it only accepts from the start of the frame.

(c) Flow Control

When the amount of receive data or the number of receive frames in the receive FIFO leads to one of the following conditions, the E-DMAC notifies the E-MAC to control E-MAC writing to the receive FIFO.

- When the space used in the receive FIFO exceeds the data amount specified by FCFTR
- When the number of receive frames in the receive FIFO exceeds the value specified by FCFTR

The threshold of the receive data amount can be set in a range from 256 to 65536 bytes in 256-byte units.
The threshold of receive frames can be set in a range from 1 to 24 frames (by the frame) in frame units.

chanhvinh.com

(d) Receive Descriptor Empty

When the RACT bit of the read descriptor is 0 (invalid), the receive descriptor empty state is determined and DMA transfer is stopped. Then the following operation is performed.

- Writes the RR bit in EDRRR to 0
- Sets the RDE bit in EESR to 1 and generates an interrupt to the CPU.

To resume the DMA transfer to the receive buffer, the interrupt source needs to be cleared by software, the receive descriptor needs to be re-set and the RR bit in EDRRR should be set to 1.
Even if receive descriptor is empty, frame reception from the MII to the receive FIFO is continued if there is empty space left in the receive FIFO and receive frame information management area. Therefore, even if a receive descriptor empty state is determined, the DMA transfer can be performed without discarding the frames received from the MII if DMA transfer to the receive buffer can be resumed before an overflow occurs.

chanhvinh.com

26.4.4 CAM Function

Frames input to the E-MAC are grouped into the following four types; unicast for this LSI, broadcast, multicast, and unicast to other destinations. The MAC addresses of unicast for this LSI and broadcast are fixed, and determined only by register settings. Consequently, only multicast and unicast to other destinations determine whether to receive or not by using the CAM (unicast frames whose destination MAC addresses match this LSI are called unicast frames to this LSI, and those that do not are called unicast frames to other destinations).

Furthermore, the evaluation of reception of unicast to other destinations and multicast frames by using CAM are performed by referencing the registered MAC addresses of the CAM entry table in the TSU. By using this function, receive FIFO overflow can be prevented caused by accumulation of frame data not required for reception, and CPU processing for determining reception can be reduced.

The on-chip CAM has entry tables which can register the MAC address of 32 entries, the details of which can be set by TSU_ADRH0 to TSU_ADRH31 and TSU_ADRL0 to TSU_ADRL31. The setting to enable/disable referencing of the on-chip CAM entry table is performed by the CAM entry table enable setting register which sets whether to perform CAM evaluation or not. When on-chip CAM entry table referencing is enabled, the destination address in the frame and MAC address registered in the CAM entry table are compared, and it is determined whether to transfer the frames input to the E-MAC to E-DMAC (have E-DMAC receive the frames) or discard the frames. Table 26.4 shows the processing method of frames (receive or discard) in reception from E-MAC to E-DMAC.

Table 26.4 Receive Frame Processing

CAM Entry Table Referencing Results	Types of Frame	Normal Mode		Promiscuous Mode	
		MCT = 0	MCT = 1	MCT = 0	MCT = 1
CAM hit (when addresses match)	Frame to this LSI	Discarded		Discarded	
	Broadcast frame	Discarded		Discarded	
	Multicast frame	Discarded	Received	Discarded	Received
	Frames having destinations other than this LSI	Received		Discarded	
CAM mishit (when addresses do not match)	Frames to this LSI	Received		Received	
	Broadcast frame	Received		Received	
	Multicast frame	Received	Discarded	Received	Discarded
	Frames having destinations other than this LSI	Discarded		Received	

[Legend]

MCT (Bit 13 in ECMR): Multicast receive mode (0: Receive when CAM mishit/1: Receive when CAM hit)

chanhvinh.com

26.4.5 Transmit Processing of Multi-Buffer Frame (Single-Frame/Multi-Descriptor)

(1) Multi-Buffer Frame Transmit Processing

If an error occurs during multi-buffer frame transmission, the processing shown in Figure 26.12 is carried out by the E-DMAC.

In the figure where the transmit descriptor is shown as inactive (TACT bit = 0), buffer data has already been transmitted successfully, and where the transmit descriptor is shown as active (TACT bit = 1), buffer data has not been transmitted. If a frame transmit error occurs in the first descriptor part where the transmit descriptor is active (TACT bit = 1), transmission is halted, and the TACT bit is cleared to 0, immediately. The next descriptor is then read, and the position within the transmit frame is determined on the basis of bits TFP1 and TFP0 (continuing [B′00] or end [B′01]). In the case of a continuing descriptor, the TACT bit is cleared to 0, and the next descriptor is read immediately. If the descriptor is the final descriptor, not only is the TACT bit cleared to 0, but write-back is also performed to the TFE and TFS bits at the same time. Data in the buffer is not transmitted between the occurrence of an error and write-back to the final descriptor. If error interrupts are enabled in EESIPR, an interrupt is generated immediately after the final descriptor write-back.

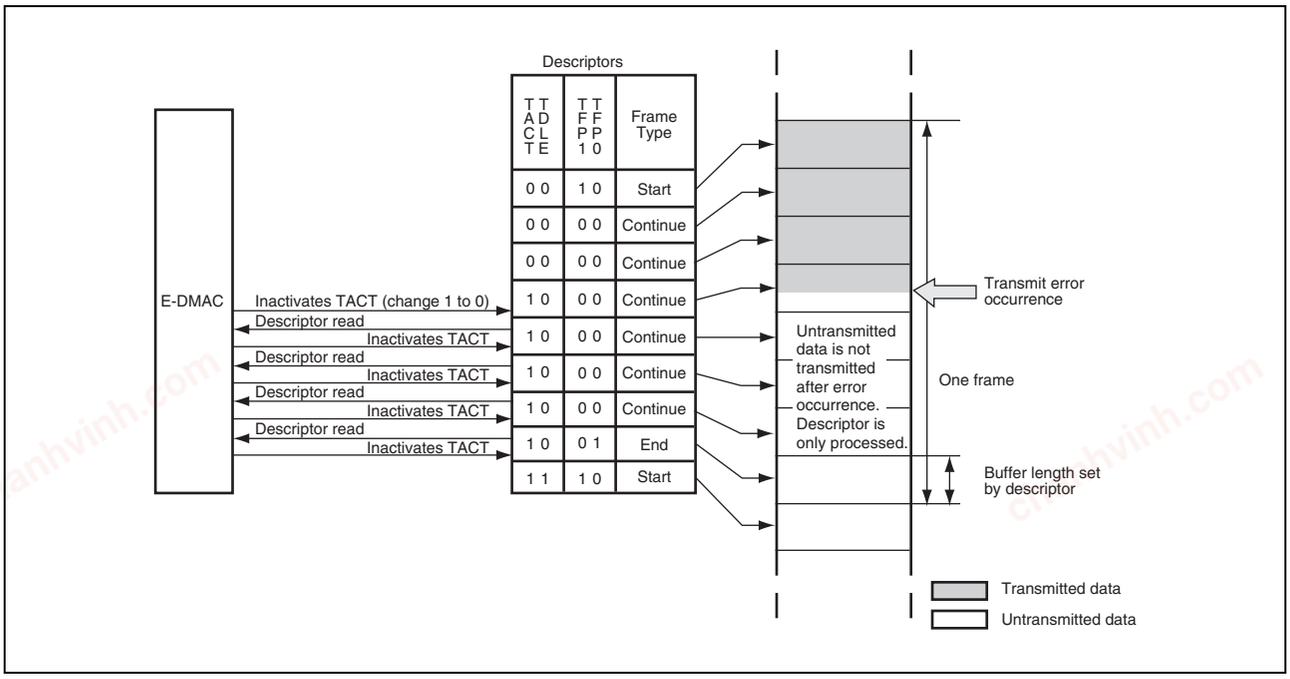


Figure 26.12 E-DMAC Operation after Transmit Error

chanhvinh.com

26.4.6 Padding Insertion in Receive Data

In the E-DMAC, one to three bytes of padding can be inserted in any byte position of receive data to improve software handling capability. By using this function, for instance, inserting 2-byte padding after the MAC header (14 bytes) of Ethernet frame enables data following the MAC header to set in 4-byte boundary.

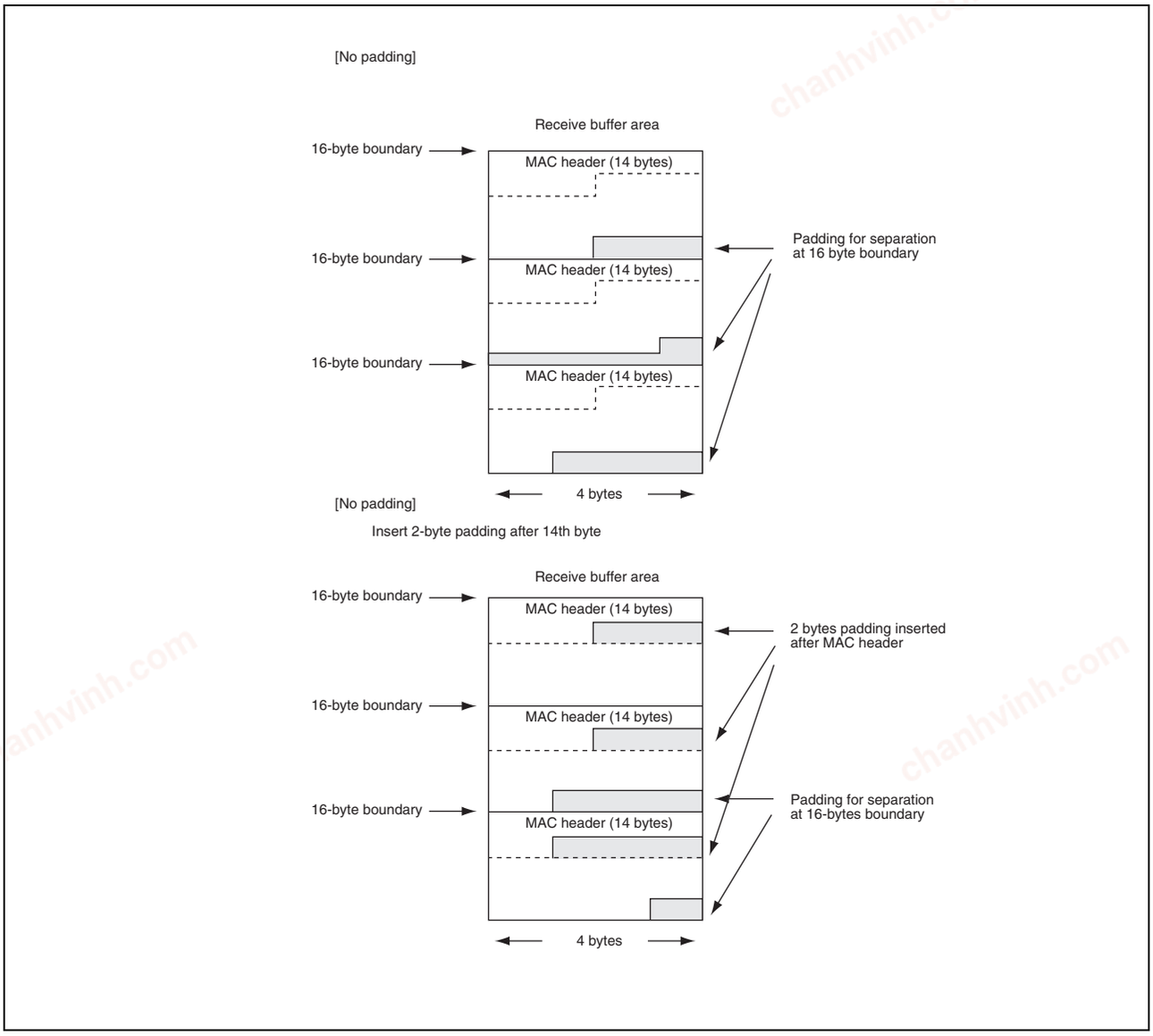


Figure 26.13 Padding Insertion in Receive Data

chanhvinh.com

26.4.7 Interrupt Processing

(1) Interrupt Sources

The ETHER issues one type of interrupt to the CPU: receive/transmit interrupts (ETHERI).
ETHERI interrupts are generated in correspondence with the transmit/receive operation. When an interrupt source is generated, it is set in EESR0 and an interrupt is issued to the CPU. For some interrupt sources, the EESR0 setting and an interrupt to the CPU are performed after a write-back operation to a descriptor is completed, not immediately after the interrupt source is detected. Interrupt sources other than the E-MAC status register source (ECI bit) are cleared by writing a 1 to the corresponding source bit. The E-MAC status register source (ECI bit) is cleared by writing a 1 to the corresponding source bit in ECSR. Interrupt source bits retain the values until they are cleared. ETHERI interrupt source is allowed to issue interrupts by setting the corresponding bit in EESIPR0. Each E-MAC state register source (ECI bit) is allowed to issue an interrupt by setting the corresponding bit in ECSIPR. In the initial value, interrupts are disabled.
Table 26.5 shows these three interrupts, interrupt sources, interrupt status registers and bits set at interrupt occurrence and interrupt generation timing.

Table 26.5 List of ETHER Interrupts

Interrupt	Interrupt Source	Register and Bit	Interrupt Generation Timing
Transmit/ receive interrupt (ETHERI)	Write-back completed	EESR0.TWB	After write-back
	Transmit underflow frame write-back completed	EESR0.TUC	After write-back
	Receive underflow frame write-back completed	EESR0.ROC	After write-back
	Transmission abort detection	EESR0.TABT	After write-back
	Reception abort detection	EESR0.RABT	After write-back
	Receive frame counter overflow	EESR0.RFCOF	When the interrupt source is detected
	E-MAC status register source	EESR0.ECI	When the interrupt source is detected
	Frame transmission completed	EESR0.TC	After write-back
	Transmit descriptor empty	EESR0.TDE	When the interrupt source is detected
	Transmit FIFO underflow	EESR0.TFUF	When the interrupt source is detected
	Frame reception	EESR0.FR	After write-back
	Receive descriptor empty	EESR0.RDE	When the interrupt source is detected
	Receive FIFO overflow	EESR0.RFOF	When the interrupt source is detected
	Receive Multicast Address Frame	EESR0.RMAF	After write-back
	Receive Residual-Bit Frame	EESR0.RRF	After write-back
	Receive Too-Long Frame	EESR0.RTLF	After write-back
	Receive Too-Short Frame	EESR0.RTSF	After write-back
	PHY-LSI Receive Error	EESR0.PRE	After write-back
	CRC Error on Received Frame	EESR0.CERF	After write-back

chanhvinh.com

26.4.8 Activation Procedure

The ETHER should be activated by the following procedure:

(1) Reset

1. Perform a power-on reset.
2. Set the ET_TXCLK and ET_RXCLK pins and supply the clock signal (for details on pin function settings, see section 41, Ports).
3. Start the E-DMAC transmitter and receiver (activation of descriptor engine).
   —Set ENT to 1 and ENR to 1 in EDSR.
4. Perform a software reset.
   —Set SWRR to 1 and SWRT to 1 in EDMR simultaneously.
5. Initialize the descriptor entry table.
6. Confirm cancellation of the software reset.
   —Check that the SWRR and SWRT bits in EDMR are cleared to 0.

(2) Pin Settings

See section 41, Ports.

(3) Registration of Descriptor Ring

The address of a descriptor ring configured in memory is registered in the descriptor entry table.

1. Transmit descriptor setting
   —Set TDLAR.
   —Set TDFAR.
   —Set TDFXR.
   —Set TDFFR. When the descriptor indicated by TDFXR is the last descriptor in the descriptor list, set H'00000001.
2. Receive descriptor setting
   —Set RDLAR.
   —Set RDFAR.
   —Set RDFXR.
   —Set RDFFR. When the descriptor indicated by RDFXR is the last descriptor in the descriptor list, set H'00000001.

(4) Register Settings

The following registers should be set as necessary.

1. E-DMAC related registers
   —Set EDMR: Operating mode, etc.
   —Set EESIPR: Interrupt masks
   —Set TRSCER: Error masks
   —Set TFTR: Transmit FIFO threshold
   —Set FDR: External FIFO size
   —Set RMCR: Reset method for reception activation
   —Set RPADIR: Padding insertion into receive data
   —Set FCFTR: Receive BSY output threshold
2. E-MAC related registers
   —Set ECMR setting: Transmission/reception specifications
   —Set ECSIPR setting: Interrupt masks
   —Set MAHR: MAC address
   —Set MALR: MAC address

chanhvinh.com

—Set RFLR: Maximum receive frame length
—Set APR: TIME parameter value of an automatic pause frame
—Set MPR: TIME parameter value of a manual PAUSE frame
—Set TPAUSER: Upper limit of automatic PAUSE frame retransmission

(5) Activation

1. Start the E-DMAC transmission/reception function
   —Set the TR bits in EDTRR to 11.
   —Set the RR bit in EDRRR to 1.
2. Start the E-MAC transmission/reception function
   —Set the TE and RE bits in ECMR to 1.

chanhvinh.com

26.4.9 Flow Control

The ETHER supports flow control functions conforming to IEEE802.3x for full-duplex operation. The flow control can be applied to both receive and transmit operations. When transmitting PAUSE frames, flow control can be performed by the following two procedures:

(1) Automatic PAUSE Frame Transmission

For receive frames, PAUSE frames are automatically transmitted when the number of data written to the receive FIFO reaches the value set in FCFTR. The TIME parameter included in the PAUSE frame is set by APR. The automatic PAUSE frame transmission is repeated until the number of data in the receive FIFO becomes less than the value set in FCFTR as the receive data is read from the FIFO. Using TPAUSER, the upper limit of retransmission counts of the PAUSE frames can also be set in the range from 1 to 65535. In this case, PAUSE frame transmission is repeated until the number of receive FIFO data becomes less than the FCFTR value, or the number of transmits reaches the value set by TPAUSER. The transmission counter is cleared to 0 when the next PAUSE frame is transmitted after the number of data in the receive FIFO becomes less than the FCFTR value.
The automatic PAUSE frame transmission is enabled when the TXF bit in ECMR is 1.

(2) Manual PAUSE Frame Transmission

PAUSE frames are transmitted by directives from the software. When writing the Timer value to MPR, manual PAUSE frame transmission is started. With this method, PAUSE frame transmission is carried out only once.

(3) PAUSE Frame Reception

The next frame is not transmitted until the time indicated by the Timer value elapses after receiving a PAUSE frame. However, the transmission of the current frame is continued. A received PAUSE frame is valid only when the RXF bit in ECMR is set to 1. The number of times of PAUSE frame receptions is counted.

(4) 0-Time PAUSE Frame Control

Flow control is performed using a PAUSE frame with the TIME parameter value set to 0. The PAUSE frame with the TIME parameter set to 0 can be enabled or disabled by the ZPF bit in ECMR.

- When PAUSE frame control with the TIME parameter value set to 0 is enabled
  A PAUSE frame with the TIME parameter value set to 0 is transmitted when the number of data in the receive FIFO is less than the FCFTR value before the time indicated by the TIME parameter value has not elapsed. When a PAUSE frame with the time indicated by the TIME parameter value set to 0 is received, the transmit standby state is canceled.
- When PAUSE frame control with the TIME parameter value set to 0 is disabled
  A PAUSE frame with the TIME parameter value set to 0 is not transmitted. When a PAUSE frame with the TIME parameter value set to 0 is received, the PAUSE frame is discarded.

chanhvinh.com

26.4.10 Intelligent Checksum Calculation Function

This function accelerates checksum calculation on received packets, and provides the following two modes.

- MAC/IP packet analyzing intelligent checksum calculation mode
- All-data intelligent checksum calculation mode with bytes to be skipped specified

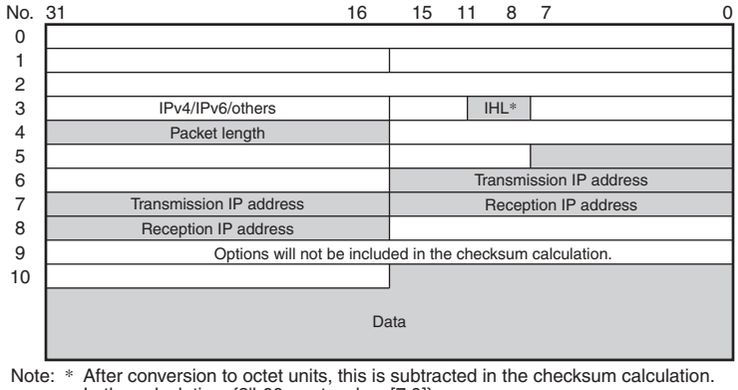
(1) MAC/IP Packet Analyzing Intelligent Checksum Calculation Mode (CSEBL = 1 and CSMD = 1 in CSMR)

In this mode, the checksum of received packets indicated in the table is calculated. However, if a MAC packet payload includes padding data in fields other than those for the IP packet itself because there is too little data for a full packet, it is not included in the checksum.

IPver	Items
IPv4	Option present
	Option not present
	Fragment*1
IPv6	Extension header not present
	Hop-by-hop options extension header length
	Routing extension header length
	Fragment extension header length*1
	Destination options header length
	AH extension header length
	ESP extension header length*2
	Extension header length for mobile IPv6*2

Note 1. This packet is to be checksumed, however, the RCS[15:0] bits and RCSE bit in RD0 are to be undefined even if the data is successfully received.
Note 2. The RD0.RCSE bit is set to 1 without calculating the value of the RD0.RCS[15:0] bits.

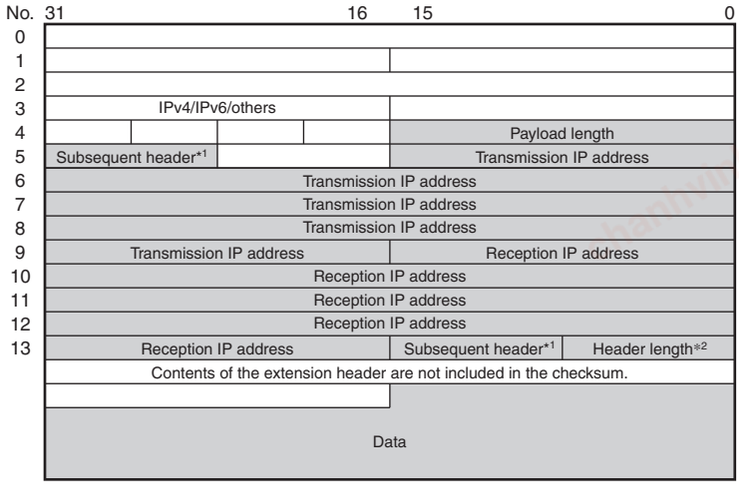
The shaded regions of the following figure indicate the parts of an IPv4 packet which are used to obtain the checksum.



Note: * After conversion to octet units, this is subtracted in the checksum calculation.
In the calculation: {8'h00, protocol no.[7:0]}

The shaded regions of the following figure indicate the parts of an IPv6 packet which are used to obtain the checksum.

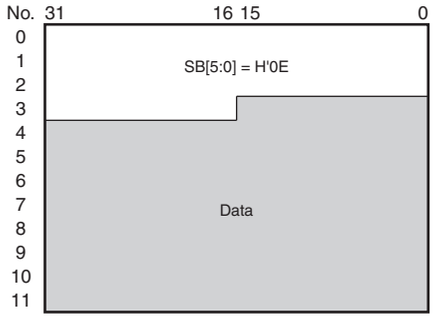
chanhvinh.com



Notes: *1 Only included in the checksum when the header is for the TCP or UDP. Calculation coverage is extended to {8'h00, protocol No.[7:0]} when the checksum is taken.
*2 After conversion to octet units, this is subtracted in the checksum calculation.

(2) All-Data Intelligent Checksum Calculation Mode with Bytes to be Skipped Specified (CSELB = 1 and CSMD = 0 in CSMR)

After having skipped the number of bytes specified in the SB[5:0] bits in CSMR, counting from the beginning of the packet, the checksum is calculated for all subsequent data (e.g. 14 bytes may be skipped).



chanhvinh.com

26.5 Connection to PHY-LSI

26.5.1 MII Frame Transmission/Reception Timing

Each MII frame transmission/reception timing is shown in Figure 26.14 to Figure 26.17.

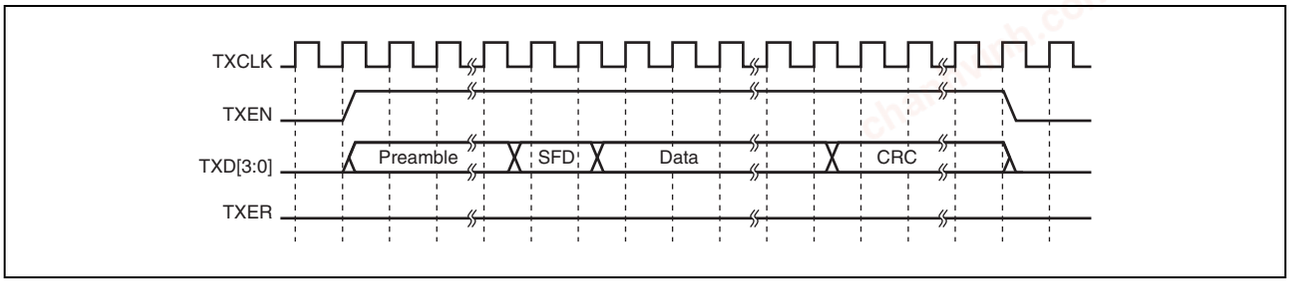


Figure 26.14 MII Frame Transmit Timing (Normal Transmission)

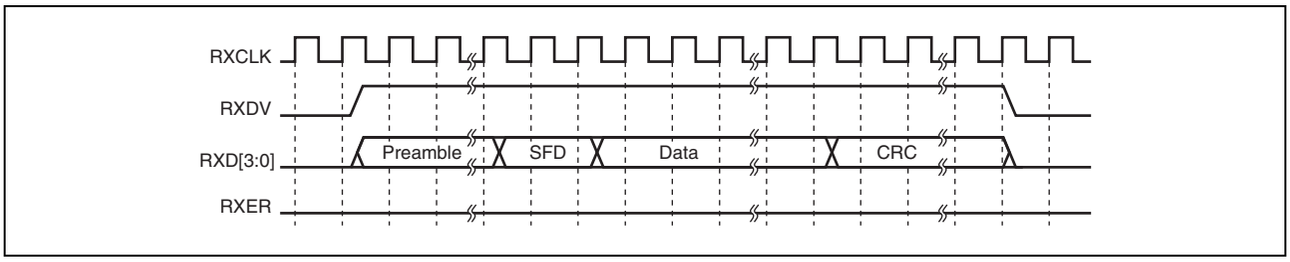


Figure 26.15 MII Frame Receive Timing (Normal Reception)

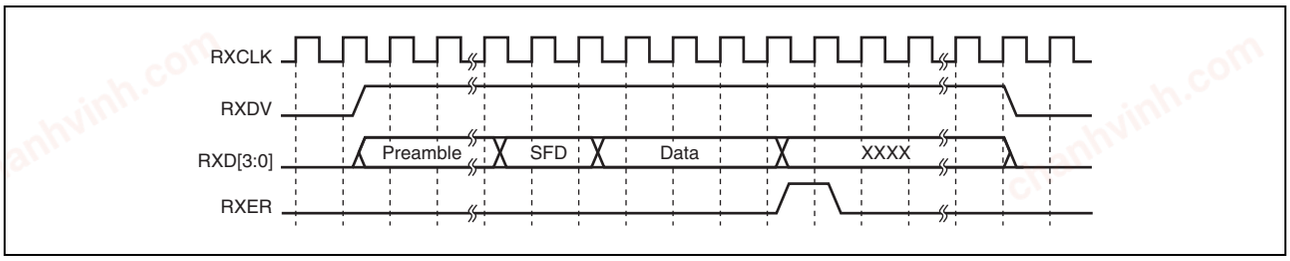


Figure 26.16 MII Frame Receive Timing (Reception Error (1))

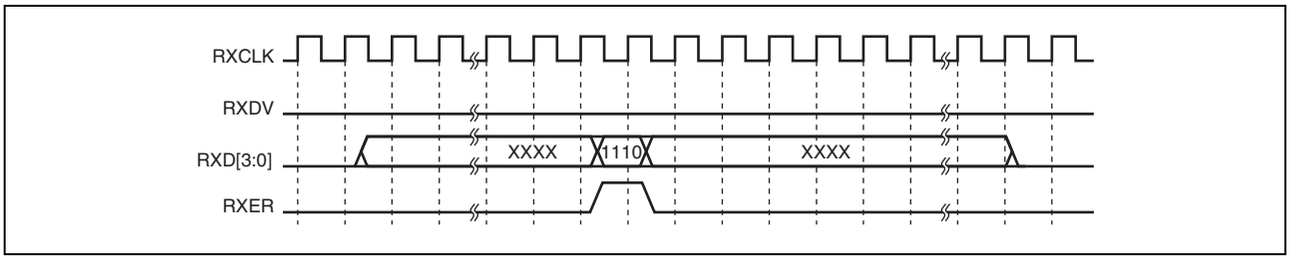


Figure 26.17 MII Frame Receive Timing (Reception Error (2))

chanhvinh.com

26.5.2 Accessing MII Registers

MII registers in the PHY-LSI are accessed via PIR in this LSI. PIR is used as a serial interface conforming to the MII frame format specified in IEEE802.3u.

(1) MII Management Frame Format

Figure 26.18 shows the format of an MII management frame. To access an MII register, a management frame is implemented by the program in accordance with the procedures shown in MII Register Access Procedure.

Access Type	MII Management Frame							
Item	PRE	ST	OP	PHYAD	REGAD	TA	DATA	IDLE
Number of bits	32	2	2	5	5	2	16	
Read	1..1	01	10	00001	RRRRR	Z0	D..D	
Write	1..1	01	01	00001	RRRRR	10	D..D	X

[Legend]

PRE: 32 consecutive 1s
ST: Write of 01 indicating start of frame
OP: Write of code indicating access type
PHYAD: Write of 0001 if the PHY-LSI address is 1 (sequential write starting with the MSB).
This bit changes depending on the PHY-LSI address.
REGAD: Write of 000q if the register address is 1 (sequential write starting with the MSB).
This bit changes depending on the PHY-LSI register address.
TA: Time for switching data transmission source on MII interface
(a) Write: 10 written
(b) Read: Bus release (notation: Z0) performed
DATA: 16-bit data. Sequential write or read from MSB
(a) Write: 16-bit data write
(b) Read: 16-bit data read
IDLE: Wait time until next MII management format input
(a) Write: Independent bus release (notation: X) performed
(d) Read: Bus already released in TA: control unnecessary

Figure 26.18 MII Management Frame Format

chanhvinh.com

(2) MII Register Access Procedure

The program accesses MII registers via PIR. Access is implemented by a combination of 1-bit-unit data write, 1-bit-unit data read, bus release, and independent bus release. Figure 26.19 (1) to Figure 26.22 (4) show the MII register access timing. The timing will differ depending on the PHY-LSI type.

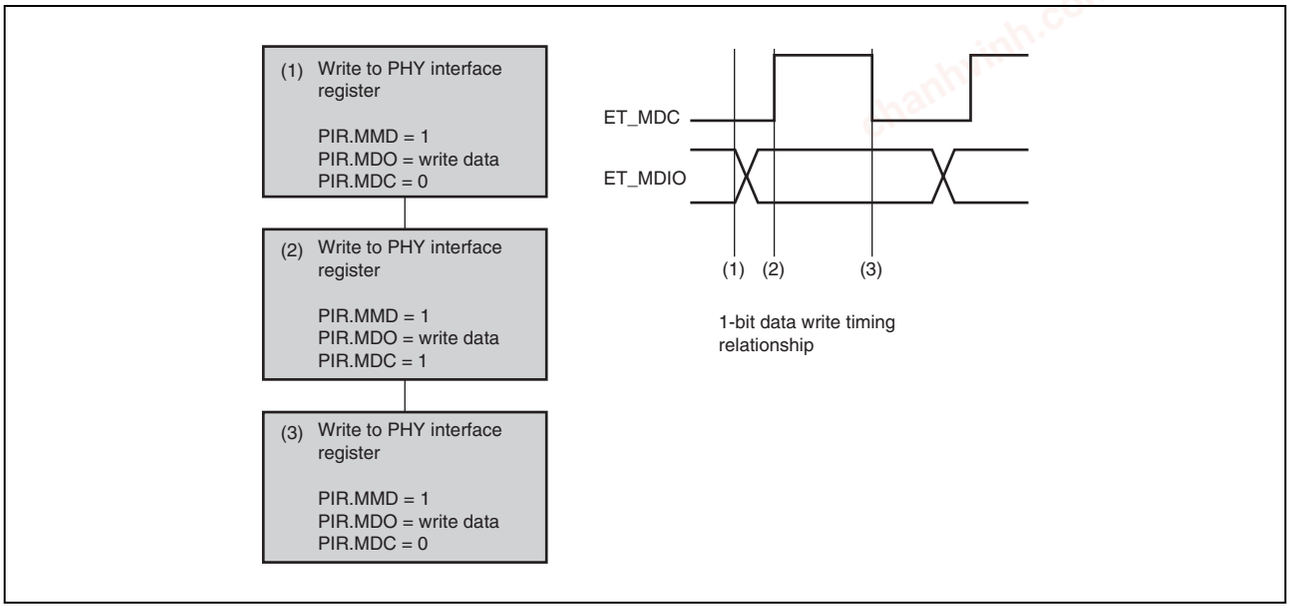


Figure 26.19 1-Bit Data Write Flowchart

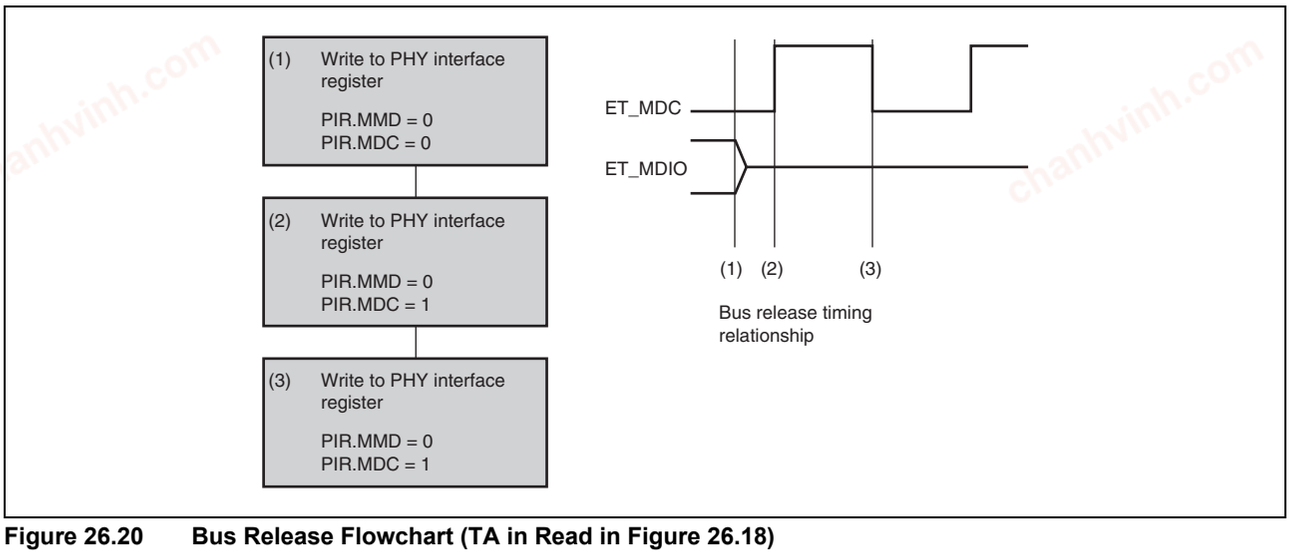


Figure 26.20 Bus Release Flowchart (TA in Read in Figure 26.18)

chanhvinh.com

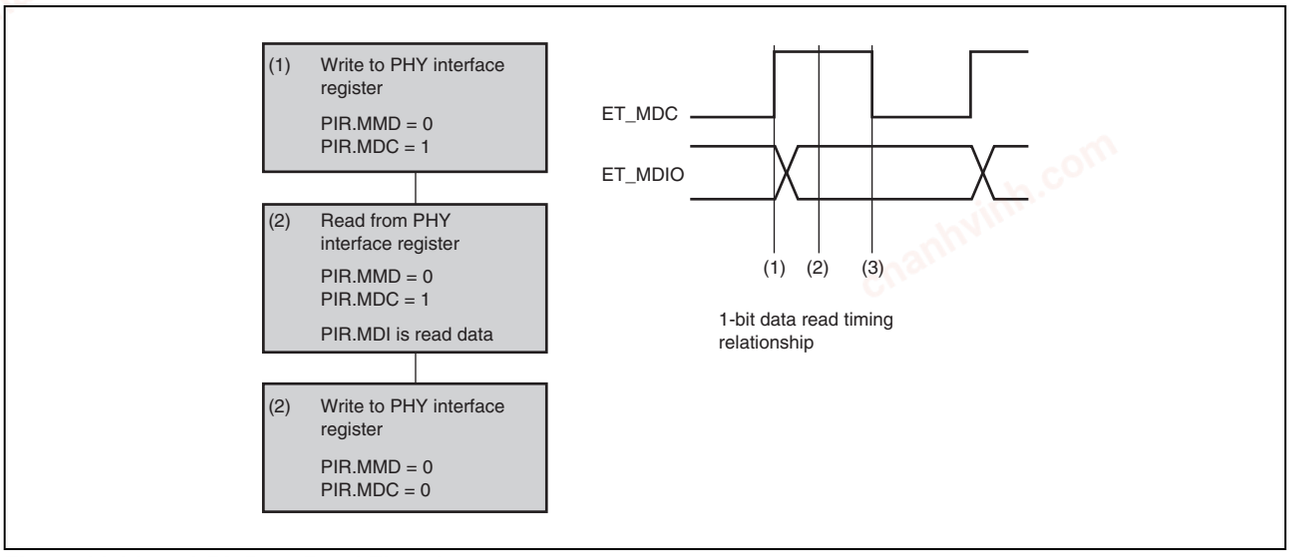


Figure 26.21 1-Bit Data Read Flowchart

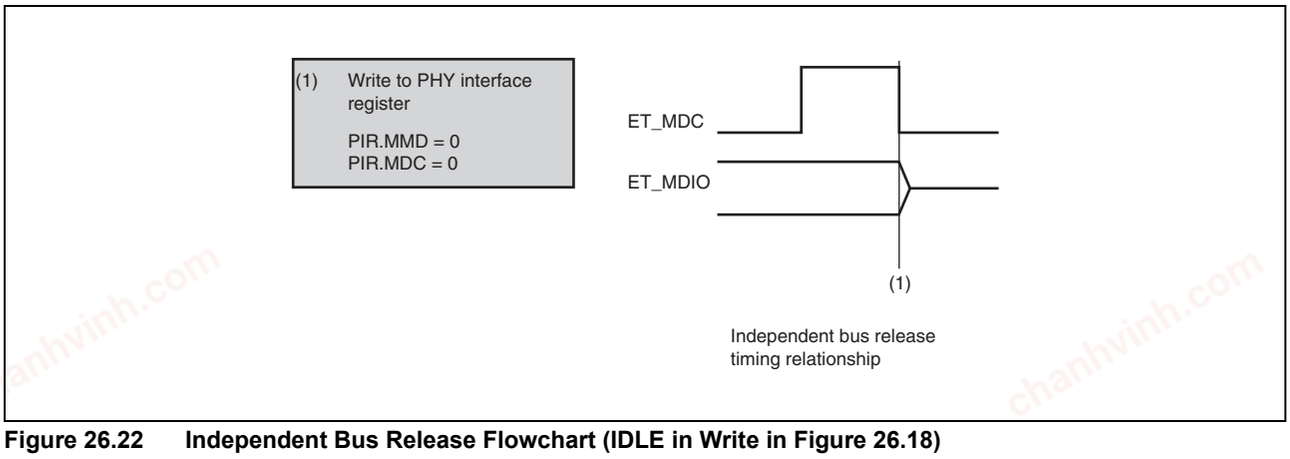


Figure 26.22 Independent Bus Release Flowchart (IDLE in Write in Figure 26.18)

chanhvinh.com

26.6 Usage Notes

26.6.1 Checksum Calculation of Ethernet Frames

This LSI is capable of calculating the checksum data of the received frames. Only the data fields of the Ethernet frames are subject to checksum calculation. Specifically, a data field follows the length/type field and is followed by the CRC field. Figure 26.23 shows schematics indicating which parts of the Ethernet frames are calculated. Calculation involves 16-bit addition only; it does not involve bit inversion. Note that when the checksum data is valid, the CRC data (4 bytes) is not transferred as a receive frame, and the checksum data (sum data) is added automatically. Figure 26.24 shows schematics of Ethernet frames to which the checksum data has been added.

Note: • Also for the frames with VLANtag inserted, the 15th byte from the top and the following bytes before the CRC field are subject to calculation.

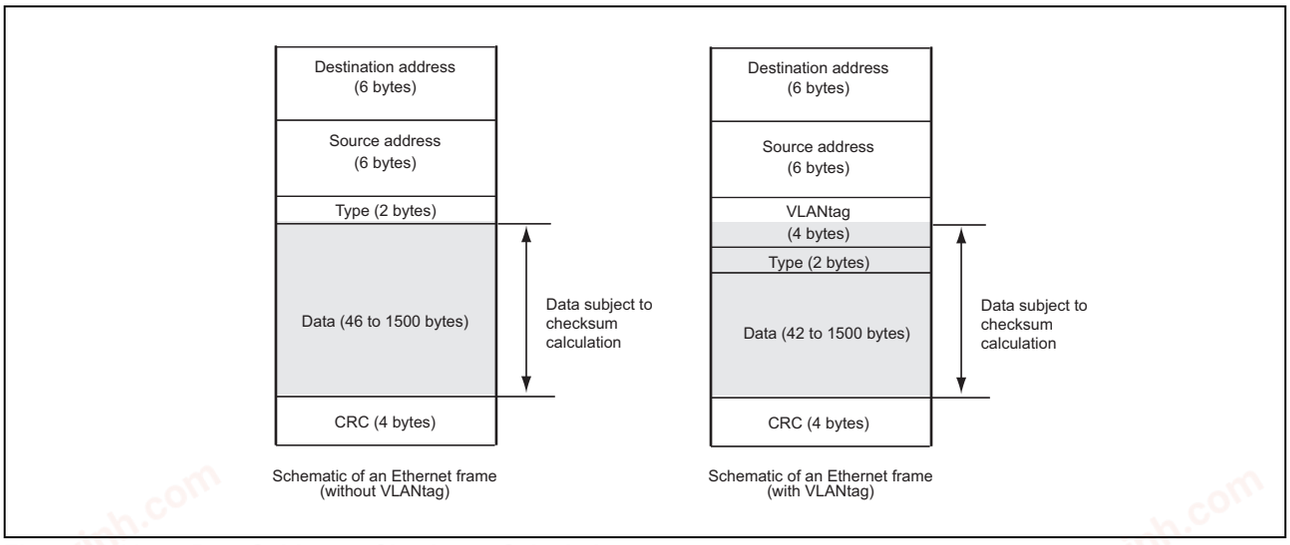


Figure 26.23 Data Subject to Checksum Calculation

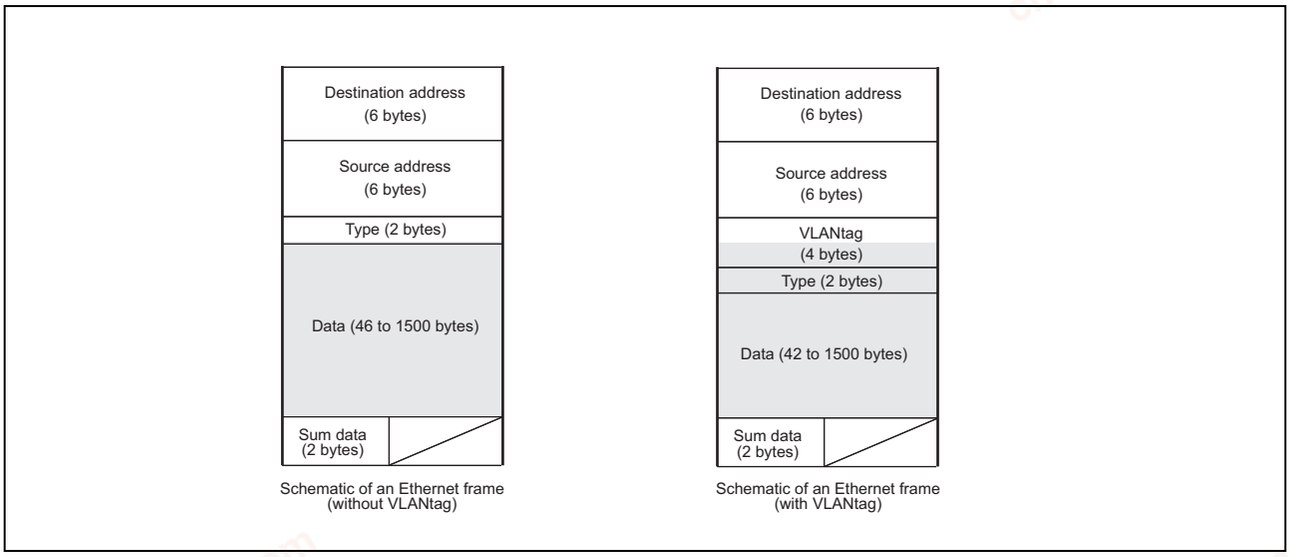


Figure 26.24 Data after Checksum Data Addition

chanhvinh.com

26.6.2 Notes on Using the Intelligent Checksum Function

Checksum calculation using the intelligent checksum function is not affected by padding insertion specified by the receive data padding insert register (RPADIR). This is because checksum calculation is performed when transferring receive data from E-MAC to E-DMAC, while padding of receive data is performed when transferring receive data from E-DMAC to the receive buffer in memory.

26.6.3 Software Reset

For transitions to the software reset state by the ARST bit in the software reset register (ARSTR) or SWRT and SWRR bits in the E-DMAC mode register (EDMR0), see section 42.3.6, Software Reset. However, where the procedure refers to the SRST bit, read this as the ARST bit or SWRT and SWRR bits.

chanhvinh.com

27. A/D Converter

This LSI includes a 12-bit successive-approximation A/D converter allowing selection of up to eight analog input channels.

27.1 Features

- Resolution: 12 bits
- Input channels: Eight channels
- Minimum conversion time: 5.0 µs per channel
- Absolute accuracy: ±11 LSB
- Operating modes: Three
  - Single mode: A/D conversion on one channel
  - Multi mode: A/D conversion on one to four channels or on one to eight channels
  - Scan mode: Continuous A/D conversion on one to four channels or on one to eight channels
- Data registers: Eight
  Conversion results are held in a 16-bit data register for each channel
- Sample-and-hold function
- A/D conversion start methods: Three methods
  - Software
  - Conversion start trigger from the multi-function timer pulse unit 2
  - External trigger signal
- Interrupt source: Two sources
  An A/D conversion end interrupt (ADI) request can be generated on completion of A/D conversion.
  An over-limit interrupt (LMTI) request can be generated when a result of conversion exceeds the upper or lower limit for each channel.*

Note: * This is only available when 10-bit precision is in use.

- Module standby mode can be set

chanhvinh.com

Figure 27.1 shows a block diagram of the A/D converter.

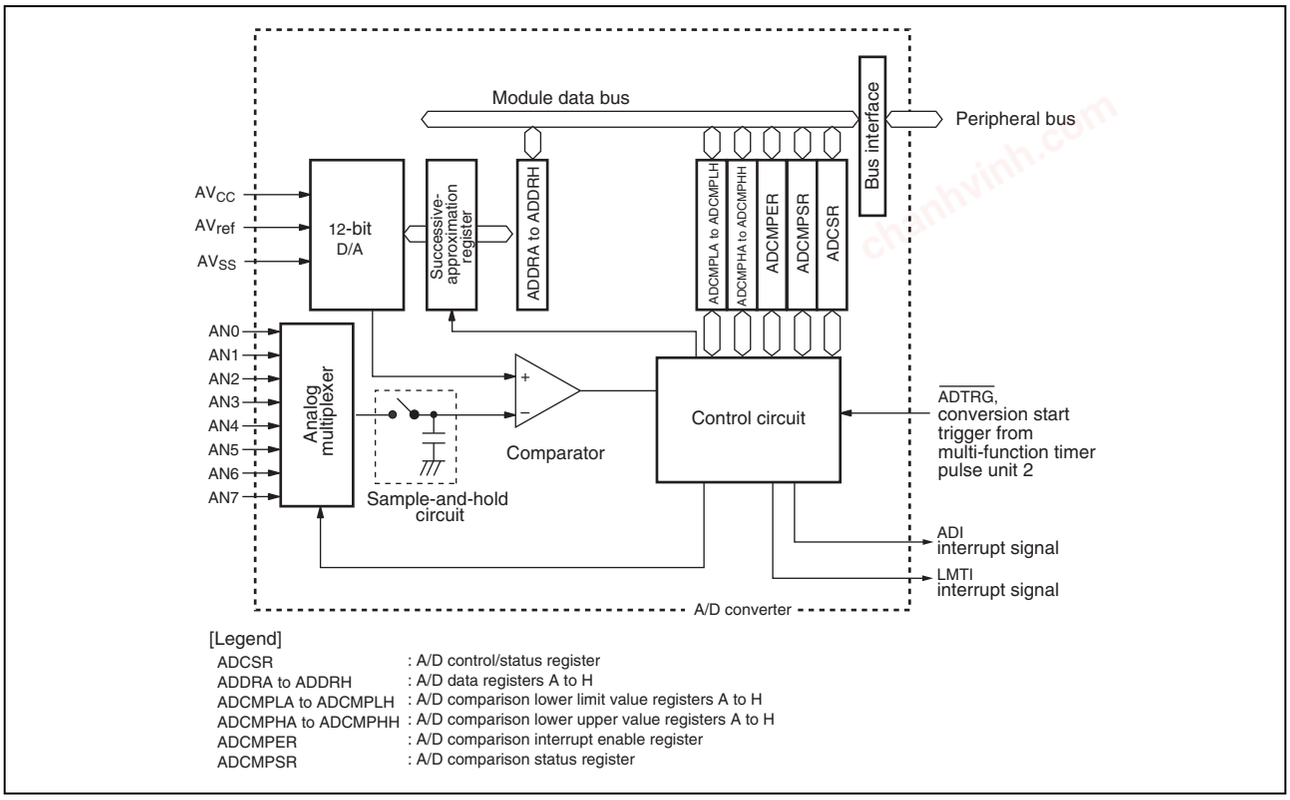


Figure 27.1 Block Diagram of A/D Converter

chanhvinh.com

27.2 Input/Output Pins

Table 27.1 shows the A/D converter pins.

Table 27.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Analog power supply pin	AVcc	Input	Analog power supply pin
Analog ground pin	AVss	Input	Analog ground pin and A/D conversion reference ground
Analog reference voltage pin	AVref	Input	A/D converter reference voltage pin
Analog input pin 0	AN0	Input	Analog input
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
A/D external trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input to start A/D conversion

chanhvinh.com

27.3 Register Descriptions

The A/D converter has the following registers.

Table 27.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
A/D data register A	ADDRA	R	H'0000	H'E8005800	16
A/D data register B	ADDRB	R	H'0000	H'E8005802	16
A/D data register C	ADDRC	R	H'0000	H'E8005804	16
A/D data register D	ADDRD	R	H'0000	H'E8005806	16
A/D data register E	ADDRE	R	H'0000	H'E8005808	16
A/D data register F	ADDRF	R	H'0000	H'E800580A	16
A/D data register G	ADDRG	R	H'0000	H'E800580C	16
A/D data register H	ADDRH	R	H'0000	H'E800580E	16
A/D comparison upper limit value register A	ADCMPHA	R/W	H'0000	H'E8005820	16
A/D comparison lower limit value register A	ADCMPLA	R/W	H'0000	H'E8005822	16
A/D comparison upper limit value register B	ADCMPHB	R/W	H'0000	H'E8005824	16
A/D comparison lower limit value register B	ADCMPLB	R/W	H'0000	H'E8005826	16
A/D comparison upper limit value register C	ADCMPHC	R/W	H'0000	H'E8005828	16
A/D comparison lower limit value register C	ADCMPLC	R/W	H'0000	H'E800582A	16
A/D comparison upper limit value register D	ADCMPHD	R/W	H'0000	H'E800582C	16
A/D comparison lower limit value register D	ADCMPLD	R/W	H'0000	H'E800582E	16
A/D comparison upper limit value register E	ADCMPHE	R/W	H'0000	H'E8005830	16
A/D comparison lower limit value register E	ADCMPLE	R/W	H'0000	H'E8005832	16
A/D comparison upper limit value register F	ADCMPHF	R/W	H'0000	H'E8005834	16
A/D comparison lower limit value register F	ADCMPLF	R/W	H'0000	H'E8005836	16
A/D comparison upper limit value register G	ADCMPHG	R/W	H'0000	H'E8005838	16
A/D comparison lower limit value register G	ADCMPLG	R/W	H'0000	H'E800583A	16
A/D comparison upper limit value register H	ADCMPHH	R/W	H'0000	H'E800583C	16
A/D comparison lower limit value register H	ADCMPLH	R/W	H'0000	H'E800583E	16
A/D control/status register	ADCSR	R/W	H'0000	H'E8005860	16
A/D comparison interrupt enable register	ADCMPER	R/W	H'0000	H'E8005862	16
A/D comparison status register	ADCMPSR	R/W	H'0000	H'E8005864	16

chanhvinh.com

27.3.1 A/D Data Registers A to H (ADDRA to ADDRH)

The eight A/D data registers, ADDRA to ADDRH, are 16-bit read-only registers that store the results of A/D conversion. An A/D conversion produces 12-bit data, which is transferred for storage into bits 15 to 4 of the ADDR corresponding to the selected channel. Bits 3 to 0 of ADDR are reserved bits that are always read as 0.

Access to ADDR in 8-bit units is prohibited. ADDR must always be accessed in 16-bit units.

Table 27.3 indicates the correspondence between analog input channels and ADDR.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 4		All 0	R	Bit Data (12bits)
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Table 27.3 Correspondence between Analog Input Channels and ADDR

Analog Input Channel	A/D Data Register where Conversion Result is Stored
AN0	ADDRA
AN1	ADDRB
AN2	ADDRC
AN3	ADDRD
AN4	ADDRE
AN5	ADDRF
AN6	ADDRG
AN7	ADDRH

chanhvinh.com

27.3.2 A/D Comparison Upper Limit Value Registers A to H (ADCMPHA to ADCMPHH)

ADCMPH is a 16-bit readable/writable register that holds the upper limit value to be compared with an A/D conversion result. Eight registers, ADCMPHA to ADCMPHH, are provided.

On completion of A/D conversion on each channel, the A/D conversion result is compared with the data stored in the corresponding upper limit value register. When the result is greater than the upper limit value, the corresponding bit in the A/D comparison status register (ADCMPSR) is set. Comparison with the upper limit value is performed only when the corresponding bit in the A/D comparison interrupt enable register (ADCMPER) is set.

Access to ADCMPH in 8-bit units is prohibited. ADCMPH must always be accessed in 16-bit units.

Table 27.4 indicates the correspondence between the ADDR registers and the ADCMPH/ADCMPL registers.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 6		All 0	R/W	Comparison Upper Limit Value (10 bits)
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

27.3.3 A/D Comparison Lower Limit Value Registers A to H (ADCMPLA to ADCMPLH)

ADCMPL is a 16-bit readable/writable register that holds the lower limit value to be compared with an A/D conversion result. Eight registers, ADCMPLA to ADCMPLH, are provided.

On completion of A/D conversion on each channel, the A/D conversion result is compared with the data stored in the corresponding lower limit value register. When the result is smaller than the lower limit value, the corresponding bit in the A/D comparison status register (ADCMPSR) is set. Comparison with the lower limit value is performed only when the corresponding bit in the A/D comparison interrupt enable register (ADCMPER) is set.

Access to ADCMPL in 8-bit units is prohibited. ADCMPL must always be accessed in 16-bit units.

Table 27.4 indicates the correspondence between the ADDR registers and the ADCMPH/ADCMPL registers.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 6		All 0	R/W	Comparison Lower Limit Value (10 bits)
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Table 27.4 Correspondence between ADDR and ADCMPH/ADCMPL Registers

ADDR where Conversion Result is Stored	ADCMPH/ADCMPL (Target for Comparison)
ADDRA	ADCMPHA/ADCMPLA
ADDRB	ADCMPHB/ADCMPLB
ADDRC	ADCMPHC/ADCMPLC
ADDRD	ADCMPHD/ADCMPLD
ADDRE	ADCMPHE/ADCMPLE
ADDRF	ADCMPHF/ADCMPLF
ADDRG	ADCMPHG/ADCMPLG
ADDRH	ADCMPHH/ADCMPLH

chanhvinh.com

27.3.4 A/D Control/Status Register (ADCSR)

ADCSR is a 16-bit readable/writable register that selects the operating mode, controls the A/D converter, and enables or disables starting of A/D conversion by external trigger input.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADF	ADIE	ADST	TRGS[3:0]				CKS[2:0]			MDS[2:0]			CH[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: *1 Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
15	ADF	0	R/(W)*1	A/D End Flag Status flag indicating the end of A/D conversion. [Clearing conditions] • Cleared by reading ADF while ADF = 1, then writing 0 to ADF • Cleared when the direct memory access controller is activated by ADI interrupt and ADDR is read [Setting conditions] • A/D conversion ends in single mode • A/D conversion ends for all the selected channels in multi mode • A/D conversion ends for all the selected channels in scan mode
14	ADIE	0	R/W	A/D Interrupt Enable Enables or disables the interrupt (ADI) requested at the end of A/D conversion. Set the ADIE bit while A/D conversion is not being performed. 0: A/D conversion end interrupt (ADI) request is disabled 1: A/D conversion end interrupt (ADI) request is enabled
13	ADST	0	R/W*2	A/D Start Starts or stops A/D conversion. This bit remains set to 1 during A/D conversion. 0: A/D conversion is stopped 1: Single mode: A/D conversion starts. This bit is automatically cleared to 0 when A/D conversion ends on the selected channel. Multi mode: A/D conversion starts. This bit is automatically cleared to 0 when A/D conversion is completed cycling through the selected channels. Scan mode: A/D conversion starts. A/D conversion is continuously performed until this bit is cleared to 0 by software, by a power-on reset as well as by a transition to deep standby mode.
12 to 9	TRGS[3:0]	0000	R/W	Timer Trigger Select These bits enable or disable starting of A/D conversion by a trigger signal. 0000: Start of A/D conversion by external trigger input is disabled 0001: A/D conversion is started by conversion trigger TRGAN from the multi-function timer pulse unit 2 0010: A/D conversion is started by conversion trigger TRG0N from the multi-function timer pulse unit 2 0011: A/D conversion is started by conversion trigger TRG4AN from the multi-function timer pulse unit 2 0100: A/D conversion is started by conversion trigger TRG4BN from the multi-function timer pulse unit 2 1001: A/D conversion is started by $\overline{\text{ADTRG}}$ Other than above: Setting prohibited
8 to 6	CKS[2:0]	000	R/W	Clock Select These bits select the A/D conversion time.*3 Set the A/D conversion time while A/D conversion is halted (ADST = 0). 000: Conversion time = 256 t_{cyc}*4 (maximum) 001: Conversion time = 298 t_{cyc}*4 (maximum) 010: Conversion time = 340 t_{cyc}*4 (maximum) 011: Conversion time = 382 t_{cyc}*4 (maximum) 100, 101, 110, 111: Setting prohibited

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
5 to 3	MDS[2:0]	000	R/W	Multi-scan Mode These bits select the operating mode for A/D conversion. 0xx: Single mode 100: Multi mode: A/D conversion on 1 to 4 channels 101: Multi mode: A/D conversion on 1 to 8 channels 110: Scan mode: A/D conversion on 1 to 4 channels 111: Scan mode: A/D conversion on 1 to 8 channels
2 to 0	CH[2:0]	000	R/W	Channel Select These bits and the MDS bits in ADCSR select the analog input channels.

MDS = 0xx	MDS = 100 or MDS = 110	MDS = 101 or MDS = 111
000: AN0	000: AN0	000: AN0
001: AN1	001: AN0, AN1	001: AN0, AN1
010: AN2	010: AN0 to AN2	010: AN0 to AN2
011: AN3	011: AN0 to AN3	011: AN0 to AN3
100: AN4	100: AN4	100: AN0 to AN4
101: AN5	101: AN4, AN5	101: AN0 to AN5
110: AN6	110: AN4 to AN6	110: AN0 to AN6
111: AN7	111: AN4 to AN7	111: AN0 to AN7

[Legend]
x: Don't care
Note 1. Only 0 can be written to clear the flag after 1 is read.
Note 2. Clear the ADST to stop A/D conversion before transition to software standby mode or module standby mode.
Note 3. Set the A/D conversion time to at least the minimum value to obtain the absolute accuracy specified for the A/D converter.
Note 4. t_{cyc} indicates a cycle time of the peripheral clock 1 (P1φ).

chanhvinh.com

27.3.5 A/D Comparison Interrupt Enable Register (ADCMPER)

ADCMPER is a 16-bit readable/writable register that enables or disables comparison between the A/D data register (ADDR) that holds the A/D conversion result on each channel and the A/D comparison upper/lower limit value register (ADCMPH/ADCMPL), and enables or disables interrupt request.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HLMENH	HLMENG	HLMENF	HLMENE	HLMEND	HLMENC	HLMENB	HLMENA	LLMENH	LLMENG	LLMENF	LLMENE	LLMEND	LLMENC	LLMENB	LLMENA
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	HLMENH to HLMENA	All 0	R/W	Comparison with Upper Limit Value and Interrupt Enable Enables or disables comparison between an A/D conversion result and the upper limit value as well as an LMTI interrupt request in the case of the result being greater than the upper limit value. 0: Disables comparison between an A/D conversion result and the upper limit value as well as an over-limit interrupt (LMTI) request. 1: Enables comparison between an A/D conversion result and the upper limit value as well as an over-limit interrupt (LMTI) request.
7 to 0	LLMENH to LLMENA	All 0	R/W	Comparison with Lower Limit Value and Interrupt Enable Enables or disables comparison between an A/D conversion result and the lower limit value as well as an LMTI interrupt request in the case of the result being smaller than the lower limit value. 0: Disables comparison between an A/D conversion result and the lower limit value as well as an over-limit interrupt (LMTI) request. 1: Enables comparison between an A/D conversion result and the lower limit value as well as an over-limit interrupt (LMTI) request.

chanhvinh.com

27.3.6 A/D Comparison Status Register (ADCMPSR)

ADCMPSR is a 16-bit read-only register that indicates the results of comparison between the A/D data register (ADDR) that holds an A/D conversion result on each channel and the A/D comparison upper/lower limit value register (ADCMPH/ADCMPL).

The bits in this register are valid only when the comparison and an LMTI interrupt are enabled in the A/D comparison interrupt enable register (ADCMPER). When they are disabled, the bits are always in the cleared state.

When an A/D conversion result exceeds the upper or lower limit value, the corresponding bit is set to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HOVRH	HOVRG	HOVRF	HOVRE	HOVRD	HOVRC	HOVRB	HOVRA	LUDRH	LUDRG	LUDRF	LUDRE	LUDRD	LUDRC	LUDRB	LUDRA
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	HOVRH to HOVRA	All 0	R	Upper Limit Over Flag Status flag that indicates that an A/D conversion result on each channel has exceeded the upper limit value in ADCMPH. [Clearing condition] • Cleared by writing 0 to the HLMENn bit for the corresponding channel in ADCMPER register [Setting condition] The HLMEN bit for the corresponding channel in ADCMPER is 1, and an A/D conversion result for the corresponding channel that is greater than the upper limit value in ADCMPH is stored in the ADDR register
7 to 0	LUDRH to LUDRA	All 0	R	Lower Limit Under Flag Status flag that indicates that an A/D conversion result on each channel has exceeded the lower limit value in ADCMPL. [Clearing condition] • Cleared by writing 0 to the LLMENn bit for the corresponding channel in ADCMPER register [Setting condition] The LLMEN bit for the corresponding channel in ADCMPER is 1, and an A/D conversion result for the corresponding channel that is smaller than the lower limit value in ADCMPL is stored in the ADDR register

chanhvinh.com

27.4 Operation

The A/D converter uses the successive-approximation method, and the resolution is 12 bits. It has three operating modes: single mode, multi mode, and scan mode. Switching the operating mode or analog input channels must be done while the ADST bit in ADCSR is 0 to prevent incorrect operation. The ADST bit can be set at the same time as the operating mode or analog input channels are changed.

27.4.1 Single Mode

Single mode should be selected when only A/D conversion on one channel is required.
In single mode, A/D conversion is performed once for the specified one analog input channel, as follows:

1. A/D conversion for the selected channel starts when the ADST bit in ADCSR is set to 1 by software, the multi-function timer pulse unit 2, or external trigger input.
2. When A/D conversion is completed, the A/D conversion result is transferred to the A/D data register corresponding to the channel.
3. After A/D conversion has completed, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated.
4. The ADST bit that remains 1 during A/D conversion is automatically cleared to 0 when A/D conversion is completed, and the A/D converter becomes idle.

When the operating mode or analog input channel selection must be changed during A/D conversion, to prevent incorrect operation, first clear the ADST bit to 0 to halt A/D conversion. After making the necessary changes, set the ADST bit to 1 to start A/D conversion again. The ADST bit can be set at the same time as the mode or channel selection is switched.

Typical operations when a single channel (AN1) is selected in single mode (MDS[2] = 0) are described next. Figure 27.2 shows a timing diagram for this example (the bits which are set in this example belong to ADCSR).

1. Single mode is selected, input channel AN1 is selected (CH[2:0] = 001), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started (ADST = 1).
2. When A/D conversion is completed, the A/D conversion result is transferred into ADDRB. At the same time the ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becomes idle.
3. Since ADF = 1 and ADIE = 1, an ADI interrupt is requested.
4. The A/D interrupt handling routine starts.
5. The routine reads ADF = 1, and then writes 0 to the ADF flag.
6. The routine reads and processes the A/D conversion result (ADDRB).
7. Execution of the A/D interrupts handling routine ends. Then, when the ADST bit is set to 1, A/D conversion starts and steps 2 to 7 are executed.

chanhvinh.com

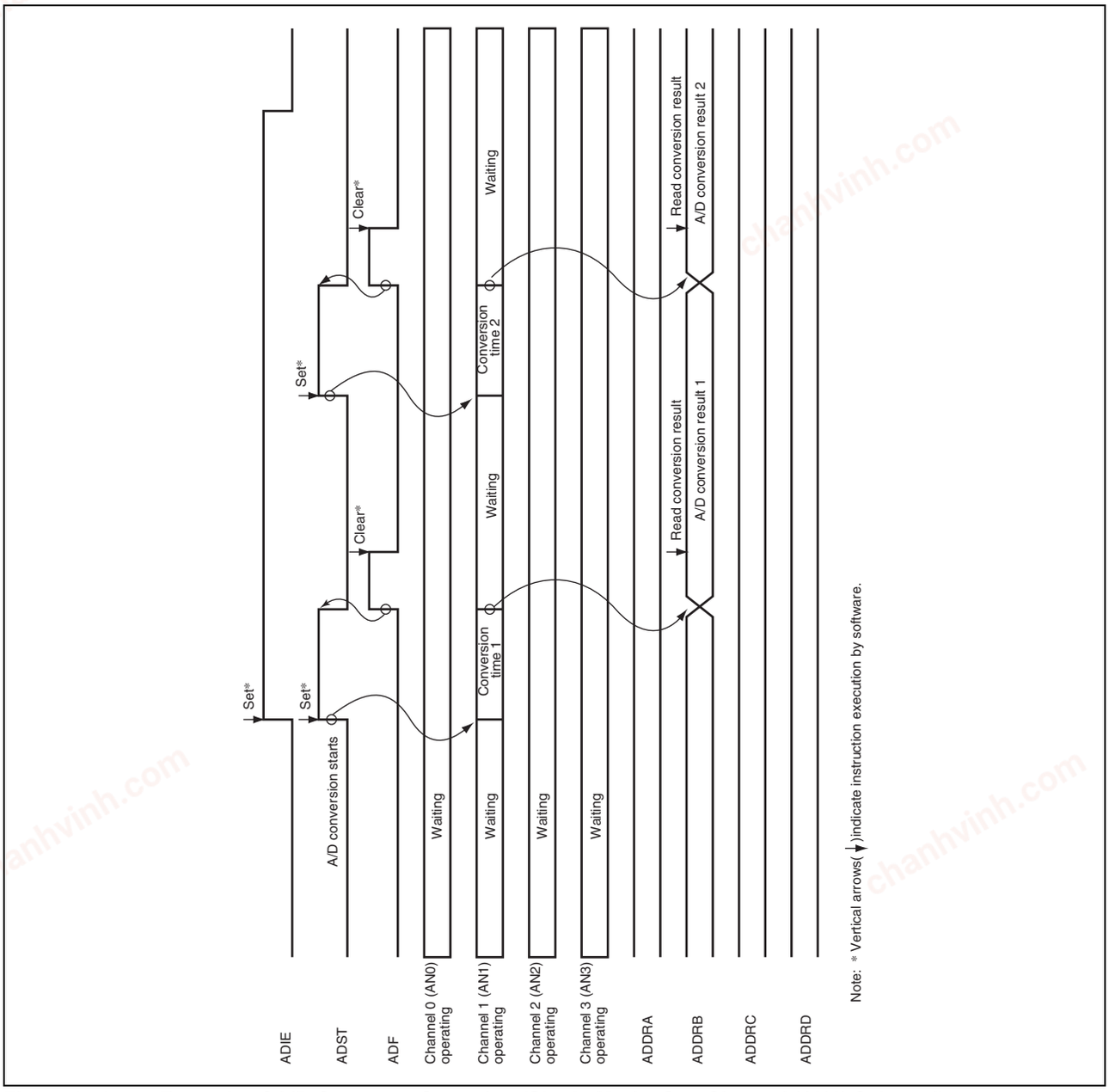


Figure 27.2 Example of A/D Converter Operation (Single Mode, One Channel (AN1) Selected)

chanhvinh.com

27.4.2 Multi Mode

Multi mode should be selected when performing A/D conversion once on one or more channels.
In multi mode, A/D conversion is performed once for a maximum of eight specified analog input channels, as follows:

1. A/D conversion starts from the analog input channel with the lowest number (e.g. AN0, AN1, …, AN3) when the ADST bit in ADCSR is set to 1 by software, the multi-function timer pulse unit 2, or external trigger input.
2. When A/D conversion is completed on each channel, the A/D conversion result is sequentially transferred to the A/D data register corresponding to that channel.
3. After A/D conversion on all selected channels has completed, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated.
4. The ADST bit that remains 1 during A/D conversion is automatically cleared to 0 when A/D conversion is completed, and the A/D converter becomes idle. If the ADST bit is cleared to 0 during A/D conversion, A/D conversion is halted and the A/D converter becomes idle. The ADF bit is cleared by reading ADF while ADF = 1, then writing 0 to the ADF bit.

A/D conversion is to be performed once on all the specified channels. The conversion results are transferred for storage into the A/D data registers (ADDR) corresponding to the channels.
When the operating mode or analog input channel selection must be changed during A/D conversion, to prevent incorrect operation, first clear the ADST bit to 0 to halt A/D conversion. After making the necessary changes, set the ADST bit to 1. A/D conversion will start again from the first channel in the group. The ADST bit can be set at the same time as the mode or channel selection is changed.
Typical operations when three channels (AN0 to AN2) are selected in multi mode are described next. Figure 27.3 shows a timing diagram for this example.

1. Multi mode is selected (MDS[2] = 1, MDS[1] = 0), analog input channels AN0 to AN2 are selected (CH[2:0] = 010), and A/D conversion is started (ADST = 1).
2. A/D conversion of the first channel (AN0) starts. When A/D conversion is completed, the A/D conversion result is transferred into ADDRA.
3. Next, the second channel (AN1) is selected automatically and A/D conversion starts.
4. Conversion proceeds in the same way through the third channel (AN2).
5. When conversion of all selected channels (AN0 to AN2) is completed, the ADF flag is set to 1 and the ADST bit is cleared to 0.
6. If the ADIE bit is set to 1 at this time, an ADI interrupt is requested.

chanhvinh.com

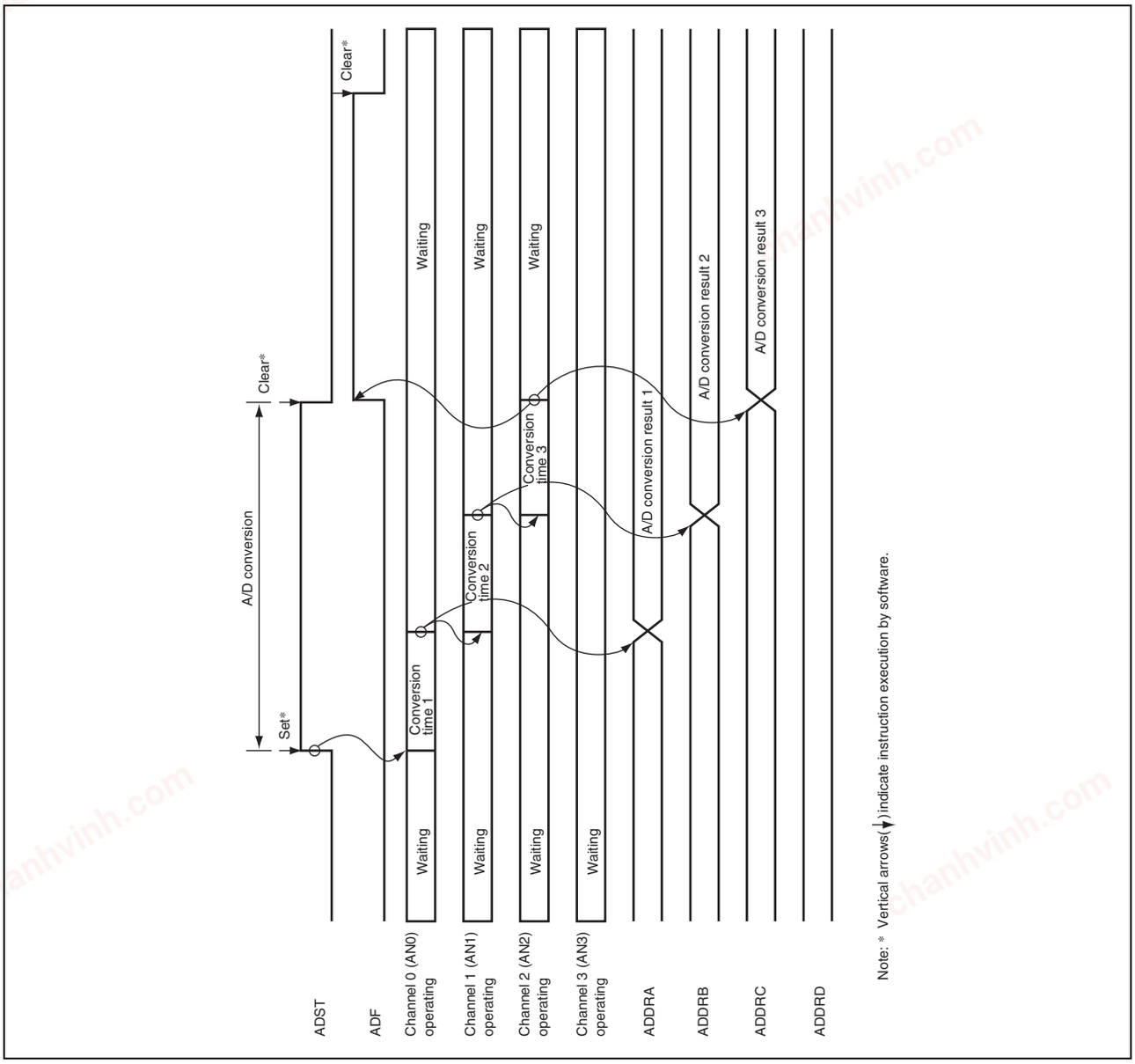


Figure 27.3 Example of A/D Converter Operation (Multi Mode, Three Channels (AN0 to AN2) Selected)

chanhvinh.com

27.4.3 Scan Mode

Scan mode is useful for monitoring analog inputs in a group of one or more channels at all times. In scan mode, A/D conversion is performed sequentially for a maximum of eight specified analog input channels, as follows:

1. A/D conversion starts from the analog input channel with the lowest number (e.g. AN0, AN1, …, AN3) when the ADST bit in ADCSR is set to 1 by software, the multi-function timer pulse unit 2, or external trigger input.
2. When A/D conversion is completed on each channel, the A/D conversion result is sequentially transferred to the A/D data register corresponding to that channel.
3. After A/D conversion on all selected channels has completed, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated. The A/D converter starts A/D conversion again from the channel with the lowest number.
4. The ADST bit is not cleared automatically, so steps 2. and 3. are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion halts and the A/D converter becomes idle.
   The ADF bit is cleared by reading ADF while ADF = 1, then writing 0 to the ADF bit.

When the operating mode or analog input channel selection must be changed during A/D conversion, to prevent incorrect operation, first clear the ADST bit to 0 to halt A/D conversion. After making the necessary changes, set the ADST bit to 1. A/D conversion will start again from the first channel in the group. The ADST bit can be set at the same time as the mode or channel selection is changed.
Typical operations when three channels (AN0 to AN2) are selected in scan mode are described as follows. Figure 27.4 shows a timing diagram for this example.

1. Scan mode is selected (MDS[2] = 1, MDS[1] = 1), analog input channels AN0 to AN2 are selected (CH[2:0] = 010), and A/D conversion is started (ADST = 1).
2. A/D conversion of the first channel (AN0) starts. When A/D conversion is completed, the A/D conversion result is transferred into ADDRA.
3. Next, the second channel (AN1) is selected automatically and A/D conversion starts.
4. Conversion proceeds in the same way through the third channel (AN2).
5. When conversion of all the selected channels (AN0 to AN2) is completed, the ADF flag is set to 1 and conversion of the first channel (AN0) starts again. If the ADIE bit is set to 1 at this time, an ADI interrupt is requested after A/D conversion of the third channel.
6. The ADST bit is not cleared automatically, so steps 2. to 4. are repeated as long as the ADST bit remains set to 1. When steps 2. to 4. are repeated, the ADF flag is kept to 1. When the ADST bit is cleared to 0, A/D conversion stops.
   The ADF bit is cleared by reading ADF while ADF = 1, then writing 0 to the ADF bit.

If both the ADF flag and ADIE bit are set to 1 while steps 2. to 4. are repeated, an ADI interrupt is requested at all times. To generate an interrupt on completing conversion of the third channel, clear the ADF bit to 0 after an interrupt is requested.

chanhvinh.com

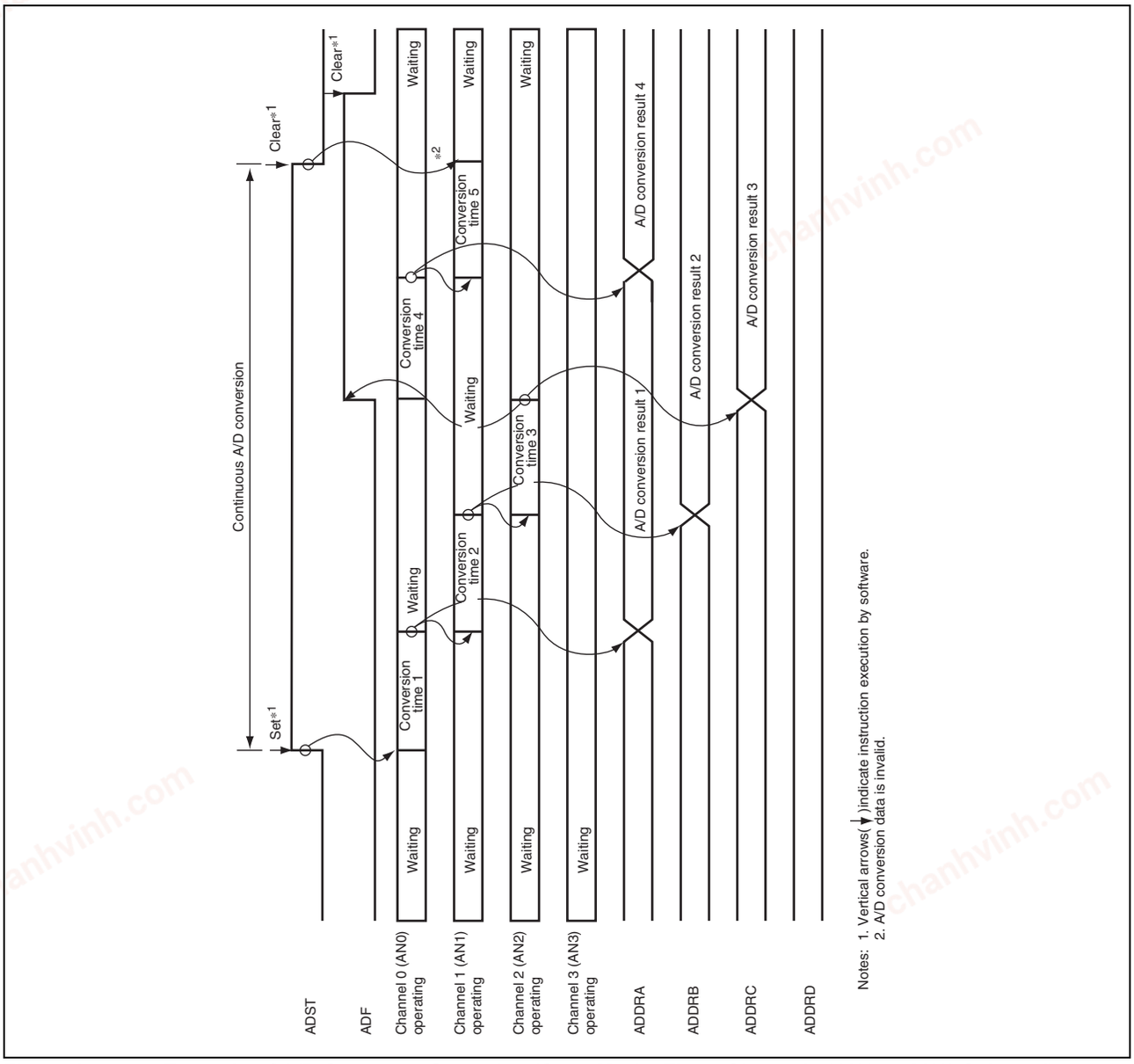


Figure 27.4 Example of A/D Converter Operation (Scan Mode, Three Channels (AN0 to AN2) Selected)

chanhvinh.com

27.4.4 A/D Converter Activation by External Trigger or Multi-Function Timer Pulse Unit 2

The A/D converter can be independently activated by an external trigger or an A/D conversion request from the multi-function timer pulse unit 2. To activate the A/D converter by an external trigger or the multi-function timer pulse unit 2, set the A/D trigger enable bits (TRGS[3:0]). When an external trigger or an A/D conversion request from the multi-function timer pulse unit 2 is generated with this bit setting, the ADST bit is set to 1 to start A/D conversion. The channel combination is determined by bits CH2 to CH0 in ADCSR. The timing from setting of the ADST bit until the start of A/D conversion is the same as when 1 is written to the ADST bit by software.

27.4.5 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the analog input at the A/D conversion start delay time (t_D) after the ADST bit in ADCSR is set to 1, then starts conversion. Figure 27.5 shows the A/D conversion timing. Table 27.5 indicates the A/D conversion time.

As indicated in Figure 27.5, the A/D conversion time (t_{CONV}) includes t_D and the input sampling time(t_{SPL}). The length of t_D varies depending on the timing of the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in Table 27.5.

In multi mode and scan mode, the values given in Table 27.5 apply to the first conversion. In the second and subsequent conversions, time is the values given in Table 27.6.

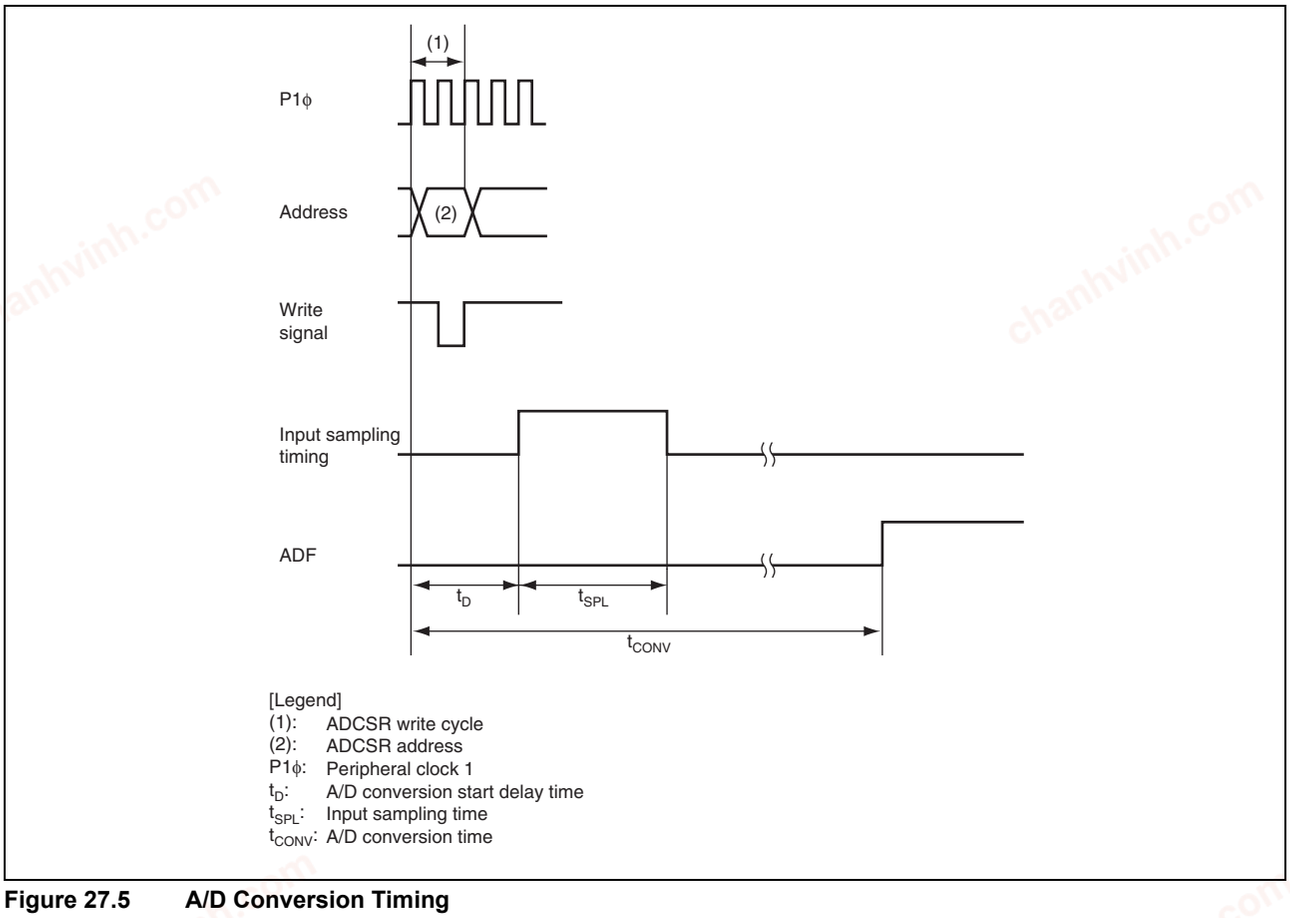


Figure 27.5 A/D Conversion Timing

chanhvinh.com

Table 27.5 A/D Conversion Time (Single Mode)

		CKS1 = 0						CKS1 = 1					
		CKS0 = 0			CKS0 = 1			CKS0 = 0			CKS0 = 1		
Item	**Symbol**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**
A/D conversion start delay time	t_D	9	—	14	10	—	16	11	—	18	12	—	20
Input sampling time	t_{SPL}	—	78	—	—	91	—	—	104	—	—	117	—
A/D conversion time	t_{CONV}	251	—	256	292	—	298	333	—	340	374	—	382

Note: Values in the table are represented in terms of t_{cyc}. t_{cyc} indicates a cycle time of the peripheral clock 1 (P1φ).

Table 27.6 A/D Conversion Time (Multi Mode and Scan Mode)

CKS1	CKS0	Conversion Time (t_{cyc})
0	0	240 (constant)
	1	280 (constant)
1	0	320 (constant)
	1	360 (constant)

Note: Values in the table are represented in terms of t_{cyc}. t_{cyc} indicates a cycle time of the peripheral clock 1 (P1φ).

chanhvinh.com

27.4.6 External Trigger Input Timing

A/D conversion can also be externally triggered. When the TRGS[3:0] bits in ADCSR are set to B'1001, an external trigger is input to the $\overline{\text{ADTRG}}$ pin. The ADST bit in ADCSR is set to 1 at the falling edge of the $\overline{\text{ADTRG}}$ pin, thus starting A/D conversion. Other operations, regardless of the operating mode, are the same as when the ADST bit has been set to 1 by software. Figure 27.6 shows the timing.

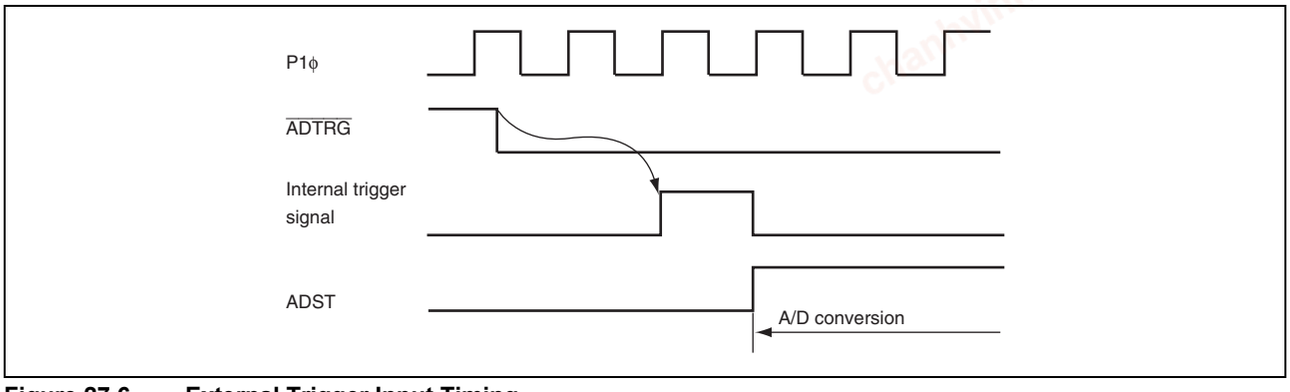


Figure 27.6 External Trigger Input Timing

chanhvinh.com

27.5 Interrupt Sources and DMA Transfer Request

The A/D converter generates an A/D conversion end interrupt (ADI) and an over-limit interrupt (LMTI). The direct memory access controller can be activated by an ADI interrupt request depending on the setting of the direct memory access controller.

Table 27.7 shows the relationship between interrupt sources and the DMA transfer request.

Table 27.7 Relation between Interrupt Sources and DMA Transfer Request

Name	Interrupt Source	Interrupt Flag	Direct Memory Access Controller Activation
ADI	A/D conversion end	ADF in ADCSR	Possible
LMTI	Conversion result exceeding the limit value	HOVRA to HOVRH and LUDRA to LUDRH in ADCMPSR	Impossible

(a) ADI Interrupt

The A/D converter generates an A/D conversion end interrupt (ADI) at the end of A/D conversion. An ADI interrupt request is generated if the ADIE bit is set to 1 when the ADF bit in ADCSR is set to 1 on completion of A/D conversion. Note that the direct memory access controller can be activated by an ADI interrupt depending on the setting of the direct memory access controller. In this case, an interrupt is not issued to the CPU. If the setting to activate the direct memory access controller has not been made, an interrupt request is sent to the CPU. Having the converted data read by the direct memory access controller in response to an ADI interrupt enables continuous conversion to be achieved without imposing a load on software.

In single mode, set the direct memory access controller so that DMA transfer initiated by an ADI interrupt only proceeds once (single transfer mode). In the case of A/D conversion on multiple channels in scan mode or multi mode, setting the number of DMA transfers to one causes DMA transfer to finish after transferring only one channel of data. To make the direct memory access controller transfer all conversion data, set the ADDR where A/D conversion data is stored as the transfer source address, and the number of converted channels as the number of transfers (the number of DMA transactions in block transfer mode).

For details, see section 9, Direct Memory Access Controller.

When the direct memory access controller is activated by ADI, the ADF bit in ADCSR is automatically cleared to 0 when data is transferred by the direct memory access controller.

(b) LMTI Interrupt

An A/D conversion result is compared with the upper or lower limit value on each channel, and if the result exceeds the limit, the over-limit interrupt (LMTI) is generated. While the HLMEN bit in ADCMPER is 1, a conversion result is compared with the upper limit in ADCMPH the instant the conversion value is determined. When the value exceeds the upper limit, the corresponding bit in ADCMPSR is set to 1, which causes an LMTI interrupt to be generated. In the same way, while the LLMEN bit in ADCMPER is 1, a conversion result is compared with the lower limit. If it exceeds the limit, an LMTI interrupt request is generated.

The ADCMPSR register can be referenced in the LMTI interrupt handling to identify the channel causing over-limit value and to know whether the result has exceeded the upper limit value or the lower one.

chanhvinh.com

27.6 Definitions of A/D Conversion Accuracy

The A/D converter compares an analog value input from an analog input channel with its analog reference value and converts it to 12-bit digital data. The absolute accuracy of this A/D conversion is the deviation between the input analog value and the output digital value. It includes the following errors:

- Offset error
- Full-scale error
- Quantization error
- Nonlinearity error

These four error quantities are explained below with reference to Figure 27.7. In the figure, the 12-bit A/D converter is illustrated as the 3-bit A/D converter for explanation. Offset error is the deviation between actual and ideal A/D conversion characteristics when the digital output value changes from the minimum (zero voltage) B'000000000000 (000 in the figure) to B'000000000001 (001 in the figure) (Figure 27.7, item (1)). Full-scale error is the deviation between actual and ideal A/D conversion characteristics when the digital output value changes from B'111111111110 (110 in the figure) to the maximum B'111111111111 (111 in the figure)(Figure 27.7, item (2)). Quantization error is the intrinsic error of the A/D converter and is expressed as 1/2 LSB (Figure 27.7, item (3)). Nonlinearity error is the deviation between actual and ideal A/D conversion characteristics between zero voltage and full-scale voltage (Figure 27.7, item (4)). Note that it does not include offset, full-scale, or quantization error.

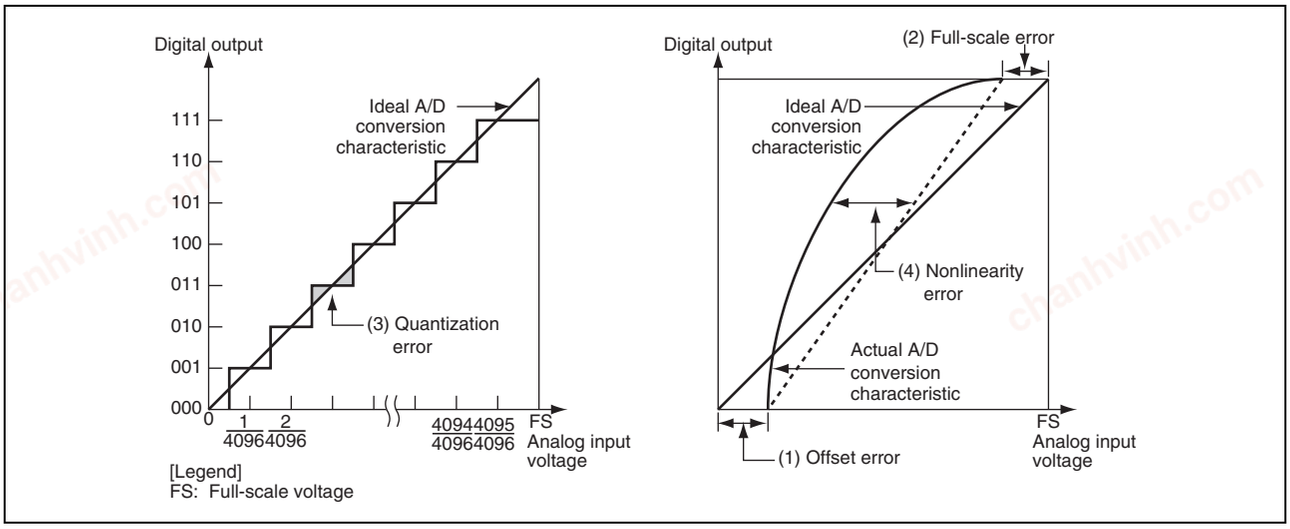


Figure 27.7 Definitions of A/D Conversion Accuracy

chanhvinh.com

27.7 Usage Notes

When using the A/D converter, note the following points.

27.7.1 Module Standby Mode Setting

Operation of the A/D converter can be disabled or enabled using the standby control register. The initial setting is for operation of the A/D converter to be halted. Register access is enabled by clearing module standby mode. For details, see section 42, Power-Down Modes. The wait time (20 µs) is required after release from the module standby state. After the wait time has elapsed, perform conversion by the A/D converter.

27.7.2 Setting Analog Input Voltage

Using the LSI outside the following voltage ranges may impair the LSI reliability.

1. Analog input range
   During A/D conversion, voltages on the analog input pins ANn should not go beyond the following range: AVss ≤ ANn ≤ AVcc (n = 0 to 7).
2. AVcc and AVss input voltages
   Input voltages AVcc and AVss should be PVcc − 0.3 V ≤ AVcc ≤ PVcc and AVss = Vss. Do not leave the AVcc and AVss pins open even when the A/D converter is not in use and in software standby mode. When not in use, connect AVcc to the power supply (PVcc) and AVss to the ground (Vss).
3. Setting range of AVref input voltage
   Set the reference voltage range of the AVref pin as 3.0 V ≤ AVref ≤ AVcc.

27.7.3 Notes on Board Design

In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and layout in which digital circuit signal lines and analog circuit signal lines cross or are in close proximity should be avoided as far as possible. Failure to do so may result in incorrect operation of the analog circuitry due to inductance, adversely affecting A/D conversion values.
Digital circuitry must be isolated from the analog input signals (AN0 to AN7), analog reference voltage (AVref), and analog power supply (AVcc) by the analog ground (AVss). Also, the analog ground (AVss) should be connected at one point to a stable digital ground (Vss) on the board.

chanhvinh.com

27.7.4 Processing of Analog Input Pins

To prevent damage from voltage surges at the analog input pins (AN0 to AN7), connect an input protection circuit like the one shown in Figure 27.8. The circuit shown also includes a CR filter to suppress noise. This circuit is shown as an example; the circuit constants should be selected according to actual application conditions.

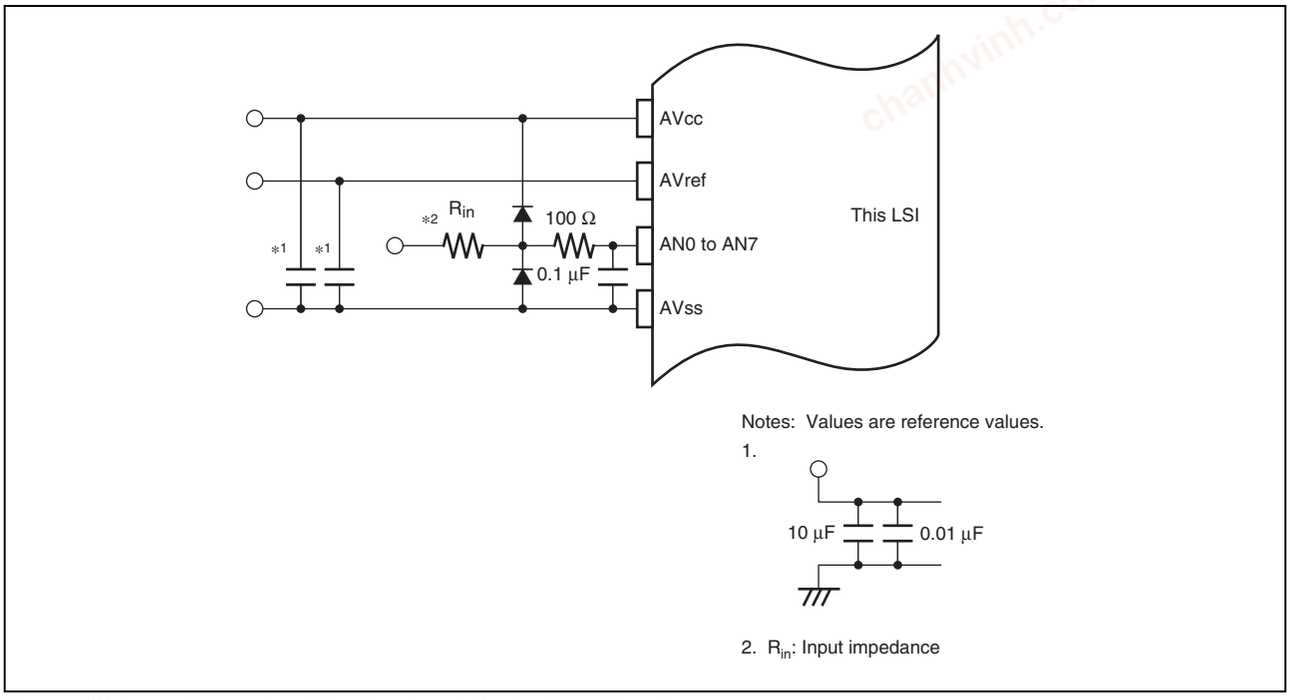


Figure 27.8 Example of Analog Input Protection Circuit

chanhvinh.com

27.7.5 Permissible Signal Source Impedance

This LSI's analog input is designed such that conversion precision is guaranteed for an input signal for which the signal source impedance is 3 kΩ or less. This specification is provided to enable the A/D converter's sample-and-hold circuit input capacitance to be charged within the sampling time; if the sensor output impedance exceeds 3 kΩ, charging may be insufficient and it may not be possible to guarantee A/D conversion precision. However, for A/D conversion in single mode with a large capacitance provided externally for A/D conversion in single mode, the input load will essentially comprise only the internal input resistance of 1 kΩ, and the signal source impedance is ignored. However, as a low-pass filter effect is obtained in this case, it may not be possible to follow an analog signal with a large differential coefficient (e.g., 5 mV/µs or greater) (see Figure 27.9). When converting a high-speed analog signal or performing A/D conversion in scan mode, a low-impedance buffer should be inserted.

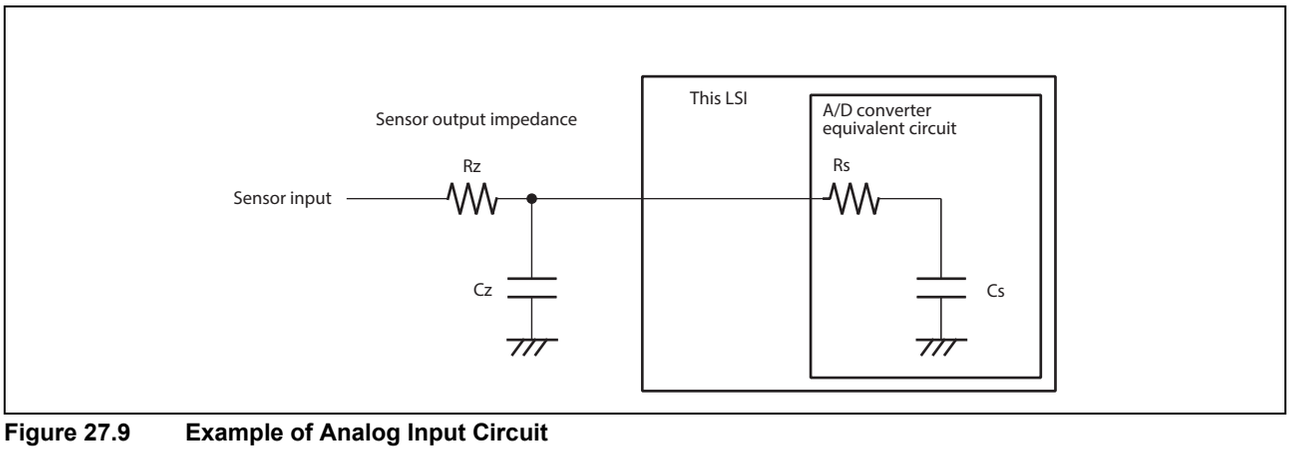


Figure 27.9 Example of Analog Input Circuit

Table 27.8 Analog Input Pin Ratings

Item	Symbol	Min.	Max.	Unit
Permissible Signal Source Impedance	Rz	—	3	kΩ
Low-Pass Filter	Cz	—	0.1	µF
Equivalent Circuit of A/D Converter	Rs	—	1	kΩ
	Cs	—	20	pF

Note: Values are reference values.

27.7.6 Influences on Absolute Precision

Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute precision. Be sure to connect AVss, etc. to an electrically stable GND.

Care is also required to insure that filter circuits do not communicate with digital signals on the mounting board (i.e., acting as antennas).

27.7.7 Usage Note on Port Pins

The analog input pins (AN0 to AN7) are multiplexed with general-purpose I/O port pin functions (pins P1_8 to P1_15) and the latter can be used as digital inputs. Do not use these general-purpose I/O port pins as digital inputs while the A/D converter is in use with 12-bit precision.

chanhvinh.com

28. USB2.0 Host/Function Module

This LSI includes a two-channel USB 2.0 host/function module. This module is a USB controller which provides capabilities as a USB host controller and USB function controller.
This module supports high-speed transfer, full-speed transfer, and low-speed transfer defined by USB (Universal Serial Bus) Specification 2.0, when used as the host controller, and supports high-speed transfer and full-speed transfer defined by USB (Universal Serial Bus) Specification 2.0, when used as the function controller.
This module supports all of the transfer types defined by the USB Specification. This module has an 8-Kbyte buffer memory for data transfer, providing a maximum of 16 pipes. Any endpoint numbers can be assigned to PIPE1 to PIPE15, based on the peripheral devices or user system for communication.

28.1 Features

(1) Host Controller and Function Controller Supporting USB High-Speed Operation

- The USB host controller and USB function controller are incorporated.
- The USB host controller and USB function controller can be switched by register settings.
- On-chip USB transceiver

(2) All Types of USB Transfers Supported

All types of USB transfers including isochronous transfer are supported.

- Control transfer
- Bulk transfer
- Interrupt transfer (high bandwidth transfers not supported)
- Isochronous transfer (high bandwidth transfers not supported)

(3) Internal Bus Interfaces

- Two DMA interfaces available for each channel

(4) Pipe Configuration

- Up to 8 Kbytes of buffer memory for USB communications are supported for each channel
- Up to sixteen pipes can be selected for each channel (including the default control pipe)
- Programmable pipe configuration
- Endpoint numbers can be assigned flexibly to PIPE1 to PIPE15.
- Transfer conditions that can be set for each pipe:

PIPE0:	Control transfer, 256-byte fixed single buffer
PIPE1 and PIPE2:	Bulk transfer or isochronous transfer can be selected, continuous transfer mode, programmable buffer size (up to 2 Kbytes: double buffer can be specified)
PIPE3 to PIPE5:	Bulk transfer, continuous transfer mode, programmable buffer size (up to 2 Kbytes: double buffer can be specified)
PIPE6 to PIPE8:	Interrupt transfer, 64-byte fixed single buffer
PIPE9:	Bulk transfer (only when the function controller mode is selected) Interrupt transfer (only when the host controller mode is selected), programmable buffer size (up to 2 Kbytes: double buffer can be specified only when bulk transfer has been selected)
PIPE10	Bulk transfer or interrupt transfer can be selected (only when the function controller mode is selected), programmable buffer size (up to 2 Kbytes: double buffer can be specified only when bulk transfer has been selected)

chanhvinh.com

PIPE11 to PIPE15: Bulk transfer (only when the function controller mode is selected), programmable buffer size (up to 2 Kbytes: double buffer can be specified)

(5) Features of the USB Host Controller

- High-speed transfer (480 Mbps), full-speed transfer (12 Mbps), and low-speed transfer (1.5 Mbps) are supported.
- Communications with multiple peripheral devices connected via a single HUB
- Automatic response to the reset handshake
- Automatic scheduling for SOF and packet transmissions
- Programmable intervals for isochronous and interrupt transfers

(6) Features of the USB Function Controller

- High-speed transfer (480 Mbps) and full-speed transfer (12 Mbps) are supported.
- Automatic recognition of high-speed operation or full-speed operation based on automatic response to the reset handshake
- Control transfer stage monitoring function
- Device state monitoring function
- Auto response function for SET_ADDRESS request
- NAK response interrupt function (NRDY)
- SOF interpolation function

(7) Other Features

- Transfer ending function using transaction count
- BRDY interrupt event notification timing change function (BFRE)
- Function that automatically clears the buffer memory after the data for the pipe specified at the DnFIFO port has been read (DCLRM)
- NAK setting function for response PID generated by end of transfer (SHTNAK)

chanhvinh.com

28.2 Input/Output Pins

Table 28.1 shows the pin configuration.

Table 28.1 Pin Configuration

Channel	Name	Pin Name	I/O	Function
0	USB D+ data	DP0	Input/ Output	The D+ input/output pin of the on-chip USB transceiver Connect this pin to the D+ pin of the USB bus.
	USB D- data	DM0	Input/ Output	The D- input/output pin of the on-chip USB transceiver Connect this pin to the D- pin of the USB bus.
	VBUS input	VBUS0	Input	A pin for monitoring connection of the USB cable Connect this pin to the Vbus pin of the USB bus so that it can detect connection and disconnection of the Vbus pin. When it is not connected to the Vbus pin of the USB, It should be fixed to 5 V. Also supply 5 V to this pin when the host controller mode is selected. Note: This module is not capable of supplying Vbus power to connected peripheral devices.
1	USB D+ data	DP1	Input/ Output	The D+ input/output pin of the on-chip USB transceiver Connect this pin to the D+ pin of the USB bus.
	USB D- data	DM1	Input/ Output	The D- input/output pin of the on-chip USB transceiver Connect this pin to the D- pin of the USB bus.
	VBUS input	VBUS1	Input	A pin for monitoring connection of the USB cable Connect this pin to the Vbus pin of the USB bus so that it can detect connection and disconnection of the Vbus pin. When it is not connected to the Vbus pin of the USB, It should be fixed to 5 V. Also supply 5 V to this pin when the host controller mode is selected. Note: This module is not capable of supplying Vbus power to connected peripheral devices.
Common	Reference input	REFRIN	Input	A pin for connecting the reference resistor Connect this pin to the USBAPVss pin through a resistor with a value of 5.6 kΩ ± 1%. (QFP package) Connect this pin to the Vss pin through a resistor with a value of 5.6 kΩ ± 1%. (BGA package)
	USB crystal oscillator/external clock	USB_X1	Input	Connect this pin to the crystal oscillator for USB. An external clock can also be input to the USB_X1 pin.
		USB_X2	Output	
	Transceiver digital core power*	USBDVcc	Input	Power supply for the module's digital core
	Transceiver digital core ground*	USBDVss	Input	Ground for the module's digital core
	Transceiver digital pin power source*	USBDPVcc	Input	Power supply for pins
	Transceiver digital pin ground*	USBDPVss	Input	Ground for pins
	Transceiver analog pin power source	USBAPVcc	Input	Power supply for pins
	Transceiver analog pin ground*	USBAPVss	Input	Ground for pins
	Transceiver analog core power	USBAVcc	Input	Power supply for the module's digital core
	Transceiver analog core ground*	USBAVss	Input	Ground for the module's digital core
	Power source for the UTMI module*	USBUVcc	Input	Power supply for operation at 480 MHz
	Ground for the UTMI module*	USBUVss	Input	Power supply for operation at 480 MHz

Note: *These pins are not present on products in the BGA package.

chanhvinh.com

28.3 Register Descriptions

Table 28.2 shows the register configuration of this module.

Table 28.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Address	Access Size
0	System configuration control register_0	SYSCFG0_0	R/W	H'E801 0000	16
	CPU bus wait setting register_0	BUSWAIT_0	R/W	H'E801 0002	16
	System configuration status register_0	SYSSTS0_0	R	H'E801 0004	16
	Device state control register 0_0	DVSTCTR0_0	R/W	H'E801 0008	16
	Test mode register_0	TESTMODE_0	R/W	H'E801 000C	16
	DMA0-FIFO bus configuration register_0	D0FBCFG_0	R/W	H'E801 0010	16
	DMA1-FIFO bus configuration register_0	D1FBCFG_0	R/W	H'E801 0012	16
	CFIFO port register_0	CFIFO_0	R/W	H'E801 0014	8, 16, 32
	D0FIFO port register_0	D0FIFO_0	R/W	H'E801 0018	8, 16, 32
	D1FIFO port register_0	D1FIFO_0	R/W	H'E801 001C	8, 16, 32
	CFIFO port select register_0	CFIFOSEL_0	R/W	H'E801 0020	16
	CFIFO port control register_0	CFIFOCTR_0	R/W	H'E801 0022	16
	D0FIFO port select register_0	D0FIFOSEL_0	R/W	H'E801 0028	16
	D0FIFO port control register_0	D0FIFOCTR_0	R/W	H'E801 002A	16
	D1FIFO port select register_0	D1FIFOSEL_0	R/W	H'E801 002C	16
	D1FIFO port control register_0	D1FIFOCTR_0	R/W	H'E801 002E	16
	Interrupt enable register 0_0	INTENB0_0	R/W	H'E801 0030	16
	Interrupt enable register 1_0	INTENB1_0	R/W	H'E801 0032	16
	BRDY interrupt enable register_0	BRDYENB_0	R/W	H'E801 0036	16
	NRDY interrupt enable register_0	NRDYENB_0	R/W	H'E801 0038	16
	BEMP interrupt enable register_0	BEMPENB_0	R/W	H'E801 003A	16
	SOF output configuration register_0	SOFCFG_0	R/W	H'E801 003C	16
	Interrupt status register 0_0	INTSTS0_0	R/W	H'E801 0040	16
	Interrupt status register 1_0	INTSTS1_0	R/W	H'E801 0042	16
	BRDY interrupt status register_0	BRDYSTS_0	R/W	H'E801 0046	16
	NRDY interrupt status register_0	NRDYSTS_0	R/W	H'E801 0048	16
	BEMP interrupt status register_0	BEMPSTS_0	R/W	H'E801 004A	16
	Frame number register_0	FRMNUM_0	R/W	H'E801 004C	16
	µFrame number register_0	UFRMNUM_0	R	H'E801 004E	16
	USB address register_0	USBADDR_0	R	H'E801 0050	16
	USB request type register_0	USBREQ_0	R/W	H'E801 0054	16
	USB request value register_0	USBVAL_0	R/W	H'E801 0056	16
	USB request index register_0	USBINDX_0	R/W	H'E801 0058	16
	USB request length register_0	USBLENG_0	R/W	H'E801 005A	16
	DCP configuration register_0	DCPCFG_0	R/W	H'E801 005C	16
	DCP maximum packet size register_0	DCPMAXP_0	R/W	H'E801 005E	16

chanhvinh.com

Table 28.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Address	Access Size
0	DCP control register_0	DCPCTR_0	R/W	H'E801 0060	16
	Pipe window select register_0	PIPESEL_0	R/W	H'E801 0064	16
	Pipe configuration register_0	PIPECFG_0	R/W	H'E801 0068	16
	Pipe buffer setting register_0	PIPEBUF_0	R/W	H'E801 006A	16
	Pipe maximum packet size register_0	PIPEMAXP_0	R/W	H'E801 006C	16
	Pipe timing control register_0	PIPEPERI_0	R/W	H'E801 006E	16
	Pipe 1 control register_0	PIPE1CTR_0	R/W	H'E801 0070	16
	Pipe 2 control register_0	PIPE2CTR_0	R/W	H'E801 0072	16
	Pipe 3 control register_0	PIPE3CTR_0	R/W	H'E801 0074	16
	Pipe 4 control register_0	PIPE4CTR_0	R/W	H'E801 0076	16
	Pipe 5 control register_0	PIPE5CTR_0	R/W	H'E801 0078	16
	Pipe 6 control register_0	PIPE6CTR_0	R/W	H'E801 007A	16
	Pipe 7 control register_0	PIPE7CTR_0	R/W	H'E801 007C	16
	Pipe 8 control register_0	PIPE8CTR_0	R/W	H'E801 007E	16
	Pipe 9 control register_0	PIPE9CTR_0	R/W	H'E801 0080	16
	Pipe A control register_0	PIPEACTR_0	R/W	H'E801 0082	16
	Pipe B control register_0	PIPEBCTR_0	R/W	H'E801 0084	16
	Pipe C control register_0	PIPECCTR_0	R/W	H'E801 0086	16
	Pipe D control register_0	PIPEDCTR_0	R/W	H'E801 0088	16
	Pipe E control register_0	PIPEECTR_0	R/W	H'E801 008A	16
	Pipe F control register_0	PIPEFCTR_0	R/W	H'E801 008C	16
	Pipe 1 transaction counter enable register_0	PIPE1TRE_0	R/W	H'E801 0090	16
	Pipe 1 transaction counter register_0	PIPE1TRN_0	R/W	H'E801 0092	16
	Pipe 2 transaction counter enable register_0	PIPE2TRE_0	R/W	H'E801 0094	16
	Pipe 2 transaction counter register_0	PIPE2TRN_0	R/W	H'E801 0096	16
	Pipe 3 transaction counter enable register_0	PIPE3TRE_0	R/W	H'E801 0098	16
	Pipe 3 transaction counter register_0	PIPE3TRN_0	R/W	H'E801 009A	16
	Pipe 4 transaction counter enable register_0	PIPE4TRE_0	R/W	H'E801 009C	16
	Pipe 4 transaction counter register_0	PIPE4TRN_0	R/W	H'E801 009E	16
	Pipe 5 transaction counter enable register_0	PIPE5TRE_0	R/W	H'E801 00A0	16
	Pipe 5 transaction counter register_0	PIPE5TRN_0	R/W	H'E801 00A2	16
	Pipe B transaction counter enable register_0	PIPEBTRE_0	R/W	H'E801 00A4	16
	Pipe B transaction counter register_0	PIPEBTRN_0	R/W	H'E801 00A6	16
	Pipe C transaction counter enable register_0	PIPECTRE_0	R/W	H'E801 00A8	16
	Pipe C transaction counter register_0	PIPECTRN_0	R/W	H'E801 00AA	16
	Pipe D transaction counter enable register_0	PIPEDTRE_0	R/W	H'E801 00AC	16
	Pipe D transaction counter register_0	PIPEDTRN_0	R/W	H'E801 00AE	16
	Pipe E transaction counter enable register_0	PIPEETRE_0	R/W	H'E801 00B0	16
	Pipe E transaction counter register_0	PIPEETRN_0	R/W	H'E801 00B2	16
	Pipe F transaction counter enable register_0	PIPEFTRE_0	R/W	H'E801 00B4	16
	Pipe F transaction counter register_0	PIPEFTRN_0	R/W	H'E801 00B6	16
	Pipe 9 transaction counter enable register_0	PIPE9TRE_0	R/W	H'E801 00B8	16
	Pipe 9 transaction counter register_0	PIPE9TRN_0	R/W	H'E801 00BA	16

chanhvinh.com

Table 28.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Address	Access Size
0	Pipe A transaction counter enable register_0	PIPEATRE_0	R/W	H'E801 00BC	16
	Pipe A transaction counter register_0	PIPEATRN_0	R/W	H'E801 00BE	16
	Device address 0 configuration register_0	DEVADD0_0	R/W	H'E801 00D0	16
	Device address 1 configuration register_0	DEVADD1_0	R/W	H'E801 00D2	16
	Device address 2 configuration register_0	DEVADD2_0	R/W	H'E801 00D4	16
	Device address 3 configuration register_0	DEVADD3_0	R/W	H'E801 00D6	16
	Device address 4 configuration register_0	DEVADD4_0	R/W	H'E801 00D8	16
	Device address 5 configuration register_0	DEVADD5_0	R/W	H'E801 00DA	16
	Device address 6 configuration register_0	DEVADD6_0	R/W	H'E801 00DC	16
	Device address 7 configuration register_0	DEVADD7_0	R/W	H'E801 00DE	16
	Device address 8 configuration register_0	DEVADD8_0	R/W	H'E801 00E0	16
	Device address 9 configuration register_0	DEVADD9_0	R/W	H'E801 00E2	16
	Device address A configuration register_0	DEVADDA_0	R/W	H'E801 00E4	16
	Suspend mode register_0	SUSPMODE_0	R/W	H'E801 0102	16
	D0FIFO continuous transfer port register 0_0	D0FIFOB0_0	R/W	H'E801 0160	32
	D0FIFO continuous transfer port register 1_0	D0FIFOB1_0	R/W	H'E801 0164	32
	D0FIFO continuous transfer port register 2_0	D0FIFOB2_0	R/W	H'E801 0168	32
	D0FIFO continuous transfer port register 3_0	D0FIFOB3_0	R/W	H'E801 016C	32
	D0FIFO continuous transfer port register 4_0	D0FIFOB4_0	R/W	H'E801 0170	32
	D0FIFO continuous transfer port register 5_0	D0FIFOB5_0	R/W	H'E801 0174	32
	D0FIFO continuous transfer port register 6_0	D0FIFOB6_0	R/W	H'E801 0178	32
	D0FIFO continuous transfer port register 7_0	D0FIFOB7_0	R/W	H'E801 017C	32
	D1FIFO continuous transfer port register 0_0	D1FIFOB0_0	R/W	H'E801 0180	32
	D1FIFO continuous transfer port register 1_0	D1FIFOB1_0	R/W	H'E801 0184	32
	D1FIFO continuous transfer port register 2_0	D1FIFOB2_0	R/W	H'E801 0188	32
	D1FIFO continuous transfer port register 3_0	D1FIFOB3_0	R/W	H'E801 018C	32
	D1FIFO continuous transfer port register 4_0	D1FIFOB4_0	R/W	H'E801 0190	32
	D1FIFO continuous transfer port register 5_0	D1FIFOB5_0	R/W	H'E801 0194	32
	D1FIFO continuous transfer port register 6_0	D1FIFOB6_0	R/W	H'E801 0198	32
	D1FIFO continuous transfer port register 7_0	D1FIFOB7_0	R/W	H'E801 019C	32

chanhvinh.com

Table 28.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Address	Access Size
1	System configuration control register_1	SYSCFG0_1	R/W	H'E820 7000	16
	CPU bus wait setting register_1	BUSWAIT_1	R/W	H'E820 7002	16
	System configuration status register_1	SYSSTS0_1	R	H'E820 7004	16
	Device state control register 0_1	DVSTCTR0_1	R/W	H'E820 7008	16
	Test mode register_1	TESTMODE_1	R/W	H'E820 700C	16
	DMA0-FIFO bus configuration register_1	D0FBCFG_1	R/W	H'E820 7010	16
	DMA1-FIFO bus configuration register_1	D1FBCFG_1	R/W	H'E820 7012	16
	CFIFO port register_1	CFIFO_1	R/W	H'E820 7014	8, 16, 32
	D0FIFO port register_1	D0FIFO_1	R/W	H'E820 7018	8, 16, 32
	D1FIFO port register_1	D1FIFO_1	R/W	H'E820 701C	8, 16, 32
	CFIFO port select register_1	CFIFOSEL_1	R/W	H'E820 7020	16
	CFIFO port control register_1	CFIFOCTR_1	R/W	H'E820 7022	16
	D0FIFO port select register_1	D0FIFOSEL_1	R/W	H'E820 7028	16
	D0FIFO port control register_1	D0FIFOCTR_1	R/W	H'E820 702A	16
	D1FIFO port select register_1	D1FIFOSEL_1	R/W	H'E820 702C	16
	D1FIFO port control register_1	D1FIFOCTR_1	R/W	H'E820 702E	16
	Interrupt enable register 0_1	INTENB0_1	R/W	H'E820 7030	16
	Interrupt enable register 1_1	INTENB1_1	R/W	H'E820 7032	16
	BRDY interrupt enable register_1	BRDYENB_1	R/W	H'E820 7036	16
	NRDY interrupt enable register_1	NRDYENB_1	R/W	H'E820 7038	16
	BEMP interrupt enable register_1	BEMPENB_1	R/W	H'E820 703A	16
	SOF output configuration register_1	SOFCFG_1	R/W	H'E820 703C	16
	Interrupt status register 0_1	INTSTS0_1	R/W	H'E820 7040	16
	Interrupt status register 1_1	INTSTS1_1	R/W	H'E820 7042	16
	BRDY interrupt status register_1	BRDYSTS_1	R/W	H'E820 7046	16
	NRDY interrupt status register_1	NRDYSTS_1	R/W	H'E820 7048	16
	BEMP interrupt status register_1	BEMPSTS_1	R/W	H'E820 704A	16
	Frame number register_1	FRMNUM_1	R/W	H'E820 704C	16
	µFrame number register_1	UFRMNUM_1	R	H'E820 704E	16

chanhvinh.com

Table 28.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Address	Access Size
1	USB address register_1	USBADDR_1	R	H'E820 7050	16
	USB request type register_1	USBREQ_1	R/W	H'E820 7054	16
	USB request value register_1	USBVAL_1	R/W	H'E820 7056	16
	USB request index register_1	USBINDX_1	R/W	H'E820 7058	16
	USB request length register_1	USBLENG_1	R/W	H'E820 705A	16
	DCP configuration register_1	DCPCFG_1	R/W	H'E820 705C	16
	DCP maximum packet size register_1	DCPMAXP_1	R/W	H'E820 705E	16
	DCP control register_1	DCPCTR_1	R/W	H'E820 7060	16
	Pipe window select register_1	PIPESEL_1	R/W	H'E820 7064	16
	Pipe configuration register_1	PIPECFG_1	R/W	H'E820 7068	16
	Pipe buffer setting register_1	PIPEBUF_1	R/W	H'E820 706A	16
	Pipe maximum packet size register_1	PIPEMAXP_1	R/W	H'E820 706C	16
	Pipe timing control register_1	PIPEPERI_1	R/W	H'E820 706E	16
	Pipe 1 control register_1	PIPE1CTR_1	R/W	H'E820 7070	16
	Pipe 2 control register_1	PIPE2CTR_1	R/W	H'E820 7072	16
	Pipe 3 control register_1	PIPE3CTR_1	R/W	H'E820 7074	16
	Pipe 4 control register_1	PIPE4CTR_1	R/W	H'E820 7076	16
	Pipe 5 control register_1	PIPE5CTR_1	R/W	H'E820 7078	16
	Pipe 6 control register_1	PIPE6CTR_1	R/W	H'E820 707A	16
	Pipe 7 control register_1	PIPE7CTR_1	R/W	H'E820 707C	16
	Pipe 8 control register_1	PIPE8CTR_1	R/W	H'E820 707E	16
	Pipe 9 control register_1	PIPE9CTR_1	R/W	H'E820 7080	16
	Pipe A control register_1	PIPEACTR_1	R/W	H'E820 7082	16
	Pipe B control register_1	PIPEBCTR_1	R/W	H'E820 7084	16
	Pipe C control register_1	PIPECCTR_1	R/W	H'E820 7086	16
	Pipe D control register_1	PIPEDCTR_1	R/W	H'E820 7088	16
	Pipe E control register_1	PIPEECTR_1	R/W	H'E820 708A	16
	Pipe F control register_1	PIPEFCTR_1	R/W	H'E820 708C	16
	Pipe 1 transaction counter enable register_1	PIPE1TRE_1	R/W	H'E820 7090	16
	Pipe 1 transaction counter register_1	PIPE1TRN_1	R/W	H'E820 7092	16
	Pipe 2 transaction counter enable register_1	PIPE2TRE_1	R/W	H'E820 7094	16
	Pipe 2 transaction counter register_1	PIPE2TRN_1	R/W	H'E820 7096	16
	Pipe 3 transaction counter enable register_1	PIPE3TRE_1	R/W	H'E820 7098	16
	Pipe 3 transaction counter register_1	PIPE3TRN_1	R/W	H'E820 709A	16
	Pipe 4 transaction counter enable register_1	PIPE4TRE_1	R/W	H'E820 709C	16
	Pipe 4 transaction counter register_1	PIPE4TRN_1	R/W	H'E820 709E	16
	Pipe 5 transaction counter enable register_1	PIPE5TRE_1	R/W	H'E820 70A0	16
	Pipe 5 transaction counter register_1	PIPE5TRN_1	R/W	H'E820 70A2	16
	Pipe B transaction counter enable register_1	PIPEBTRE_1	R/W	H'E820 70A4	16
	Pipe B transaction counter register_1	PIPEBTRN_1	R/W	H'E820 70A6	16
	Pipe C transaction counter enable register_1	PIPECTRE_1	R/W	H'E820 70A8	16
	Pipe C transaction counter register_1	PIPECTRN_1	R/W	H'E820 70AA	16
	Pipe D transaction counter enable register_1	PIPEDTRE_1	R/W	H'E820 70AC	16

chanhvinh.com

Table 28.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Address	Access Size
1	Pipe D transaction counter register_1	PIPEDTRN_1	R/W	H'E820 70AE	16
	Pipe E transaction counter enable register_1	PIPEETRE_1	R/W	H'E820 70B0	16
	Pipe E transaction counter register_1	PIPEETRN_1	R/W	H'E820 70B2	16
	Pipe F transaction counter enable register_1	PIPEFTRE_1	R/W	H'E820 70B4	16
	Pipe F transaction counter register_1	PIPEFTRN_1	R/W	H'E820 70B6	16
	Pipe 9 transaction counter enable register_1	PIPE9TRE_1	R/W	H'E820 70B8	16
	Pipe 9 transaction counter register_1	PIPE9TRN_1	R/W	H'E820 70BA	16
	Pipe A transaction counter enable register_1	PIPEATRE_1	R/W	H'E820 70BC	16
	Pipe A transaction counter register_1	PIPEATRN_1	R/W	H'E820 70BE	16
	Device address 0 configuration register_1	DEVADD0_1	R/W	H'E820 70D0	16
	Device address 1 configuration register_1	DEVADD1_1	R/W	H'E820 70D2	16
	Device address 2 configuration register_1	DEVADD2_1	R/W	H'E820 70D4	16
	Device address 3 configuration register_1	DEVADD3_1	R/W	H'E820 70D6	16
	Device address 4 configuration register_1	DEVADD4_1	R/W	H'E820 70D8	16
	Device address 5 configuration register_1	DEVADD5_1	R/W	H'E820 70DA	16
	Device address 6 configuration register_1	DEVADD6_1	R/W	H'E820 70DC	16
	Device address 7 configuration register_1	DEVADD7_1	R/W	H'E820 70DE	16
	Device address 8 configuration register_1	DEVADD8_1	R/W	H'E820 70E0	16
	Device address 9 configuration register_1	DEVADD9_1	R/W	H'E820 70E2	16
	Device address A configuration register_1	DEVADDA_1	R/W	H'E820 70E4	16
	Suspend mode register_1	SUSPMODE_1	R/W	H'E820 7102	16
	D0FIFO continuous transfer port register 0_1	D0FIFOB0_1	R/W	H'E820 7160	32
	D0FIFO continuous transfer port register 1_1	D0FIFOB1_1	R/W	H'E820 7164	32
	D0FIFO continuous transfer port register 2_1	D0FIFOB2_1	R/W	H'E820 7168	32
	D0FIFO continuous transfer port register 3_1	D0FIFOB3_1	R/W	H'E820 716C	32
	D0FIFO continuous transfer port register 4_1	D0FIFOB4_1	R/W	H'E820 7170	32
	D0FIFO continuous transfer port register 5_1	D0FIFOB5_1	R/W	H'E820 7174	32
	D0FIFO continuous transfer port register 6_1	D0FIFOB6_1	R/W	H'E820 7178	32
	D0FIFO continuous transfer port register 7_1	D0FIFOB7_1	R/W	H'E820 717C	32
	D1FIFO continuous transfer port register 0_1	D1FIFOB0_1	R/W	H'E820 7180	32
	D1FIFO continuous transfer port register 1_1	D1FIFOB1_1	R/W	H'E820 7184	32
	D1FIFO continuous transfer port register 2_1	D1FIFOB2_1	R/W	H'E820 7188	32
	D1FIFO continuous transfer port register 3_1	D1FIFOB3_1	R/W	H'E820 718C	32
	D1FIFO continuous transfer port register 4_1	D1FIFOB4_1	R/W	H'E820 7190	32
	D1FIFO continuous transfer port register 5_1	D1FIFOB5_1	R/W	H'E820 7194	32
	D1FIFO continuous transfer port register 6_1	D1FIFOB6_1	R/W	H'E820 7198	32
	D1FIFO continuous transfer port register 7_1	D1FIFOB7_1	R/W	H'E820 719C	32

chanhvinh.com

28.3.1 System Configuration Control Register (SYSCFG0)

SYSCFG is a register that enables high-speed operation, selects the host controller mode or function controller mode, controls the DP and DM pins, and enables operation of this module.
This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	HSE	DCFM	DRPD	DPRPU	—	UCK SEL	UPLLE	USBE
Initial value:	—	—	—	—	—	—	—	0	0	0	1	0	—	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
8	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
7	HSE	0	R/W	High-Speed Operation Enable 0: High-speed operation is disabled When the function controller mode is selected: Only full-speed operation is enabled. When the host controller mode is selected: Full-speed or low-speed operation is enabled. 1: High-speed operation is enabled (communication speed is detected by this module) (1) When the host controller mode is selected When HSE = 0, the USB port performs low-speed or full-speed operation. Set HSE to 0 when connection of a low-speed peripheral device to the USB port has been detected. When HSE = 1, this module executes the reset handshake protocol, and automatically allows the USB port to perform high-speed or full-speed operation according to the protocol execution result. This bit should be modified after detecting device connection (after detecting the ATTCH interrupt) and before executing a USB bus reset (before setting USBRST to 1). (2) When the function controller mode is selected When HSE = 0, this module performs full-speed operation. When HSE = 1, this module executes the reset handshake protocol, and automatically performs high-speed or full-speed operation according to the protocol execution result. This bit should be modified while DPRPU is 0.
6	DCFM	0	R/W	Controller Mode Select Selects the host controller mode or function controller mode. 0: Function controller mode is selected. 1: Host controller mode is selected. This bit should be modified while DPRPU and DRPD are 0.
5	DRPD	1	R/W	D+/D– Line Resistor Control Enables or disables pulling down D+ and D- lines when the host controller mode is selected. 0: Pulling down the lines is disabled. 1: Pulling down the lines is enabled. This bit should be set to 1 if the host controller mode is selected, and should be set to 0 if the function controller mode is selected.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
4	DPRPU	0	R/W	D+ Line Resistor Control Enables or disables pulling up D+ line when the function controller mode is selected. 0: Pulling up the line is disabled. 1: Pulling up the line is enabled. Setting this bit to 1 when the function controller mode is selected allows this module to pull up the D+ line, thus notifying the USB host of connection. Modifying this bit from 1 to 0 allows this module to cancel pulling up the D+ line, thus notifying the USB host of disconnection. This bit should be set to 1 if the function controller mode is selected, and should be set to 0 if the host controller mode is selected.
3	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
2	UCKSEL*1	0	R/W	Input Clock Selection Selects the clock to be supplied to this module from 48-MHz USB_X1 or 12-MHz EXTAL. 0: The 48-MHz USB_X1 clock is selected. 1: The 12-MHz EXTAL clock is selected. This bit should be modified while SUSPMODE.SUSPM for channel 0 and channel 1 is 0. Note that this bit is available only in channel 0 (SYSCFG0_0). Make the required setting for SYSCFG0_0 to use channel 1.
1	UPLLE*1	0	R/W	USB Internal PLL Operation Enable Enables or disables operation of the internal PLL of the USB block. 0: Disables operation of the internal PLL. 1: Enables operation of the internal PLL. This bit should be modified while SUSPMODE.SUSPM for channel 0 and channel 1 is 0. Set this bit to 0 when this module is to enter software standby or USB standby mode. Note that this bit is available only in channel 0 (SYSCFG0_0). Make the required setting for SYSCFG0_0 to use channel 1.
0	USBE	0	R/W	USB Module Operation Enable Enables or disables operation of this module. 0: USB module operation is disabled. 1: USB module operation is enabled. Modifying this bit from 1 to 0 initializes some register bits as listed in Table 28.3 and Table 28.4. This bit should be modified while SUSPMODE.SUSPM is 1. When the host controller mode is selected, this bit should be set to 1 after setting DPRD to 1, eliminating LNST bit chattering, and checking that the USB bus has been settled.

Note: • Writing to this register is possible even while the clock supply to this module is stopped (SUSPM = 0).
Note 1. UCKSEL and UPLLE are available only in channel 0 (SYSCFG0_0).

Table 28.3 Register Bits Initialized by Writing USBE = 0 (when Function Controller Mode is Selected)

Register Name	Bit Name	Remarks
SYSSTS0	LNST	The value is retained when the host controller mode is selected.
DVSTCTR0	RHST	
INTSTS0	DVSQ	The value is retained when the host controller mode is selected.
USBADDR	USBADDR	The value is retained when the host controller mode is selected.
USEREQ	BRequest, bmRequestType	The values are retained when the host controller mode is selected.
USBVAL	wValue	The value is retained when the host controller mode is selected.
USBINDX	wIndex	The value is retained when the host controller mode is selected.
USBLENG	wLength	The value is retained when the host controller mode is selected.

chanhvinh.com

Table 28.4 Register Bits Initialized by Writing USBE = 0 (when Host Controller Mode is Selected)

Register Name	Bit Name	Remarks
DVSTCTR	RHST	
FRMNUM	FRNM	The value is retained when the function controller mode is selected.
UFRMNUM	UFRNM	The value is retained when the function controller mode is selected.

28.3.2 CPU Bus Wait Setting Register (BUSWAIT)

BUSWAIT is a register that specifies the number of wait cycles to be inserted during an access from the CPU to this module.
This register can be modified even when the SUSPM bit in SUSPMODE is 0.
This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	BWAIT[5:0]					
Initial value:	—	—	0	0	1	1	1	1	—	—	0	0	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
13, 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 8	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 0.
7, 6	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
5 to 0	BWAIT[5:0]	001111	R/W	CPU Bus Wait Specifies the number of wait cycles to be inserted during an access to this module. 000000: 0 wait cycles (2 access cycles) : 000010: 2 wait cycles (4 access cycles) : 000100: 4 wait cycles (6 access cycles) : 001111: 15 wait cycles (17 access cycles) (initial value) : 111111: 63 wait cycles (65 access cycles) There is the following constraint imposed on the cycle period required to access SYSSTS0 and the subsequent registers of this module: Wait constraint: The cycle period required to consecutively access registers of this controller must be at least 67 ns. To satisfy this constraint, it is necessary to exercise wait control according to the frequency of peripheral clock 1 (P1ϕ). The initial value is 17 clock cycles. Select an appropriate value. This setting is also applied to the accesses to the FIFO port registers. The maximum access speeds for the FIFO ports are as follows: MBW = 10 (32-bit width): up to 60 MBytes/sec MBW = 01 (16-bit width): up to 30 MBytes/sec MBW = 00 (8-bit width): up to 15 MBytes/sec

chanhvinh.com

28.3.3 System Configuration Status Register (SYSSTS0)

SYSSTS is a register that monitors the line status (D+ and D– lines) of the USB data bus.
This register is initialized by a power-on reset or a USB bus reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	HTACT	SOFEA	—	—	—	LNST[1:0]	
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
6	HTACT	Undefined	R	USB Host Sequencer Status Monitor Indicates the status of the host sequencer. 0: Stopped 1: Running This bit is set to 0 when the host sequencer in this controller is completely stopped. Make sure that the HTACT bit is set to 0 to put this controller in the USB suspended state by setting the UACT bit to 0 and stop the clock by setting the SUSPM bit to 0 in the host-mode communication state.
5	SOFEA	Undefined	R	SOF Active Monitor when the Host Controller Mode Is Selected Indicates the SOF output status. 0: SOF output stopped 1: SOF output in progress This bit allows checking if the last SOF has been output after setting the UACT bit to 0 when putting this controller in the USB suspended state with the host controller mode selected. To set the SUSPM bit to 0 (to stop the clock) after stopping this controller by setting the USBE bit to 0 in the host-mode communication state, make sure that both the HTACT bit and the SOFEA bit are set to 0.
4 to 2	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
1, 0	LNST[1:0]	Undefined	R	USB Data Line Status Monitor Indicates the status of the USB data bus lines (D+ and D-) as shown in Table 28.5. These bits should be read after setting DPRPU to 1 to notify connection when the function controller mode is selected; whereas after setting DRPD to 1 to enable pulling down the lines when the host controller mode is selected.

Table 28.5 USB Data Bus Line States

LNST[1]	LNST[0]	During Low-Speed Operation (Only when Host Controller Mode is Selected)	During Full-Speed Operation	During High-Speed Operation	During Chirp Operation
0	0	SE0	SE0	Squelch	Squelch
0	1	K state	J state	UnSquelch	Chirp J
1	0	J state	K state	Invalid	Chirp K
1	1	SE1	SE1	Invalid	Invalid

[Legend]
Chirp: The reset handshake protocol (RHSP) is being executed in high-speed operation enabled state (the HSE bit in SYSCFG is set to 1).
Squelch: SE0 or idle state
UnSquelch: High-speed J state or high-speed K state
Chirp J: Chirp J state
Chirp K: Chirp K state

chanhvinh.com

28.3.4 Device State Control Register 0 (DVSTCTR0)

DVSTCTR is a register that controls and confirms the state of the USB data bus.
This register is initialized by a power-on reset. After a USB bus reset, only the WKUP bit is initialized.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	WKUP	RWUPE	USBRST	RESUME	UACT	—	RHST[2:0]		
Initial value:	—	—	—	—	0	0	0	0	0	0	0	0	—	0	0	0
R/W:	R	R	R	R	R	R	R	R/W*1	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	WKUP	0	R/W*1	Wakeup Output Enables or disables outputting the remote wakeup signal (resume signal) to the USB bus when the function controller mode is selected. 0: Remote wakeup signal is not output. 1: Remote wakeup signal is output. This module controls the output time of a remote wakeup signal. When this bit is set to 1, this module clears this bit to 0 after outputting the 10-ms K state. According to the USB Specification, the USB bus idle state must be kept for 5 ms or longer before a remote wakeup signal is output. If this module writes 1 to this bit right after detection of suspended state, the K state will be output after 2 ms. Do not write 1 to this bit unless the device state is in the suspended state (the DVSQ bits in the INTSTS0 register are set to 1xx) and the USB host enables the remote wakeup signal. When this bit is set to 1, the internal clock must not be stopped even in the suspended state (write 1 to this bit while SUSPMODE.SUSPM is 1). This bit should be set to 0 if the host controller mode is selected.
7	RWUPE	0	R/W	Remote Wakeup Detection Enable Enables or disables the downstream port peripheral device to use the remote wakeup function (resume signal output) when the host controller mode is selected. 0: Downstream port remote wakeup is disabled. 1: Downstream port remote wakeup is enabled. With this bit set to 1, on detecting the resume signal (K-state for 2.5 µs) to the downstream port when the remote wakeup signal has been detected, this module performs the resume process (drives the port to the K-state). With this bit set to 0, this module ignores the detected remote wakeup signal (K-state) from the peripheral device connected to the downstream port. While this bit is 1, the internal clock should not be stopped even in the suspended state (SUSPMODE.SUSPM should be set to 1). Also note that the USB bus should not be reset from the suspended state (USBRST should not be set to 1); it is prohibited by USB Specification 2.0. This bit should be set to 0 if the function controller mode is selected.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
6	USBRST	0	R/W	Bus Reset Output Controls the USB bus reset signal output when the host controller mode is selected. 0: USB bus reset signal is not output. 1: USB bus reset signal is output. When the host controller mode is selected, setting this bit to 1 allows this module to drive the USB port to SE0 to reset the USB bus. Here, this module performs the reset handshake protocol if the HSE bit is 1. This module continues outputting SE0 while USBRST is 1 (until 0 is written to USBRST). Ensure the period over which USBRST being set to 1 (= USB bus reset period) conforms to the USB Specification 2.0. Even if 1 is written to this bit during communication (UACT = 1) or during the resume process (RESUME = 1), this module does not start the USB bus reset process until both UACT and RESUME become 0. Write 1 to the UACT bit simultaneously with the end of the USB bus reset process (writing 0 to USBRST). This bit should be set to 0 if the function controller mode is selected.
5	RESUME	0	R/W	Resume Output Controls the resume signal output when the host controller mode is selected. 0: Resume signal is not output. 1: Resume signal is output. Setting this bit to 1 allows this module to drive the port to the K-state and output the resume signal. The controller sets this bit to 1 when detecting a remote wakeup signal while the RWUPE bit is set to 1 and the controller is in the USB suspended state. This module continues outputting K-state while RESUME is 1 (until 0 is written to RESUME). Ensure the period over which RESUME being set to 1 (= resume period) conforms to the USB Specification 2.0. This bit should only be set to 1 in the suspended state. Write 1 to the UACT bit simultaneously with the end of the resume process (writing 0 to RESUME). This bit should be set to 0 if the function controller mode is selected.
4	UACT	0	R/W	USB Bus Enable Enables operation of the USB bus (controls the SOF or µSOF packet transmission to the USB bus) when the host controller mode is selected. 0: Downstream port is disabled (SOF/µSOF transmission is disabled). 1: Downstream port is enabled (SOF/µSOF transmission is enabled). With this bit set to 1, this module puts the USB port to the USB-bus enabled state and performs SOF output and data transmission and reception. This module starts outputting SOF/µSOF within 1 (µ) frame after 1 is written to UACT. With this bit set to 0, this module enters the idle state after outputting SOF/µSOF. This module sets this bit to 0 on any of the following conditions. • A DTCH interrupt is detected during communication (while UACT = 1). • An EOFERR interrupt is detected during communication (while UACT = 1). Writing 1 to this bit should be done at the end of the USB bus reset process (writing 0 to USBRST) or at the end of the resume process from the suspended state (writing 0 to RESUME). This bit should be set to 0 if the function controller mode is selected.
3	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	RHST[2:0]	000	R	Reset Handshake Indicates the status of the reset handshake. (1) When the host controller mode is selected 000: Communication speed not determined (powered state or no connection) 1xx: Reset handshake in progress 001: Low-speed connection 010: Full-speed connection 011: High-speed connection These bits indicate 100 after 1 is written to USBRST. If HSE has been set to 1, these bits indicate 111 as soon as this module detects Chirp-K from the peripheral device. This module fixes the value of the RHST bits when 0 is written to USBRST and this module completes SE0 driving. When 1xxx is written to UTST (parameters for the host test have been set), these bits indicate 011. (2) When the function controller mode is selected 000: Communication speed not determined (powered state or no connection) 100: Reset handshake in progress 010: Full-speed connection 011: High-speed connection If HSE has been set to 1, these bits indicate 100 as soon as this module detects the USB bus reset. Then, these bits indicate 011 as soon as this module outputs Chirp-K and detects Chirp-JK from the USB host three times. If the connection speed is not fixed to high speed within 2.5 ms after Chirp-K output, these bits indicate 010. If HSE has been set to 0, these bits indicate 010 as soon as this module detects the USB bus reset. A DVST interrupt is generated as soon as this module detects the USB bus reset and then the value of the RHST bits is fixed to 010 or 011.

Note 1. Only 1 can be written.

chanhvinh.com

28.3.5 Test Mode Register (TESTMODE)

TESTMODE is a register that controls the USB test signal output during high-speed operation.
This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	UTST[3:0]			
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
3 to 0	UTST[3:0]	0000	R/W	Test Mode This module outputs the USB test signals during the high-speed operation, when these bits are written appropriate value. (1) When the host controller mode is selected These bits can be set after writing 1 to DRPD. This module outputs waveforms when both DPRD and UACT have been set to 1. This module also performs high-speed termination for the USB port by writing the appropriate value to these bits. • Procedure for setting the UTST bits 1. Power-on reset. 2. Start the clock supply. 3. Set SUSPM to 1. 4. Set DCFM and DPRD to 1 (setting HSE to 1 is not required). 5. Set USBE to 1. 6. Set the UTST bits to the appropriate value according to the test specifications. 7. Set the UACT bit to 1. • Procedure for modifying the UTST bits 1. (In the state after executing step 7 above) Set UACT and USBE to 0. 2. Set USBE to 1. 3. Set the UTST bits to the appropriate value according to the test specifications. 4. Set the UACT bit to 1. When these bits are set to Test_SE0_NAK (1011), this module does not output the SOF packet even when 1 has been set to UACT for the port. When these bits are set to Test_Force_Enable (1101), this module outputs the SOF packet when 1 has been set to UACT. In this test mode, this module does not perform hardware control consequent to detection of high-speed disconnection (detection of the DTCH interrupt). When setting the UTST bits, the PID bits for all the pipes should be set to NAK. To return to normal USB communication after a test mode has been set, a power-on reset should be applied. (2) When the function controller mode is selected The appropriate value should be set to these bits according to the SetFeature request from the USB host during high-speed communication. This module does not make a transition to the suspended state while these bits are 0001 to 0100. Perform a power-on reset to carry out normal USB communication after configuring the test mode.

chanhvinh.com

Table 28.6 Test Mode Operation

	UTST Bit Setting	
Test Mode	**When Function Controller Mode is Selected**	**When Host Controller Mode is Selected**
Normal operation	0000	0000
Test_J	0001	1001
Test_K	0010	1010
Test_SE0_NAK	0011	1011
Test_Packet	0100	1100
Test_Force_Enable	—	1101
Reserved	0101 to 0111	1110 to 1111

28.3.6 DMAn-FIFO Bus Configuration Registers (D0FBCFG, D1FBCFG)

D0FBCFG is a register that controls bus access to the DMA0-FIFO and D1FBCFG is a register that controls bus access to the DMA1-FIFO. Note that the setting of this register is invalid when the DMA0-FIFO bus or DMA1-FIFO bus is connected to the local bus.

These registers are initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	DFACC[1:0]		—	—	—	—	—	—	—	TENDE	—	—	—	—
Initial value:	—	—	0	0	—	—	—	—	—	—	—	0	—	—	—	—
R/W:	R	R	R/W	R/W	R	R	R	R	R	R	R	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
13, 12	DFACC[1:0]	00	R/W	DMAx FIFO Access Mode Specifies the access mode of the FIFO port. 00: Cycle steal mode (initial value) 01: 16-byte continuous access mode 10: 32-byte continuous access mode 11: Setting prohibited These bits specify a DMA transfer mode. (a) In cycle steal mode, use the DnFIFO port to access the FIFO buffer. (b) In 16-byte/32-byte continuous access mode, use the DnFIFO continuous transfer port to access the FIFO buffer. The MBW bit in DnFIFOSEL can be set to 10 (32-bit width) only. Be sure to follow the procedure below when setting the DFACC bits for 16- or 32-byte continuous access. (1) Set the DREQE bit in the DnFIFOSEL register to 0. (2) Set the DFACC bits to 01 (16 bytes) or 10 (32 bytes). (3) In the DnFIFOSEL register, set the CURPIPE bits to 0000 (specifying no pipe) and the MBW bits to 10 (32-bit width) at the same time. After that, read the CURPIPE bits to confirm that they have been updated to the written value (0000). (4) Use the CPU to dummy-read the DnFIFO port register (DnFIFO) with 32-bit width (the value read data can be discarded). After that, specify the target pipe by using the CURPIPE bits in the DnFIFOSEL register and read the CURPIPE bits to confirm that they have been updated to written value. (5) Set the DREQE bit in the DnFIFOSEL to 1. Note that the above procedure is not required if these bits are to be set for the cycle-stealing mode (not following the procedure has no effect).

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
11 to 5	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
4	TENDE	0	R/W	DMA Transfer End Sampling Enable Controls acceptance of DMA transfer end signal. 0: DMA transfer end signal is not sampled. 1: DMA transfer end signal is sampled. Controls acceptance of DMA transfer end signal output from the direct memory access controller on completion of a DMA transfer. This module can control input of the DMA transfer end signal to end the DMA transfer of data to the FIFO. Set this bit to 0 when the DFACC bits are set to 01 or 10.
3 to 0	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.

chanhvinh.com

28.3.7 FIFO Port Registers (CFIFO, D0FIFO, D1FIFO)

CFIFO, D0FIFO, and D1FIFO are port registers that are used to read data from the FIFO buffer memory and write data to the FIFO buffer memory.

There are three FIFO ports: the CFIFO, D0FIFO and D1FIFO ports. There are also DnFIFO continuous transfer ports for continuous transfer. Each FIFO port is configured of a port register (CFIFO, D0FIFO, or D1FIFO) that handles reading of data from the FIFO buffer memory and writing of data to the FIFO buffer memory, a select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) that is used to select the pipe assigned to the FIFO port, and a control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR).

Each FIFO port has the following characteristics.

- Access to the FIFO buffer for the DCP should be performed through the CFIFO port.
- Access to the FIFO buffer by DMA transfer should be performed through the D0FIFO or D1FIFO port when the DFACC bits are set to 00 (cycle steal mode).
- Access to the FIFO buffer by DMA transfer should be performed through D0FIFO or D1FIFO continuous transfer ports when the DFACC bits are set to 01 (16-byte continuous access mode) or 10 (32-byte continuous access mode).
- The D1FIFO or D0FIFO port can be accessed by the CPU.
- When using functions specific to the FIFO port, the pipe number (selected pipe) specified by the CURPIPE bits cannot be changed (when the DMA transfer function is used, etc.).
- Registers configuring a FIFO port do not affect other FIFO ports.
- The same pipe should not be assigned to two or more FIFO ports.
- There are two FIFO buffer states: the access right is on the CPU side and it is on the SIE side. When the FIFO buffer access right is on the SIE side, the FIFO buffer cannot be accessed from the CPU.

These registers are initialized by a power-on reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FIFOPORT[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FIFOPORT[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	FIFOPORT [31:0]	All 0	R/W	FIFO Port Accessing these bits allow reading the received data from the FIFO buffer or writing the transmit data to the FIFO buffer. These bits can be accessed only while the FRDY bit in each FIFO port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) is 1 (or this module is issuing a DMA transfer request). The valid bits in this register depend on the settings of the MBW bits (access bit width setting) and BIGEND bit (endian setting) as shown in Table 28.7 to Table 28.9.

chanhvinh.com

Table 28.7 Endian Operation in 32-Bit Access

BIGEND Bit	Bits 31 to 24	Bits 23 to 16	Bits 15 to 8	Bits 7 to 0
0	N+3 address	N+2 address	N+1 address	N+0 address
1	N+0 address	N+1 address	N+2 address	N+3 address

Table 28.8 Endian Operation in 16-Bit Access

BIGEND Bit	Bits 31 to 24	Bits 23 to 16	Bits 15 to 8	Bits 7 to 0
0	N+1 address	N+0 address	Writing: invalid, reading: prohibited*1	
1	Writing: invalid, reading: prohibited*1		N+0 address	N+1 address

Note 1. Reading data from the invalid bits in a word or byte unit is prohibited.

Table 28.9 Endian Operation in 8-Bit Access

BIGEND Bit	Bits 31 to 24	Bits 23 to 16	Bits 15 to 8	Bits 7 to 0
0	N+0 address	Writing: invalid, reading: prohibited*1		
1	Writing: invalid, reading: prohibited*1			N+0 address

Note 1. Reading data from the invalid bits in a word or byte unit is prohibited.

chanhvinh.com

28.3.8 FIFO Port Select Registers (CFIFOSEL, D0FIFOSEL, D1FIFOSEL)

CFIFOSEL, D0FIFOSEL and D1FIFOSEL are registers that assign the pipe to the FIFO port, and control access to the corresponding port.

The same pipe should not be specified by the CURPIPE bits in CFIFOSEL, D0FIFOSEL and D1FIFOSEL. When the CURPIPE bits in D0FIFOSEL and D1FIFOSEL are cleared to B'0000, no pipe is selected.

The pipe number should not be changed while the DMA transfer is enabled.

These registers are initialized by a power-on reset.

(1) CFIFOSEL

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RCNT	REW	—	—	MBW[1:0]		—	BIG END	—	—	ISEL	—	CURPIPE[3:0]			
Initial value:	0	0	—	—	0	0	—	0	—	—	0	—	0	0	0	0
R/W:	R/W	R/W*1	R	R	R/W	R/W	R	R/W	R	R	R/W	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	RCNT	0	R/W	Read Count Mode Specifies the read mode for the value in the DTLN bits in CFIFOCTR. 0: The DTLN bits are cleared when all of the receive data has been read from the CFIFO. (In double buffer mode, the DTLN bit value is cleared when all the data has been read from a single plane.) 1: The DTLN bits are decremented when the receive data is read from the CFIFO. When this bit is cleared to 0, this module clears the DTLN bits in CFIFOCTR to 0 when all of the receive data has been read from the FIFO buffer that is assigned to the pipe specified in the CURPIPE bits (called the specified pipe) (in double buffer mode, the timing to clear is when finished reading data from one FIFO buffer plane). When this bit is set to 1, this module decrements the DTLN bits in CFIFOCTR every time receive data is read from the FIFO buffer that is assigned to the specified pipe.
14	REW	0	R/W*1	Buffer Pointer Rewind Specifies whether or not to rewind the buffer pointer. 0: The buffer pointer is not rewound. 1: The buffer pointer is rewound. When the selected pipe is in the receiving direction, setting this bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read one FIFO buffer plane from the first data is allowed). Do not set REW to 1 simultaneously with modifying the CURPIPE bits. Before setting REW to 1, be sure to check that FRDY is 1. To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.
13, 12	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
11, 10	MBW[1:0]	00	R/W	FIFO Port Access Bit Width Specifies the bit width for accessing the CFIFO port. 00: 8-bit width 01: 16-bit width 10: 32-bit width 11: Setting prohibited When the pipe specified by the CURPIPE bits is in the receiving direction, once reading data is started after setting these bits, these bits should not be modified until all the data has been read. When the specified pipe is in the receiving direction, set the CURPIPE bits to a different value once, and then set the CURPIPE and MBW bits simultaneously. Regarding the procedure for changing the value of the CURPIPE bits, do so in accord with the description of the CURPIPE bits. When the specified pipe is in the transmitting direction, the bit width cannot be changed from the 8-bit width to the 16-/32-bit width or from the 16-bit width to the 32-bit width while data is being written to the buffer memory.
9	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
8	BIGEND	0	R/W	FIFO Port Endian Control Specifies the byte endian for the CFIFO port. For details, refer to the description of the FIFO port bits in section 28.3.7, FIFO Port Registers (CFIFO, D0FIFO, D1FIFO). 0: Little endian 1: Big endian
7, 6	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
5	ISEL	0	R/W	FIFO Port Access Direction when DCP is Selected Specifies the direction of FIFO port access when the DCP is selected through the CURPIPE bits. 0: Reading from the buffer memory is selected 1: Writing to the buffer memory is selected After writing to this bit with the DCP being a selected pipe, read this bit to check that the written value agrees with the read value before proceeding to the next process. When this bit is modified during access to the FIFO buffer, the access results up to that point can be retained, and after the bit is restored to the previous value, access can be continued. Set this bit and the CURPIPE bits simultaneously.
4	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
3 to 0	CURPIPE[3:0]	0000	R/W	FIFO Port Access Pipe Specification Specifies the pipe number using which data is read or written through the CFIFO port. 0000: DCP 0001: PIPE 1 0010: PIPE 2 : 1110: PIPE 14 1111: PIPE 15 After writing to these bits, read these bits to check that the written value agrees with the read value before proceeding to the next process. Do not set the same pipe number to the CURPIPE bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL. Even if the setting of these bits is modified during access to the FIFO buffer, the state of the FIFO buffer is retained, with continued access proceeding after these bits are re-set to the value before the modification.

Note 1. Only 0 can be read and 1 can be written to.

chanhvinh.com

(2) D0FIFOSEL, D1FIFOSEL

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RCNT	REW	DCLRM	DREQE	MBW[1:0]		—	BIGEND	—	—	—	—	CURPIPE[3:0]			
Initial value:	0	0	0	0	0	0	—	0	—	—	—	—	0	0	0	0
R/W:	R/W	R/W*	R/W	R/W	R/W	R/W	R	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	RCNT	0	R/W	Read Count Mode Specifies the read mode for the value in the DTLN bits in DnFIFOCTR. 0: The DTLN bits are cleared when all of the receive data has been read from the DnFIFO. (In double buffer mode, the DTLN bit value is cleared when all the data has been read from a single plane.) 1: The DTLN bits are decremented when the receive data is read from the DnFIFO. When this bit is cleared to 0, this module clears the DTLN bits in DnFIFOCTR to 0 when all of the receive data has been read from the FIFO buffer that is assigned to the pipe specified in the CURPIPE bits (called the specified pipe) (in double buffer mode, the timing to clear is when finished reading data from one FIFO buffer plane). When this bit is set to 1, this module decrements the DTLN bits in DnFIFOCTR every time receive data is read from the FIFO buffer that is assigned to the specified pipe.
14	REW	0	R/W*	Buffer Pointer Rewind Specifies whether or not to rewind the buffer pointer. 0: The buffer pointer is not rewound. 1: The buffer pointer is rewound. When the specified pipe is in the receiving direction, setting this bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed). Do not set REW to 1 simultaneously with modifying the CURPIPE bits. Before setting REW to 1, be sure to check that FRDY is 1. To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.
13	DCLRM	0	R/W	Auto Buffer Memory Clear Mode Accessed after Specified Pipe Data is Read Enables or disables the buffer memory to be cleared automatically after data has been read out using the selected pipe. 0: Auto buffer clear mode is disabled. 1: Auto buffer clear mode is enabled. With this bit set to 1, this module sets BCLR to 1 for the FIFO buffer of the selected pipe on receiving a zero-length packet while the FIFO buffer assigned to the selected pipe is empty, or on receiving a short packet and reading the data while BFRE is 1. When using this module with the BRDYM bit set to 1, set this bit to 0.
12	DREQE	0	R/W	DMA Transfer Request Enable Enables or disables the issuance of a DMA transfer request. 0: DMA transfer request is disabled. 1: DMA transfer request is enabled. To enable the issuance of a DMA transfer request, set this bit to 1 after setting the CURPIPE bits. When modifying the CURPIPE bits, first set this bit to 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
11, 10	MBW[1:0]	00	R/W	FIFO Port Access Bit Width Specifies the bit width for accessing the D0FIFO or D1FIFO port. 00: 8-bit width 01: 16-bit width 10: 32-bit width 11: Setting prohibited Set 10 when the DFACC bits are set to 01 or 10. When the pipe specified by the CURPIPE bits is in the receiving direction, once reading data is started after setting these bits, these bits should not be modified until all the data has been read. When the specified pipe is in the receiving direction, set the CURPIPE bits to a different value once, and then set the CURPIPE and MBW bits simultaneously. Regarding the procedure for changing the value of the CURPIPE bits, do so in accord with the description of the CURPIPE bits. When the specified pipe is in the transmitting direction, the bit width cannot be changed from the 8-bit width to the 16-/32-bit width or from the 16-bit width to the 32-bit width while data is being written to the buffer memory. Set these bits to 10 (32-bit width) when the DFACC bits are set to 01 (16-byte continuous access mode) or 10 (32-byte continuous access mode).
9	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
8	BIGEND	0	R/W	FIFO Port Endian Control Specifies the byte endian for the D0FIFO or D1FIFO port. For details, refer to the description of the FIFO port bits in section 28.3.7, FIFO Port Registers (CFIFO, D0FIFO, D1FIFO). 0: Little endian 1: Big endian
7 to 4	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
3 to 0	CURPIPE[3:0]	0000	R/W	FIFO Port Access Pipe Specification Specify a desired pipe number for which data is read or written through the D0FIFO or D1FIFO port. 0000: No pipe specified 0001: PIPE 1 0010: PIPE 2 : 1110: PIPE 14 1111: PIPE 15 After writing to these bits, read these bits to check that the written value agrees with the read value before proceeding to the next process. Do not set the same pipe number to the CURPIPE bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL. Even if the setting of these bits is modified during access to the FIFO buffer, the state of the FIFO buffer is retained, with continued access proceeding after these bits are re-set to the value before the modification.

chanhvinh.com

28.3.9 FIFO Port Control Registers (CFIFOCTR, D0FIFOCTR, D1FIFOCTR)

CFIFOCTR, D0FIFOCTR and D1FIFOCTR are registers that determine whether or not writing to the buffer memory has been finished, the buffer accessed from the CPU has been cleared, and the FIFO port is accessible. CFIFOCTR, D0FIFOCTR, and D1FIFOCTR are used for the corresponding FIFO ports.
These registers are initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BVAL	BCLR	FRDY	—	DTLN[11:0]											
Initial value:	0	0	0	—	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*2	R/W*1	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	BVAL	0	R/W*2	Buffer Memory Valid Flag Set this bit to 1 when writing has completed in the CPU-side FIFO buffer for the pipe specified in CURPIPE (called the selected pipe). 0: Invalid 1: Writing ended When the selected pipe is in the transmitting direction, set this bit to 1 in the following cases. Then, this module switches the FIFO buffer from the CPU side to the SIE side, enabling transmission. • To transmit a short packet, set this bit to 1 after data has been written. • To transmit a zero-length packet, set this bit to 1 before writing data to the FIFO buffer. • Set this bit to 1 after the number of data bytes has been written for the pipe in continuous transfer mode, where the number is a natural integral multiple of the maximum packet size and less than the buffer size. When the data of the maximum packet size has been written for the pipe in non-continuous transfer mode, this module sets this bit to 1 and switches the FIFO buffer from the CPU side to the SIE side, enabling transmission. Writing 1 to this bit should be done while FRDY indicates 1. When checking the FRDY bit after setting this bit, allow an interval of at least 80 ns before referencing FRDY. When the selected pipe is in the receiving direction, do not write 1 to this bit.
14	BCLR	0	R/W*1	CPU Buffer Clear This bit should be set to 1 to clear the FIFO buffer on the CPU side for the selected pipe. 0: Invalid 1: Clears the buffer memory on the CPU side. When double buffer mode is set for the FIFO buffer assigned to the selected pipe, this module clears only one plane of the FIFO buffer even when both planes are read-enabled. When the selected pipe is the DCP, setting BCLR to 1 allows this module to clear the FIFO buffer regardless of whether the FIFO buffer is on the CPU side or SIE side. When clearing the buffer on the SIE side, set the PID bits for the DCP to NAK before setting BCLR to 1. When the selected pipe is not the DCP, writing 1 to this bit should be done while FRDY indicates 1. When checking the FRDY bit after setting this bit, allow an interval of at least 80 ns before referencing FRDY.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
13	FRDY	0	R	FIFO Port Ready Indicates whether the FIFO port can be accessed. 0: FIFO port access is disabled. 1: FIFO port access is enabled. In the following cases, this module sets FRDY to 1 but data cannot be read via the FIFO port because there is no data to be read. In these cases, set BCLR to 1 to clear the FIFO buffer, and enable transmission and reception of the next data. • A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty. • A short packet is received and the data is completely read while BFRE is 1.
12	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
11 to 0	DTLN[11:0]	H'000	R	Receive Data Length Indicates the length of the receive data. While the FIFO buffer is being read, these bits indicate the different values depending on the RCNT bit value as described below. • When RCNT = 0: The length of received data is set in these bits, and the value is retained until all received data has been read from a single FIFO buffer plane. While BFRE is 1, these bits retain the length of the receive data until BCLR is set to 1 even after all the data has been read. • When RCNT = 1: This module decrements the value indicated by these bits each time data is read from the FIFO buffer. (The value is decremented by one when MBW is 00, by two when MBW is 01, and by four when MBW is 10.) This module sets these bits to 0 when all the data has been read from one FIFO buffer plane. However, in double buffer mode, if data has been received in one FIFO buffer plane before all the data has been read from the other plane, this module sets these bits to indicate the length of the receive data in the latter plane when all the data has been read from the former plane. Note: When reading these bits during FIFO buffer reading while RCNT = 1, note that these bits are updated within ten bus cycles after a read cycle for the FIFO port.

Note 1. Only 0 can be read and 1 can be written to.
Note 2. Only 1 can be written to.

chanhvinh.com

28.3.10 Interrupt Enable Register 0 (INTENB0)

INTENB0 is a register that enables various interrupts. On detecting the interrupt corresponding to the bit in this register which has been set to 1, this module generates the USB interrupt.
This module sets 1 to each status bit in INTSTS0 when a detection condition of the corresponding interrupt source has been satisfied regardless of the set value in INTENB0 (regardless of whether the interrupt output is enabled or disabled).
While the status bit in INTSTS0 corresponding to the interrupt source indicates 1, this module generates the USB interrupt when the corresponding interrupt enable bit in INTENB0 is changed from 0 to 1.
This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	VBSE	0	R/W	VBUS Interrupt Enable Enables or disables the USB interrupt request when the VBINT interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
14	RSME	0	R/W	Resume Interrupt Enable*1 Enables or disables the USB interrupt request when the RESM interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
13	SOFE	0	R/W	Frame Number Update Interrupt Enable Enables or disables the USB interrupt request when the SOFR interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
12	DVSE	0	R/W	Device State Transition Interrupt Enable*1 Enables or disables the USB interrupt request when the DVST interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
11	CTRE	0	R/W	Control Transfer Stage Transition Interrupt Enable*1 Enables or disables the USB interrupt request when the CTRT interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
10	BEMPE	0	R/W	Buffer Empty Interrupt Enable Enables or disables the USB interrupt request when the BEMP interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
9	NRDYE	0	R/W	Buffer Not Ready Response Interrupt Enable Enables or disables the USB interrupt request when the NRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
8	BRDYE	0	R/W	Buffer Ready Interrupt Enable Enables or disables the USB interrupt request when the BRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.

Note 1. The RSME, DVSE, and CTRE bits can be set to 1 only when the function controller mode is selected; do not set these bits to 1 to enable the corresponding interrupt output when the host controller mode is selected.

28.3.11 Interrupt Enable Register 1 (INTENB1)

INTENB1 is a register that enables various interrupts when the host controller mode is selected.
On detecting the interrupt corresponding to the bit in this register which has been set to 1, this module generates the USB interrupt.
This module sets 1 to each status bit in INTSTS1 when a detection condition of the corresponding interrupt source has been satisfied regardless of the set value in INTENB1 (regardless of whether the interrupt output is enabled or disabled).
While the status bit in INTSTS1 corresponding to the interrupt source indicates 1, this module generates the USB interrupt when the corresponding interrupt enable bit in INTENB1 is changed from 0 to 1.
When the function controller mode is selected, the interrupts should not be enabled.
This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	BCHGE	—	DTCHE	ATTCHE	—	—	—	—	EOFERRE	SIGNE	SACKE	—	—	—	—
Initial value:	0	0	—	0	0	—	0	0	—	0	0	0	—	—	—	0
R/W:	R	R/W	R	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	BCHGE	0	R/W	USB Bus Change Interrupt Enable Enables or disables the USB interrupt request when the BCHG interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
13	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
12	DTCHE	0	R/W	Disconnection Detection Interrupt Enable Enables or disables the USB interrupt request when the DTCH interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
11	ATTCHE	0	R/W	Connection Detection Interrupt Enable Enables or disables the USB interrupt request when the ATTCH interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
10	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
9, 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
6	EOFERRE	0	R/W	EOF Error Detection Interrupt Enable Enables or disables the USB interrupt request when the EOFERR interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
5	SIGNE	0	R/W	Setup Transaction Error Interrupt Enable Enables or disables the USB interrupt request when the SIGN interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
4	SACKE	0	R/W	Setup Transaction Normal Response Interrupt Enable Enables or disables the USB interrupt request when the SACK interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
3 to 1	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Note 1. The INTENB1 register bits can be set to 1 only when the host controller mode is selected; do not set these bits to 1 to enable the corresponding interrupt output when the function controller mode is selected.

28.3.12 BRDY Interrupt Enable Register (BRDYENB)

BRDYENB is a register that enables or disables the BRDY bit in INTSTS0 to be set to 1 when the BRDY interrupt is detected for each pipe.

On detecting the BRDY interrupt for the pipe corresponding to the bit in this register which has been set to 1, this module sets 1 to the corresponding PIPEBRDY bit in BRDYSTS and the BRDY bit in INTSTS0, and generates the BRDY interrupt.

While at least one PIPEBRDY bit in BRDYSTS indicates 1, this module generates the BRDY interrupt when the corresponding interrupt enable bit in BRDYENB is changed from 0 to 1.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIPEBRDYE															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	PIPEBRDYE	H'0000	R/W	BRDY Interrupt Enable for each Pipe 0: Interrupt output disabled 1: Interrupt output enabled

Note 1. The bit number corresponds to the pipe number.

chanhvinh.com

28.3.13 NRDY Interrupt Enable Register (NRDYENB)

NRDYENB is a register that enables or disables the NRDY bit in INTSTS0 to be set to 1 when the NRDY interrupt is detected for each pipe.

On detecting the NRDY interrupt for the pipe corresponding to the bit in this register which has been set to 1, this module sets 1 to the corresponding PIPENRDY bit in NRDYSTS and the NRDY bit in INTSTS0, and generates the NRDY interrupt.

While at least one PIPENRDY bit in NRDYSTS indicates 1, this module generates the NRDY interrupt when the corresponding interrupt enable bit in NRDYENB is changed from 0 to 1.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIPENRDYE															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	PIPENRDYE	H'0000	R/W	NRDY Interrupt Enable for each Pipe 0: Interrupt output disabled 1: Interrupt output enabled

Note 1. The bit number corresponds to the pipe number.

28.3.14 BEMP Interrupt Enable Register (BEMPENB)

BEMPENB is a register that enables or disables the BEMP bit in INTSTS0 to be set to 1 when the BEMP interrupt is detected for each pipe.

On detecting the BEMP interrupt for the pipe corresponding to the bit in this register which has been set to 1, this module sets 1 to the corresponding PIPEBEMP bit in BEMPSTS and the BEMP bit in INTSTS0, and generates the BEMP interrupt.

While at least one PIPEBEMP bit in BEMPSTS indicates 1, this module generates the BEMP interrupt when the corresponding interrupt enable bit in BEMPENB is changed from 0 to 1.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIPEBEMPE															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	PIPEBEMPE	H'0000	R/W	BEMP Interrupt Enable for each Pipe 0: Interrupt output disabled 1: Interrupt output enabled

Note 1. The bit number corresponds to the pipe number.

chanhvinh.com

28.3.15 SOF Output Configuration Register (SOFCFG)

SOFCFG is a register that specifies the transaction-enabled time and BRDY interrupt status clear timing.
This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TRNENSEL	—	BRDYM	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	0	—	0	0	0	0	0	—	—
R/W:	R	R	R	R	R	R	R	R/W	R	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
8	TRNENSEL	0	R/W	Transaction-Enabled Time Select Selects the transaction-enabled time either for full-speed or low-speed communication, where is the time in which this module issues tokens in a frame. 0: For non-low-speed communication 1: For low-speed communication This bit is valid only when the host controller mode is selected. Even when the host controller mode is selected, the setting of this bit has no effect on the transaction-enabled time during high-speed communication. This bit should be set to 0 when the function controller mode is selected.
7	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
6	BRDYM	0	R/W	BRDY Interrupt Status Clear Timing for Each Pipe Specifies the timing for clearing the BRDY interrupt status for each pipe. 0: Clears the status by writing 0 to this bit. 1: This module automatically clears the status when data has been read from the FIFO buffer or data has been written to the FIFO buffer.
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.

chanhvinh.com

28.3.16 Interrupt Status Register 0 (INTSTS0)

INTSTS0 is a register that indicates the status of the various interrupts detected.
This register is initialized by a power-on reset. By a USB bus reset, the DVST and DVSQ[2:0] bits are initialized.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ[2:0]			VALID	CTSQ[2:0]		
Initial value:	0	0	0	0/1*1	0	0	0	0	0/1*3	0*2	0*2	0/1*2	0	0	0	0
R/W:	R/W*7	R/W*7	R/W*7	R/W*7	R/W*7	R	R	R	R	R	R	R	R/W*7	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	VBINT	0	R/W*7	VBUS Interrupt Status*4*5 0: VBUS interrupts not generated 1: VBUS interrupts generated This module sets this bit to 1 when a change in the value input to the VBUS pin (high to low or low to high) is detected. The module indicates the VBUS pin input value in the VBSTS bit. When a VBINT interrupt is generated, the VBSTS bit is read several times to remove the chattering effect until the same value is read repeatedly from the bit.
14	RESM	0	R/W*7	Resume Interrupt Status*4*5*6 0: Resume interrupts not generated 1: Resume interrupts generated When the function controller mode is selected, this module sets this bit to 1 on detecting the falling edge of the signal on the DP pin in the suspended state (DVSQ = 1XX). When the host controller mode is selected, the read value is invalid.
13	SOFR	0	R/W*7	Frame Number Refresh Interrupt Status*4 0: SOF interrupts not generated 1: SOF interrupts generated (1) When the host controller mode is selected This module sets this bit to 1 on updating the frame number when the UACT bit is set to 1. (This interrupt is detected every 1 ms.) (2) When the function controller mode is selected This module sets this bit to 1 on updating the frame number. (This interrupt is detected every 1 ms.) This module can detect an SOFR interrupt through the internal interpolation function even when a damaged SOF packet is received from the USB host.
12	DVST	0/1*1	R/W*7	Device State Transition Interrupt Status*4*6 0: Device state transition interrupts not generated 1: Device state transition interrupts generated When the function controller mode is selected, this module updates the DVSQ value and sets this bit to 1 on detecting a change in the device state. When this interrupt is generated, clear the status before this module detects the next device state transition. When the host controller mode is selected, the read value is invalid.
11	CTRT	0	R/W*7	Control Transfer Stage Transition Interrupt Status*4*6 0: Control transfer stage transition interrupts not generated 1: Control transfer stage transition interrupts generated When the function controller mode is selected, this module updates the CTSQ value and sets this bit to 1 on detecting a change in the control transfer stage. When this interrupt is generated, clear the status before this module detects the next control transfer stage transition. When the host controller mode is selected, the read value is invalid.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
10	BEMP	0	R	Buffer Empty Interrupt Status 0: BEMP interrupts not generated 1: BEMP interrupts generated This module sets this bit to 1 when at least one PIPEBEMP bit in BEMPSTS is set to 1 among the PIPEBEMP bits corresponding to the PIPEBEMPE bits in BEMPENB to which 1 has been set (when this module detects the BEMP interrupt status in at least one pipe among the pipes for which the BEMP interrupt notification is enabled). For the conditions for PIPEBEMP status assertion, refer to section 28.4.2 (4) BEMP Interrupt. This module clears this bit to 0 when 0 is written to all the PIPEBEMP bits corresponding to the PIPEBEMPE bits to which 1 has been set. This bit cannot be cleared to 0 even if 0 is written to this bit.
9	NRDY	0	R	Buffer Not Ready Interrupt Status 0: NRDY interrupts not generated 1: NRDY interrupts generated This module sets this bit to 1 when at least one PIPENRDY bit in NRDYSTS is set to 1 among the PIPENRDY bits corresponding to the PIPENRDYE bits in NRDYENB to which 1 has been set (when this module detects the NRDY interrupt status in at least one pipe among the pipes for which the NRDY interrupt notification is enabled). For the conditions for PIPENRDY status assertion, refer to section 28.4.2 (3) NRDY Interrupt. This module clears this bit to 0 when 0 is written to all the PIPENRDY bits corresponding to the PIPENRDYE bits to which 1 has been set. This bit cannot be cleared to 0 even if 0 is written to this bit.
8	BRDY	0	R	Buffer Ready Interrupt Status 0: BRDY interrupts not generated 1: BRDY interrupts generated This module sets this bit to 1 when at least one PIPEBRDY bit in BRDYSTS is set to 1 among the PIPEBRDY bits corresponding to the PIPEBRDYE bits in BRDYENB to which 1 has been set (when this module detects the BRDY interrupt status in at least one pipe among the pipes for which the BRDY interrupt notification is enabled). For the conditions for PIPEBRDY status assertion, refer to section 28.4.2 (2) BRDY Interrupt. This module clears this bit to 0 when 0 is written to all the PIPEBRDY bits corresponding to the PIPEBRDYE bits to which 1 has been set. This bit cannot be cleared to 0 even if 0 is written to this bit.
7	VBSTS	0/1*3	R	VBUS Input Status 0: The VBUS pin is low level. 1: The VBUS pin is high level.
6 to 4	DVSQ[2:0]	000/001*2	R	Device State 000: Powered state 001: Default state 010: Address state 011: Configured state 1xx: Suspended state When the host controller mode is selected, the read value is invalid.
3	VALID	0	R/W*7	USB Request Reception 0: Not detected 1: Setup packet reception When the host controller mode is selected, the read value is invalid.
2 to 0	CTSQ[2:0]	000	R	Control Transfer Stage 000: Idle or setup stage 001: Control read data stage 010: Control read status stage 011: Control write data stage 100: Control write status stage 101: Control write (no data) status stage 110: Control transfer sequence error 111: Setting prohibited When the host controller mode is selected, the read value is invalid.

chanhvinh.com

Note 1. This bit is initialized to B'0 by a power-on reset and B'1 by a USB bus reset.
Note 2. These bits are initialized to B'000 by a power-on reset and B'001 by a USB bus reset.
Note 3. This bit is 1 when the level of the VBUS pin input is high and 0 when low.
Note 4. To clear the VBINT, RESM, SOFR, DVST, or CTRT bit, write 0 only to the bits to be cleared; write 1 to the other bits. Do not write 0 to the status bits indicating 0.
Note 5. A change in the status indicated by the VBINT and RESM bits can be detected by this module even while the clock supply is stopped, and the interrupts are output when the corresponding interrupt enable bits are enabled. Clearing the status should be done after enabling the clock supply.
Note 6. A change in the status of the RESM, DVST, and CTRT bits occurs only when the function controller mode is selected; disable the corresponding interrupt enable bits (set to 0) when the host controller mode is selected.
Note 7. Only 0 can be written to.

chanhvinh.com

28.3.17 Interrupt Status Register 1 (INTSTS1)

INTSTS1 is a register that is used to confirm interrupt status.

The various interrupts indicated by the bits in this register should be enabled only when the host controller mode is selected.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	BCHG	—	DTCH	ATTCH	—	—	—	—	EOFERR	SIGN	SACK	—	—	—	—
Initial value:	0	0	—	0	0	—	0	0	—	0	0	0	—	—	—	0
R/W:	R	R/W*	R	R/W*	R/W*	R	R	R	R	R/W*	R/W*	R/W*	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	BCHG	0	R/W*	USB Bus Change Interrupt Status Indicates the status of the USB bus change interrupt. 0: BCHG interrupts not generated 1: BCHG interrupts generated This module detects the BCHG interrupt when a change in the full-speed or low-speed signal level occurs on the USB port (a change from J-state, K-state, or SE0 to J-state, K-state, or SE0), and sets this bit to 1. Here, if the corresponding interrupt enable bit is set to 1, this module generates the interrupt. This module sets the LNST bits in SYSSTS to indicate the current input state of the USB port. When the BCHG interrupt is generated, the LNST bits are read several times to remove the chattering effect until the same value is read repeatedly from the bits. A change in the USB bus state can be detected even while the internal clock supply is stopped. When the function controller mode is selected, the read value is invalid.
13	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
12	DTCH	0	R/W*	USB Disconnection Detection Interrupt Status Indicates the status of the USB disconnection detection interrupt when the host controller mode is selected. 0: DTCH interrupts not generated 1: DTCH interrupts generated This module detects the DTCH interrupt on detecting USB bus disconnection, and sets this bit to 1. Here, if the corresponding interrupt enable bit is set to 1, this module generates the interrupt. This module detects bus disconnection based on USB Specification 2.0. After detecting the DTCH interrupt, this module controls hardware as described below (irrespective of the set value of the corresponding interrupt enable bit). Software should terminate all the pipes in which communications are currently carried out for the USB port and make a transition to the wait state for bus connection to the USB port (wait state for ATTCH interrupt generation). (1) Modifies the UACT bit to 0. (2) Puts the port into the idle state. When the function controller mode is selected, the read value is invalid.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
11	ATTCH	0	R/W*	ATTCH Interrupt Status Indicates the status of the ATTCH interrupt when the host controller mode is selected. 0: ATTCH interrupts not generated 1: ATTCH interrupts generated This module detects the ATTCH interrupt on detecting J-state or K-state of the full-speed or low-speed level signal for 2.5 µs, and sets this bit to 1. Here, if the corresponding interrupt enable bit is set to 1, this module generates the interrupt. Specifically, this module detects the ATTCH interrupt on any of the following conditions. • K-state, SE0, or SE1 changes to J-state, and J-state continues 2.5 µs. • J-state, SE0, or SE1 changes to K-state, and K-state continues 2.5 µs. When the function controller mode is selected, the read value is invalid.
10	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
9, 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
6	EOFERR	0	R/W*	EOF Error Detection Interrupt Status Indicates the status of the EOFERR interrupt when the host controller mode is selected. 0: EOFERR interrupt not generated 1: EOFERR interrupt generated This module detects the EOFERR interrupt on detecting that communication is not completed at the EOF2 timing prescribed by USB Specification 2.0, and sets this bit to 1. Here, if the corresponding interrupt enable bit is set to 1, this module generates the EOFERR interrupt. After detecting the EOFERR interrupt, this module controls hardware as described below (irrespective of the set value of the corresponding interrupt enable bit). Software should terminate all the pipes in which communications are currently carried for the USB port and perform re-enumeration of the USB port. (1) Modifies the UACT bit to 0. (2) Puts the port into the idle state. When the function controller mode is selected, the read value is invalid.
5	SIGN	0	R/W*	Setup Transaction Error Interrupt Status Indicates the status of the setup transaction error interrupt when the host controller mode is selected. 0: SIGN interrupts not generated 1: SIGN interrupts generated This module detects the SIGN interrupt when ACK response is not returned from the peripheral device three consecutive times during the setup transactions issued by this module, and sets this bit to 1. Here, if the corresponding interrupt enable bit is set to 1, this module generates the SIGN interrupt. Specifically, this module detects the SIGN interrupt when any of the following response conditions occur for three consecutive setup transactions. • Timeout is detected when the peripheral device has returned no response. • A damaged ACK packet is received. • A handshake other than ACK (NAK, NYET, or STALL) is received. When the function controller mode is selected, the read value is invalid.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
4	SACK	0	R/W*	Setup Transaction Normal Response Interrupt Status Indicates the status of the setup transaction normal response interrupt when the host controller mode is selected. 0: SACK interrupts not generated 1: SACK interrupts generated This module detects the SACK interrupt when ACK response is returned from the peripheral device during the setup transactions issued by this module, and sets this bit to 1. Here, if the corresponding interrupt enable bit is set to 1, this module generates the SACK interrupt. When the function controller mode is selected, the read value is invalid.
3 to 1	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Note 1. To clear the status indicated by the bits in this register, write 0 only to the bits to be cleared; write 1 to the other bits. Do not write 0 to the status bits indicating 0.
Note 2. A change in the status indicated by the BCHG bit can be detected by this module even while the clock supply is stopped, and the interrupt is output when the corresponding interrupt enable bit is enabled. Clearing the status should be done after enabling the clock supply.
No interrupts other than BCHG can be detected while the clock supply is stopped.
* Only 0 can be written to.

28.3.18 BRDY Interrupt Status Register (BRDYSTS)

BRDYSTS is a register that indicates the BRDY interrupt status for each pipe.
This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIPEBRDY															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

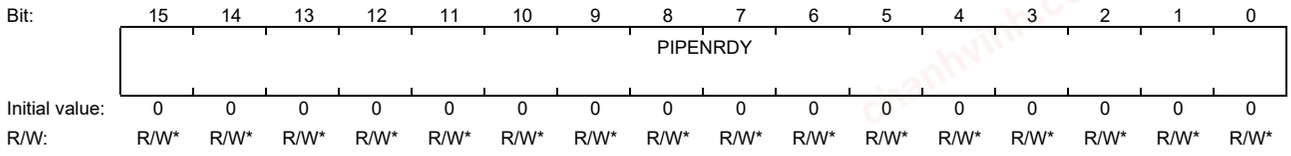
Bit	Bit Name	Initial Value	R/W	Description
15 to 0	PIPEBRDY	H'0000	R/W*1	BRDY Interrupt Status for each Pipe*2 0: Interrupts not generated 1: Interrupts generated

Note 1. The bit number corresponds to the pipe number.
Note 2. To clear the status indicated by the bits in this register when BRDYM is 0, write 0 only to the bits to be cleared; write 1 to the other bits. Do not write 0 to the status bits indicating 0.
* 1. Only 0 can be written to.
* 2. When BRDYM is 0, clearing this bit should be done before accessing the FIFO.

chanhvinh.com

28.3.19 NRDY Interrupt Status Register (NRDYSTS)

NRDYSTS is a register that indicates the NRDY interrupt status for each pipe.
This register is initialized by a power-on reset.

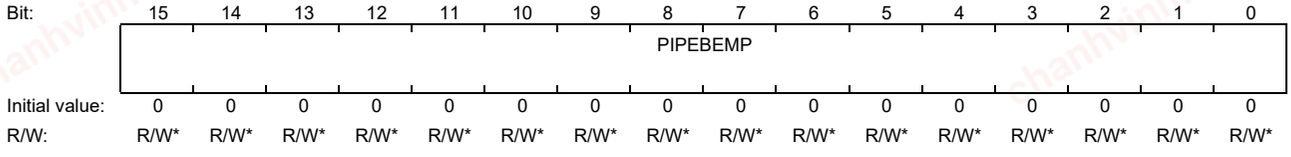


Bit	Bit Name	Initial Value	R/W	Description
15 to 0	PIPENRDY	H'0000	R/W*	NRDY Interrupt Status for each Pipe 0: Interrupts not generated 1: Interrupts generated

Note 1. The bit number corresponds to the pipe number.
Note 2. To clear the status indicated by the bits in this register, write 0 only to the bits to be cleared; write 1 to the other bits. Do not write 0 to the status bits indicating 0.
* Only 0 can be written to.

28.3.20 BEMP Interrupt Status Register (BEMPSTS)

BEMPSTS is a register that indicates the BEMP interrupt status for each pipe.
This register is initialized by a power-on reset.



Bit	Bit Name	Initial Value	R/W	Description
15 to 0	PIPEBEMP	H'0000	R/W*	BEMP Interrupt Status for each Pipe 0: Interrupts not generated 1: Interrupts generated

Note 1. The bit number corresponds to the pipe number.
Note 2. To clear the status indicated by the bits in this register, write 0 only to the bits to be cleared; write 1 to the other bits. Do not write 0 to the status bits indicating 0.
* Only 0 can be written to.

chanhvinh.com

28.3.21 Frame Number Register (FRMNUM)

FRMNUM is a register that determines the source of isochronous error notification and indicates the frame number. This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OVRN	CRCE	—	—	—	FRNM[10:0]										
Initial value:	0	0	—	—	—	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*1	R/W*1	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	OVRN	0	R/W*1	Overrun/Underrun Detection Status Indicates whether an overrun/underrun error has been detected in the pipe during isochronous transfer. On detecting either error, this module simultaneously generates the internal NRDY interrupt request. For details, refer to section 28.4.2, Interrupt Functions. 0: No error 1: An error occurred This bit can be cleared to 0 by writing 0 to the bit. Here, 1 should be written to the other bits in this register. (1) When the host controller mode is selected This module sets this bit to 1 on any of the following conditions. • For the isochronous transfer pipe in the transmitting direction, the time to issue an OUT token comes before all the transmit data has been written to the FIFO buffer. • For the isochronous transfer pipe in the receiving direction, the time to issue an IN token comes when no FIFO buffer planes are empty. (2) When the function controller mode is selected This module sets this bit to 1 on any of the following conditions. • For the isochronous transfer pipe in the transmitting direction, the IN token is received before all the transmit data has been written to the FIFO buffer. • For the isochronous transfer pipe in the receiving direction, the OUT token is received when no FIFO buffer planes are empty. Note: This bit should be used for debugging. When designing a system, control the timing so that neither overrun nor underrun occurs.
14	CRCE	0	R/W*1	Receive Data Error Indicates whether a CRC error or bit stuffing error has been detected in the pipe during isochronous transfer. On detecting either error, this module simultaneously generates the internal NRDY interrupt request. For details, refer to section 28.4.2, Interrupt Functions. 0: No error 1: An error occurred This bit can be cleared to 0 by writing 0 to the bit. Here, 1 should be written to the other bits in this register.
13 to 11	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
10 to 0	FRNM[10:0]	H'000	R	Frame Number This module sets these bits to indicate the latest frame number, which is updated every time an SOF packet is issued or received (every 1 ms). When reading these bits, repeat reading until the same value is read twice.

Note 1. Only 0 can be written to.

chanhvinh.com

28.3.22 µFrame Number Register (UFRMNUM)

UFRMNUM is a register that indicates the µframe number.
This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	UFRNM[2:0]		
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 3	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
2 to 0	UFRNM[2:0]	000	R	µFrame The µframe number can be confirmed. This module sets these bits to indicate the µframe number during high-speed operation. During operation other than high-speed operation, this module sets these bits to B'000. When reading these bits, repeat reading until the same value is read twice.

28.3.23 USB Address Register (USBADDR)

USBADDR is a register that indicates the USB address. This register is valid only when the function controller mode is selected. When the host controller mode is selected, peripheral device addresses should be set using the DEVSEL bits in PIPEMAXP.
This register is initialized by a power-on reset or a USB bus reset.

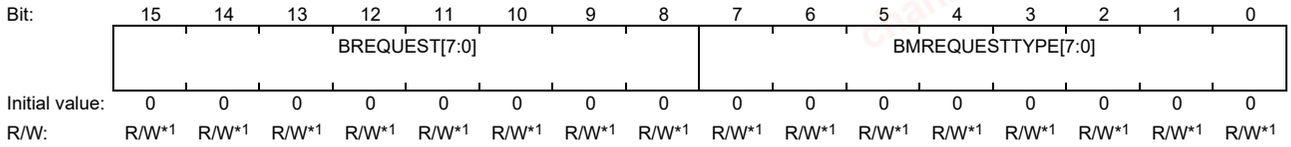
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	USBADDR[6:0]						
Initial value:	—	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
6 to 0	USBADDR[6:0]	H'00	R	USB Address When the function controller mode is selected, these bits indicate the USB address assigned by the host when the SET_ADDRESS request is successfully processed. When the function controller mode is selected, these bits indicate H'00 upon detection of a USB bus reset signal. When the host controller mode is selected, the read value is invalid.

chanhvinh.com

28.3.24 USB Request Type Register (USBREQ)

USBREQ is a register that stores setup requests for control transfers.
When the function controller mode is selected, the values of bRequest and bmRequestType that have been received are stored. When the host controller mode is selected, the values of bRequest and bmRequestType to be transmitted are set. This register is initialized by a power-on reset or a USB bus reset.



Bit	Bit Name	Initial Value	R/W	Description
15 to 8	BREQUEST[7:0]	H'00	R/W*1	Request These bits store the USB request bRequest value. (1) When the host controller mode is selected The USB request data value for the setup transaction to be transmitted should be set in these bits. After SUREQ has been set to 1, do not modify these bits until SUREQ is read as 0. (2) When the function controller mode is selected Indicates the USB request data value received during the setup transaction. Writing to these bits is invalid.
7 to 0	BMREQUEST TYPE[7:0]	H'00	R/W*1	Request Type These bits store the USB request bmRequestType value. (1) When the host controller mode is selected The USB request type value for the setup transaction to be transmitted should be set in these bits. After SUREQ has been set to 1, do not modify these bits until SUREQ is read as 0. (2) When the function controller mode is selected Indicates the USB request type value received during the setup transaction. Writing to these bits is invalid.

Note 1. When the function controller mode is selected, these bits can only be read, and writing to these bits is invalid. When the host controller mode is selected, these bits can be read and written to.

chanhvinh.com

28.3.25 USB Request Value Register (USBVAL)

USBVAL is a register that stores setup requests for control transfers. When the function controller mode is selected, the value of wValue that has been received is stored. When the host controller mode is selected, the value of wValue to be transmitted is set.

This register is initialized by a power-on reset or a USB bus reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WVALUE[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	WVALUE[15:0]	H'0000	R/W*1	Value These bits store the USB request wValue value. (1) When the host controller mode is selected The USB request wValue value for the setup transaction to be transmitted should be set in these bits. After SUREQ has been set to 1, do not modify these bits until SUREQ is read as 0. (2) When the function controller mode is selected Indicates the USB request wValue value received during the setup transaction. Writing to these bits is invalid.

Note 1. When the function controller mode is selected, these bits can only be read, and writing to these bits is invalid. When the host controller mode is selected, these bits can be read and written to.

28.3.26 USB Request Index Register (USBINDX)

USBINDX is a register that stores setup requests for control transfers. When the function controller mode is selected, the value of wIndex that has been received is stored. When the host controller mode is selected, the value of wIndex to be transmitted is set.

This register is initialized by a power-on reset or a USB bus reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WINDEX[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	WINDEX[15:0]	H'0000	R/W*1	Index These bits store the USB request wIndex value. (1) When the host controller mode is selected The USB request wIndex value for the setup transaction to be transmitted should be set in these bits. After SUREQ has been set to 1, do not modify these bits until SUREQ is read as 0. (2) When the function controller mode is selected Indicates the USB request wIndex value received during the setup transaction. Writing to these bits is invalid.

Note 1. When the function controller mode is selected, these bits can only be read, and writing to these bits is invalid. When the host controller mode is selected, these bits can be read and written to.

chanhvinh.com

28.3.27 USB Request Length Register (USBLENG)

USBLENG is a register that stores setup requests for control transfers. When the function controller mode is selected, the value of wLength that has been received is stored. When the host controller mode is selected, the value of wLength to be transmitted is set.

This register is initialized by a power-on reset or a USB bus reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WLENGTH[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	WLENGTH[15:0]	H'0000	R/W*1	Length These bits store the USB request wLength value. (1) When the host controller mode is selected The USB request wLength value for the setup transaction to be transmitted should be set in these bits. After SUREQ has been set to 1, do not modify these bits until SUREQ is read as 0. (2) When the function controller mode is selected Indicates the USB request wLength value received during the setup transaction. Writing to these bits is invalid.

Note 1. When the function controller mode is selected, these bits can only be read, and writing to these bits is invalid. When the host controller mode is selected, these bits can be read and written to.

28.3.28 DCP Configuration Register (DCPCFG)

DCPCFG is a register that specifies the data transfer direction for the default control pipe (DCP).

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CNTMD	SHTNAK	—	—	DIR	—	—	—	—
Initial value:	—	—	—	—	—	—	—	0	0	—	—	0	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
8	CNTMD	0	R/W	Continuous Transfer Mode Specifies whether to use the default control pipe in continuous transfer mode. 0: Non-continuous transfer mode 1: Continuous transfer mode This module determines whether transmitting to/receiving from the FIFO buffer allocated for the DCP has completed or not using this bit setting, as shown in Table 28.10. Modify this bit while CSSTS is 0, PID is NAK, and the pipe is not selected by the CURPIPE bits. To modify this bit after completing USB communication using the DCP, write 1 to BCLR to clear the FIFO buffer assigned to the DCP while the CSSTS, PID, and CURPIPE bits are in the above-described state. Before modifying these bits after modifying the PID bits for the DCP from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked.
7	SHTNAK	0	R/W	Pipe Disabled at End of Transfer Specifies whether to modify PID to NAK upon the end of transfer when the default control pipe is in the receiving direction. 0: Pipe continued at the end of transfer 1: Pipe disabled at the end of transfer When this bit is set to 1, this module modifies the PID bits corresponding to the DCP to NAK on determining the end of the transfer. This module determines that the transfer has ended when a short packet (including a zero-length packet) is successfully received. Modify this bit while CSSTS is 0 and PID is NAK. Before modifying this bit after modifying the PID bits for the DCP from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked. This bit should be cleared to 0 for the DCP in the transmitting direction.
6, 5	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
4	DIR	0	R/W	Transfer Direction When the host controller mode is selected, this bit sets the transfer direction of data stage and status stage for control transfers. 0: Data receiving direction 1: Data transmitting direction When the function controller mode is selected, this bit should be cleared to 0.
3 to 0	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.

chanhvinh.com

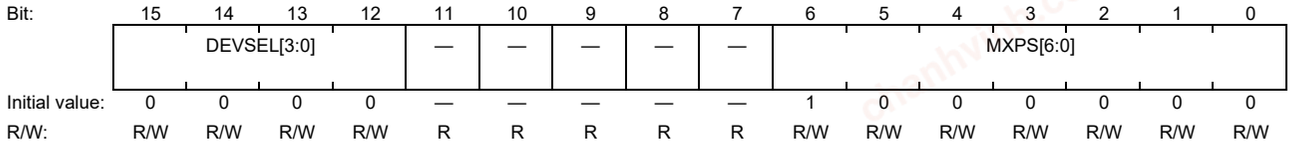
Table 28.10 Relationship between Transfer Mode Settings by CNTMD Bit and Timings at which Reading Data or Transmitting Data from FIFO Buffer is Enabled

Continuous or Non-Continuous Transfer Mode	When Reading Data or Transmitting Data is Enabled
Non-continuous transfer (CNTMD = 0)	In the receiving direction (DIR = 0), reading data from the FIFO buffer is enabled in the following case. This module receives a single packet.
	In the transmitting direction (DIR = 1), transmitting data from the FIFO buffer is enabled in ether of the following cases. (1) Data of the maximum packet size is written to the FIFO buffer. (2) Data of the short packet size (including 0-byte data) is written to the FIFO buffer and then 1 is written to the BVAL bit
Continuous transfer (CNTMD = 1)	In the receiving direction (DIR = 0), reading data from the FIFO buffer is enabled in any of the following cases. (1) The number of the data bytes received in the FIFO buffer assigned to the DCP becomes the same as the number of assigned data bytes (fixed to 256 bytes) (2) This module receives a short packet other than a zero-length packet (3) This module receives a zero-length packet when data is already stored in the FIFO buffer assigned to the DCP.
	In the transmitting direction (DIR = 1), transmitting data from the FIFO buffer is enabled in either of the following cases. (1) The number of the data bytes written to the FIFO buffer becomes the same as the number of data bytes in a single FIFO buffer plane assigned to the DCP. (2) A number of data bytes less than the size of a single FIFO buffer (including 0-byte data) assigned to the DCP being written to the FIFO buffer and then 1 being written to BVAL.

chanhvinh.com

28.3.29 DCP Maximum Packet Size Register (DCPMAXP)

DCPMAXP is a register that specifies the maximum packet size for the DCP.
This register is initialized by a power-on reset.



Bit	Bit Name	Initial Value	R/W	Description
15 to 12	DEVSEL[3:0]	0000	R/W	Device Select When the host controller mode is selected, these bits specify the communication target peripheral device address for control transfers. 0000: Address 0000 0001: Address 0001 : : 1001: Address 1001 1010: Address 1010 Others: Setting prohibited These bits should be set after setting the DEVADDn register corresponding to the value to be set in these bits. For example, before setting DEVSEL to 0010, the DEVADD2 register should be set first. These bits should be set while CSSTS is 0, PID is NAK, and SUREQ is 0. Before modifying these bits after modifying the PID bits for the DCP from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked. When the function controller mode is selected, these bits should be set to B'0000.
11 to 7	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
6 to 0	MXPS[6:0]	H'40	R/W	Maximum Packet Size Specifies the maximum data payload (maximum packet size) for the DCP. These bits are initialized to H'40 (64 bytes). These bits should be set to the value based on the USB Specification. These bits should be set while CSSTS is 0, PID is NAK, and the pipe is not selected by the CURPIPE bits. Before modifying these bits after modifying the PID bits for the DCP from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked. While MXPS is 0, do not write to the FIFO buffer or do not set PID to BUF.

chanhvinh.com

28.3.30 DCP Control Register (DCPCTR)

DCPCTR is a register that is used to confirm the buffer memory status, change and confirm the data PID sequence bit, and set the response PID for the DCP.

This register is initialized by a power-on reset. The CCPL and PID[1:0] bits are initialized by a USB bus reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BSTS	SUREQ	CSCLR	CSSTS	SUREQCLR	—	—	SQCLR	SQSET	SQMON	PBUSY	PINGE	—	CCPL	PID[1:0]	
Initial value:	0	0	0	0	0	—	—	0	0	1	0	0	—	0	0	0
R/W:	R	R/W*2	R/W*1	R	R/W*1	R	R	R/W*1	R/W*1	R	R	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	BSTS	0	R	Buffer Status Indicates whether DCP FIFO buffer access is enabled or disabled. 0: Buffer access is disabled. 1: Buffer access is enabled. The meaning of the BSTS bit depends on the ISEL bit setting as follows. • When ISEL = 0, BSTS indicates whether the received data can be read from the buffer. • When ISEL = 1, BSTS indicates whether the data to be transmitted can be written to the buffer.
14	SUREQ	0	R/W*2	Setup Token Transmission Transmits the setup packet by setting this bit to 1 when the host controller mode is selected. 0: Writing invalid 1: Transmits the setup packet. After completing the setup transaction process, this module generates either the SACK or SIGN interrupt and clears this bit to 0. This module also clears this bit to 0 when the SUREQCLR bit is set to 1. Before setting this bit to 1, set the DEVSEL bits, USBREQ register, USBVAL register, USBINDX register, and USBLENG register appropriately to transmit the desired USB request in the setup transaction. Before setting this bit to 1, check that the PID bits for the DCP are set to NAK. After setting this bit to 1, do not modify the DEVSEL bits, USBREQ register, USBVAL register, USBINDX register, or USBLENG register while the setup transaction is in progress (SUREQ = 1). Write 1 to this bit only when transmitting the setup token; for the other purposes, write 0. When the function controller mode is selected, be sure to write 0 to this bit.
13	CSCLR	0	R/W*1	C-SPLIT Status Clear for Split Transaction When the host controller mode is selected, setting this bit to 1 clears the CSSTS bit to 0 for the transfer using the split transaction. In this case, the next DCP transfer restarts with the S-SPLIT. 0: Writing invalid 1: Clears the CSSTS bit to 0. When this bit is set to 1, this module clears the CSSTS bit to 0. For the transfer using the split transaction, to restart the next transfer with the S-SPLIT forcibly, set this bit to 1. However, for the normal split transaction, this module automatically clears the CSSTS bit to 0 upon completion of the C-SPLIT; therefore, clearing the CSSTS bit is not necessary. Controlling the CSSTS bit through this bit must be done while UACT is 0 and thus communication is halted or while no transfer is being performed with bus disconnection detected. Setting this bit to 1 while CSSTS is 0 has no effect. When the function controller mode is selected, be sure to write 0 to this bit.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
12	CSSTS	0	R	COMPLETE SPLIT (C-SPLIT) Status of Split Transaction Indicates the C-SPLIT status of the split transaction when the host controller mode is selected. 0: START-SPLIT (S-SPLIT) transaction being processed or the device not using the split transaction being processed 1: C-SPLIT transaction being processed This module sets this bit to 1 upon start of the C-SPLIT and clears this bit to 0 upon detection of C-SPLIT completion. When the function controller mode is selected, the read value is invalid.
11	SUREQCLR	0	R/W*1	SUREQ Bit Clear When the host controller mode is selected, setting this bit to 1 clears the SUREQ bit to 0. 0: Writing invalid 1: Clears the SUREQ bit to 0. This bit always indicates 0. Set this bit to 1 when communication has stopped with SUREQ being 1 during the setup transaction. However, for normal setup transactions, this module automatically clears the SUREQ bit to 0 upon completion of the transaction; therefore, clearing the SUREQ bit is not necessary. Controlling the SUREQ bit through this bit must be done while UACT is 0 and thus communication is halted or while no transfer is being performed with bus disconnection detected. When the function controller mode is selected, be sure to write 0 to this bit.
10, 9	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
8	SQCLR	0	R/W*1	Toggle Bit Clear Specifies DATA0 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer. 0: Writing invalid 1: Specifies DATA0. This bit always indicates 0. Do not set the SQCLR and SQSET bits to 1 simultaneously. Set this bit to 1 while CSSTS is 0, PID is NAK, and the pipe is not selected by the CURPIPE bits. Before setting this bit to 1 after modifying the PID bits for the DCP from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked.
7	SQSET	0	R/W*1	Toggle Bit Set Specifies DATA1 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer. 0: Writing invalid 1: Specifies DATA1. Do not set the SQCLR and SQSET bits to 1 simultaneously. Set this bit to 1 while CSSTS is 0 and PID is NAK. Before setting this bit to 1 after modifying the PID bits for the DCP from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked.
6	SQMON	1	R	Sequence Toggle Bit Monitor Indicates the expected value of the sequence toggle bit for the next transaction during the DCP transfer. 0: DATA0 1: DATA1 This module allows this bit to toggle upon normal completion of the transaction. However, this bit is not allowed to toggle when a DATA-PID disagreement occurs during the transfer in the receiving direction. When the function controller mode is selected, this module sets this bit to 1 (specifies DATA1 as the expected value) upon normal reception of the setup packet. When the function controller mode is selected, this module does not reference to this bit during the IN/OUT transaction of the status stage, and does not allow this bit to toggle upon normal completion.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
5	PBUSY	0	R	Pipe Busy Indicates whether or not the actual communication state of the DCP has entered the NAK state when the PID bits for the DCP are changed from BUF to NAK. 0: Has not finished the transition to NAK 1: Has finished the transition to NAK This module modifies this bit from 0 to 1 upon start of the USB transaction for the DCP, and modifies the bit from 1 to 0 upon completion of one transaction. Reading this bit after PID has been set to NAK allows checking whether modification of the pipe settings is possible. For details, refer to section 28.4.3 (1) Pipe Control Register Switching Procedures.
4	PINGE	0	R/W	PING Token Issue Enable When the host controller mode is selected, setting this bit to 1 allows this module to issue the PING token during transfers in the transmitting direction and start a transfer in the transmitting direction with the PING transaction. 0: Disables issuing PING token. 1: Enables normal PING operation. When having detected the ACK handshake during PING transactions, this module performs the OUT transaction as the next transaction. When having detected the NAK handshake or NYET handshake during OUT transactions, this module performs the PING transaction as the next transaction. When the host controller mode is selected, setting this bit to 0 prevents this module from issuing the PING token during transfers in the transmitting direction and only allows this module to perform OUT transactions for the transfers in the transmitting direction. This bit should be modified while CSSTS is 0 and PID is NAK. Before setting this bit to 1 after modifying the PID bits for the DCP from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked. When the function controller mode is selected, be sure to write 0 to this bit.
3	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
2	CCPL	0	R/W	Control Transfer End Enable When the function controller mode is selected, setting this bit to 1 enables the status stage of the control transfer to be completed. 0: Completion of control transfer is disabled. 1: Completion of control transfer is enabled. In function controller mode, when this bit is set to 1 while the corresponding PID bits are set to BUF, this module completes the control transfer status stage. Specifically, during control read transfer, this module transmits the ACK handshake in response to the OUT transaction from the USB host, and outputs the zero-length packet in response to the IN transaction from the USB host during control write or no-data control transfer. However, on detecting the SET_ADDRESS request, this module operates in auto response mode from the setup stage up to the status stage completion irrespective of the setting of this bit. This module modifies this bit from 1 to 0 on receiving the new setup packet. 1 cannot be written to this bit while VALID is 1. When the host controller mode is selected, be sure to write 0 to this bit.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PID[1:0]	00	R/W	Response PID Controls the response type of this module during control transfer. 00: NAK response 01: BUF response (depending on the buffer state) 10: STALL response 11: STALL response (1) When the host controller mode is selected Modify the setting of these bits from NAK to BUF using the following procedure. • When the transmitting direction is set Write all the transmit data to the FIFO buffer while UACT is 1 and PID is NAK, and then set PID to BUF. After PID has been set to BUF, this module executes the OUT transaction (or PING transaction). • When the receiving direction is set Check that the FIFO buffer is empty (or empty the buffer) while UACT is 1 and PID is NAK, and then set PID to BUF. After PID has been set to BUF, this module executes the IN transaction. This module modifies the setting of these bits as follows. • This module sets PID to STALL (11) on receiving the data of the size exceeding the maximum packet size when PID has been set to BUF. • This module sets PID to NAK on detecting a reception error such as a CRC error three consecutive times. • This module also sets PID to STALL (11) on receiving the STALL handshake. Even if the PID bits are modified to NAK after this module has issued S-SPLIT of the split transaction for the selected pipe (while CSSTS indicates 1), this module continues the transaction until C-SPLIT completes. This module sets PID to NAK upon completion of C-SPLIT. (2) When the function controller mode is selected This module modifies the setting of these bits as follows. • This module modifies PID to NAK on receiving the setup packet. Here, this module sets VALID to 1. The setting of PID cannot be modified until VALID is set to 0. • This module sets PID to STALL (11) on receiving the data of the size exceeding the maximum packet size when PID has been set to BUF. • This module sets PID to STALL (1x) on detecting the control transfer sequence error. • This module sets PID to NAK on detecting the USB bus reset. This module does not reference to the setting of the PID bits while the SET_ADDRESS request is processed (auto processing).

Note 1. Only 0 can be read and 1 can be written to.
Note 2. Only 1 can be written to.

chanhvinh.com

28.3.31 Pipe Window Select Register (PIPESEL)

PIPE1 to PIPE9 should be set using PIPESEL, PIPECFG, PIPEBUF, PIPEMAXP, PIPEPERI, PIPEnCTR, PIPEnTRE, and PIPEnTRN.

After selecting the pipe using PIPESEL, functions of the pipe should be set using PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI. PIPEnCTR, PIPEnTRE, and PIPEnTRN can be set regardless of the pipe selection in PIPESEL.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	PIPESEL[3:0]			
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
3 to 0	PIPESEL[3:0]	0000	R/W	Pipe Window Select When a value between 0001 and 1111 is set in these bits, the information and settings for the corresponding pipe can be read from the PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI registers. 0000: No pipe selected 0001: PIPE1 0010: PIPE2 0011: PIPE3 0100: PIPE4 0101: PIPE5 0110: PIPE6 0111: PIPE7 1000: PIPE8 1001: PIPE9 1010: PIPE10 1011: PIPE11 1100: PIPE12 1101: PIPE13 1110: PIPE14 1111: PIPE15 Others: Setting prohibited When 0000 is set in these bits, 0 is read from all of the bits in the PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI registers. Writing to the bits in these registers is invalid.

chanhvinh.com

28.3.32 Pipe Configuration Register (PIPECFG)

PIPECFG is a register that specifies the transfer type, buffer memory access direction, and endpoint numbers for PIPE1 to PIPE15. It also selects continuous or non-continuous transfer mode, single or double buffer mode, and whether to continue or disable pipe operation at the end of transfer.
This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TYPE[1:0]		—	—	—	BFRE	DBLB	CNTMD	SHTNAK	—	—	DIR	EPNUM[3:0]			
Initial value:	0	0	—	—	—	0	0	0	0	—	—	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	TYPE[1:0]	00	R/W	Transfer Type Specifies the transfer type for the pipe selected by the PIPESEL bits (selected pipe). • PIPE1 and PIPE2 00: Pipe not used 01: Bulk transfer 10: Setting prohibited 11: Isochronous transfer • PIPE3 to PIPE5 00: Pipe not used 01: Bulk transfer 10: Setting prohibited 11: Setting prohibited • PIPE6 to PIPE8 00: Pipe not used 01: Pipe not used 10: Interrupt transfer 11: Setting prohibited • PIPE9 00: Pipe not used 01: Bulk transfer (when in function controller mode) 10: Interrupt transfer (when in host controller mode) 11: Setting prohibited • PIPE10 (available only in function controller mode) 01: Bulk transfer 10: Interrupt transfer • PIPE11 to PIPE15 (available only in function controller mode) 01: Bulk transfer Before setting PID to BUF for the selected pipe (before starting USB communication using the selected pipe), be sure to set these bits to the value other than 00. Modify these bits while the PID bits for the selected pipe are set to NAK. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked.
13 to 11	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
10	BFRE	0	R/W	BRDY Interrupt Operation Specification Specifies the BRDY interrupt generation timing from this module to the CPU with respect to the selected pipe. 0: BRDY interrupt upon transmitting or receiving data 1: BRDY interrupt upon completion of reading data This bit is valid when PIPE1 to PIPE5 and PIPE9 to PIPE15 are selected. When this bit is set to 1 and the selected pipe is in the receiving direction, this module detects the transfer completion and generates the BRDY interrupt on having read the pertinent packet. When the BRDY interrupt is generated with the above conditions, 1 needs to be written to BCLR. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to BCLR. When this bit is set to 1 and the selected pipe is in the transmitting direction, this module does not generate the BRDY interrupt. For details, refer to section 28.4.2 (2) BRDY Interrupt. Modify this bit while CSSTS is 0, PID is NAK, and the pipe is not selected by the CURPIPE bits. To modify this bit after completing USB communication using the selected pipe, write 1 and then 0 to ACLRM continuously to clear the FIFO buffer assigned to the selected pipe while the CSSTS, PID, and CURPIPE bits are in the above-described state. Before modifying this bit after modifying the PID bits for the selected pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked.
9	DBLB	0	R/W	Double Buffer Mode Selects either single or double buffer mode for the FIFO buffer used by the selected pipe. 0: Single buffer 1: Double buffer This bit is valid when PIPE1 to PIPE5 and PIPE9 to PIPE15 are selected. For PIPE9 and PIPE10, this bit is only valid when the transfer type is set to bulk transfer. When this bit is set to 1, this module assigns two planes of the FIFO buffer size specified by the BUFSIZE bits in PIPEBUF to the selected pipe. Specifically, the following expression determines the FIFO buffer size assigned to the selected pipe by this module. (BUFSIZE + 1) × 64 × (DBLB + 1) [bytes] Modify this bit while CSSTS is 0, PID is NAK, and the pipe is not selected by the CURPIPE bits. To modify this bit after completing USB communication using the selected pipe, write 1 and then 0 to ACLRM continuously to clear the FIFO buffer assigned to the selected pipe while the CSSTS, PID, and CURPIPE bits are in the above-described state. Before modifying this bit after modifying the PID bits for the selected pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked.
8	CNTMD	0	R/W	Continuous Transfer Mode Specifies whether to use the selected pipe in continuous transfer mode. 0: Non-continuous transfer mode 1: Continuous transfer mode This bit is valid when PIPE1 to PIPE5 and PIPE9 to PIPE15 are selected by the PIPESEL bits and bulk transfer is selected (TYPE = 01). Modify this bit while CSSTS is 0, PID is NAK, and the pipe is not selected by the CURPIPE bits. To modify this bit after completing USB communication using the selected pipe, write 1 and then 0 to ACLRM continuously to clear the FIFO buffer assigned to the selected pipe while the CSSTS, PID, and CURPIPE bits are in the above-described state. Before modifying this bit after modifying the PID bits for the selected pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
7	SHTNAK	0	R/W	Pipe Disabled at End of Transfer Specifies whether to modify PID to NAK upon the end of transfer when the selected pipe is in the receiving direction. 0: Pipe continued at the end of transfer 1: Pipe disabled at the end of transfer This bit is valid when the selected pipe is PIPE1 to PIPE5 and PIPE9 to PIPE15 in the receiving direction. When this bit is set to 1 for the selected pipe in the receiving direction, this module modifies the PID bits corresponding to the selected pipe to NAK on determining the end of the transfer. This module determines that the transfer has ended on any of the following conditions. • A short packet (including a zero-length packet) is successfully received. • The transaction counter is used and the number of packets specified by the counter is successfully received. Modify this bit while CSSTS is 0 and PID is NAK. Before modifying this bit after modifying the PID bits for the selected pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked. This bit should be cleared to 0 for the pipe in the transmitting direction.
6, 5	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
4	DIR	0	R/W	Transfer Direction Specifies the transfer direction for the selected pipe. 0: Receiving direction 1: Transmitting direction When this bit is set to 0, this module uses the selected pipe in the receiving direction, and when this bit is set to 1, this module uses the selected pipe in the transmitting direction. Modify this bit while CSSTS is 0, PID is NAK, and the pipe is not selected by the CURPIPE bits. To modify this bit after completing USB communication using the selected pipe, write 1 and then 0 to ACLRM continuously to clear the FIFO buffer assigned to the selected pipe while the CSSTS, PID, and CURPIPE bits are in the above-described state. Before modifying this bit after modifying the PID bits for the selected pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked.
3 to 0	EPNUM[3:0]	0000	R/W	Endpoint Number These bits specify the endpoint number for the selected pipe. Setting 0000 means unused pipe. Modify these bits while CSSTS is 0, PID is NAK, and the pipe is not selected by the CURPIPE bits. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked. Do not make the settings such that the combination of the set values in the DIR and EPNUM bits should be the same for two or more pipes (EPNUM = 0000 can be set for all the pipes).

chanhvinh.com

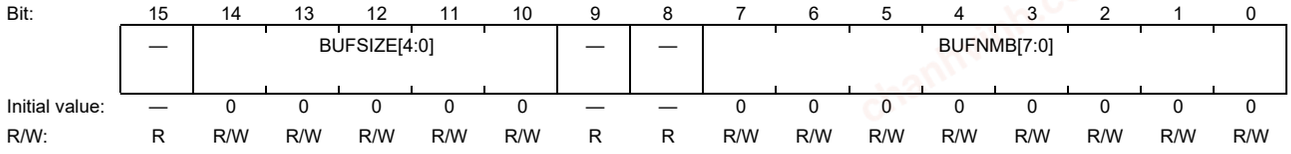
Table 28.11 Relationship between Transfer Mode Settings by CNTMD Bit and Timings at which Reading Data or Transmitting Data from FIFO Buffer is Enabled

Continuous or Non-Continuous Transfer Mode	When Reading Data or Transmitting Data is Enabled
Non-continuous transfer (CNTMD = 0)	In the receiving direction (DIR = 0), reading data from the FIFO buffer is enabled in the following case. This module receives a single packet.
	In the transmitting direction (DIR = 1), transmitting data from the FIFO buffer is enabled in either of the following cases. (1) Data of the maximum packet size is written to the FIFO buffer. (2) Data of the short packet size (including 0-byte data) is written to the FIFO buffer and then 1 is written to BVAL.
Continuous transfer (CNTMD = 1)	In the receiving direction (DIR = 0), reading data from the FIFO buffer is enabled in any of the following cases. (1) The number of the data bytes received in the FIFO buffer assigned to the selected pipe becomes the same as the number of assigned data bytes ((BUFSIZE + 1) × 64). (2) This module receives a short packet other than a zero-length packet. (3) This module receives a zero-length packet when data is already stored in the FIFO buffer assigned to the selected pipe. (4) This module receives the number of packets equal to the transaction counter value specified for the selected pipe.
	In the transmitting direction (DIR = 1), transmitting data from the FIFO buffer is enabled in any of the following cases. (1) The number of the data bytes written to the FIFO buffer becomes the same as the number of data bytes in a single FIFO buffer plane assigned to the selected pipe. (2) A number of data bytes less than the size of a single FIFO buffer (including 0 bytes of data) assigned to the selected pipe being written to the FIFO buffer and then 1 being written to BVAL (3) After setting the DMA transfer end sampling enable bit (TENDE) to 1, a number of data bytes less than the size of a single FIFO buffer (including 0 bytes of data) assigned to the selected pipe being written to the FIFO buffer by DMA transfer and the DMA transfer end signal is received when the last byte is written.

chanhvinh.com

28.3.33 Pipe Buffer Setting Register (PIPEBUF)

PIPEBUF is a register that specifies the buffer size and buffer number for PIPE1 to PIPE9.
This register is initialized by a power-on reset.



Bit	Bit Name	Initial Value	R/W	Description
15	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
14 to 10	BUFSIZE[4:0]	H'00	R/W	Buffer Size Specifies the size of the buffer for the pipe selected by the PIPESEL bits (selected pipe) in terms of blocks, where one block comprises 64 bytes. 00000 (H'00): 64 bytes 00001 (H'01): 128 bytes : : 11111 (H'1F): 2 Kbytes When the DBLB bit is set to 1, this module assigns two planes of the FIFO buffer size specified by the BUFSIZE bits to the selected pipe. Specifically, the following expression determines the FIFO buffer size assigned to the selected pipe by this module. (BUFSIZE + 1) × 64 × (DBLB + 1) [bytes] The valid value for these bits depends on the selected pipe. • PIPE1 to PIPE5 and PIPE9 to PIPE15: Any value from H'00 to H'1F is valid. • PIPE6 to PIPE8: H'00 should be set. When used with CNTMD = 1, set an integral multiple of the maximum packet size to the BUFSIZE bits. Modify these bits while CSSTS is 0, PID is NAK, and the pipe is not selected by the CURPIPE bits. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked.
9, 8	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.

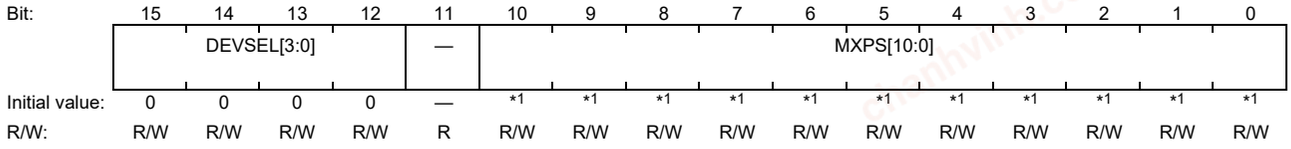
chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	BUFNMB[7:0]	H'00	R/W	Buffer Number The first block number in the FIFO buffer to be allocated for the selected pipe should be set in these bits. The FIFO buffer blocks allocated for the selected pipe by this module are determined as follows: Block number: BUFNMB to block number of (BUFNMB + (BUFSIZE + 1) × (DBLB + 1) – 1) These bits should be set to a value from H'04 to H'7F. BUFNMB = H'00 is used exclusively for DCP. BUFNMB = H'04 is used exclusively for PIPE6. When PIPE6 is not used, H'04 can be used for other pipes. When PIPE6 is selected, writing to these bits is invalid and H'04 is automatically assigned by this module. BUFNMB = H'05 is used exclusively for PIPE7. When PIPE7 is not used, H'05 can be used for other pipes. When PIPE7 is selected, writing to these bits is invalid and H'05 is automatically assigned by this module. BUFNMB = H'06 is used exclusively for PIPE8. When PIPE8 is not used, H'06 can be used for other pipes. When PIPE8 is selected, writing to these bits is invalid and H'06 is automatically assigned by this module. BUFNMB = H'07 is used exclusively for PIPE9. When PIPE9 is not used, H'07 can be used for other pipes. When PIPE9 is selected, writing to these bits is valid and H'07 is assigned as the initial value by this module. Modify these bits while CSSTS is 0, PID is NAK, and the pipe is not selected by the CURPIPE bits. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked.

chanhvinh.com

28.3.34 Pipe Maximum Packet Size Register (PIPEMAXP)

PIPEMAXP is a register that specifies the maximum packet size for PIPE1 to PIPE15.
This register is initialized by a power-on reset.



Bit	Bit Name	Initial Value	R/W	Description
15 to 12	DEVSEL[3:0]	0000	R/W	Device Select When the host controller mode is selected, these bits specify the USB address of the communication target peripheral device. 0000: Address 0000 0001: Address 0001 0010: Address 0010 : : 1010: Address 1010 Others: Setting prohibited These bits should be set after setting the address to the DEVADDn (n = 0 to 9, and A) register corresponding to the value to be set in these bits. For example, before setting DEVSEL to 0010, the DEVADD2 address should be set first. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked. When the function controller mode is selected, these bits should be set to B'0000.
11	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
10 to 0	MXPS[10:0]	*1	R/W	Maximum Packet Size Specifies the maximum data payload (maximum packet size) for the selected pipe. The valid value for these bits depends on the pipe as follows. PIPE1, PIPE2: 1 byte (H'001) to 1,024 bytes (H'400) PIPE3 to PIPE5: 8 bytes (H'008), 16 bytes (H'010), 32 bytes (H'020), 64 bytes (H'040), and 512 bytes (H'200) (Bits [2:0] are not available.) PIPE6 to PIPE8: 1 byte (H'001) to 64 bytes (H'040) PIPE9 (when in host controller mode): 1 byte (H'001) to 64 bytes (H'040) PIPE9 (when in function controller mode): 8 bytes (H'008), 16 bytes (H'010), 32 bytes (H'020), 64 bytes (H'040), 512 bytes (H'200) (Bits [2:0] are not available.) PIPE10 to PIPE15: 8 bytes (H'008), 16 bytes (H'010), 32 bytes (H'020), 64 bytes (H'040), 512 bytes (H'200) (Bits [2:0] are not available.) These bits should be set to the approp3riate value for each transfer type based on the USB Specification. For split transactions using the isochronous pipe, these bits should be set to 188 bytes or less. Modify these bits while CSSTS is 0, PID is NAK, and the pipe is not selected by the CURPIPE bits. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked. While MXPS is 0, do not write to the FIFO buffer or set PID to BUF.

Note 1. The initial value of MXPS is H'000 when no pipe is selected with the PIPESEL bits in PIPESEL and H'040 when a pipe is selected with the PIPESEL bits in PIPESEL.

chanhvinh.com

28.3.35 Pipe Timing Control Register (PIPEPERI)

PIPEPERI is a register that selects whether the buffer is flushed or not when an interval error occurred during isochronous IN transfer, and sets the interval error detection interval for PIPE1 to PIPE15.
This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	IFIS	—	—	—	—	—	—	—	—	—	IITV[2:0]		
Initial value:	—	—	—	0	—	—	—	—	—	—	—	—	—	0	0	0
R/W:	R	R	R	R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
12	IFIS	0	R/W	Isochronous IN Buffer Flush Specifies whether to flush the buffer when the pipe selected by the PIPESEL bits is used for isochronous IN transfers. 0: The buffer is not flushed. 1: The buffer is flushed. When the function controller mode is selected and the selected pipe is for isochronous IN transfers, this module automatically clears the FIFO buffer when this module fails to receive the IN token from the USB host within the interval set by the IITV bits in terms of (μ) frames. In double buffer mode (DBLB = 1), this module only clears the data in the plane used earlier. This module clears the FIFO buffer on receiving the SOF packet immediately after the (μ) frame in which this module has expected to receive the IN token. Even if the SOF packet is corrupted, this module also clears the FIFO buffer at the right timing to receive the SOF packet by using the internal interpolation. When the host controller mode is selected, set this bit to 0. When the selected pipe is not for the isochronous transfer, set this bit to 0.
11 to 3	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
2 to 0	IITV[2:0]	000	R/W	Interval Error Detection Interval Specifies the interval error detection timing for the selected pipe in terms of frames, which is expressed as n-th power of 2 (n is the value to be set). As described later, the detailed functions are different in host controller mode and in function controller mode. Modify these bits while CSSTS is 0, PID is NAK, and the pipe is not selected by the CURPIPE bits. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked. Before modifying these bits after USB communication has been completed with these bits set to a certain value, set PID to NAK and then set ACLRM to 1 to initialize the interval timer. The IITV bits are invalid for PIPE3 to PIPE5 and PIPE10 to PIPE15; set these bits to 000 for these pipes.

chanhvinh.com

28.3.36 PIPEn Control Registers (PIPEnCTR) (n = 1 to F)

PIPEnCTR is a register that is used to confirm the buffer memory status for the corresponding pipe, change and confirm the data PID sequence bit, determine whether auto response mode is set, determine whether auto buffer clear mode is set, and set a response PID for PIPE1 to PIPE15. This register can be set regardless of the pipe selection in PIPESEL. These registers are initialized by a power-on reset. The PID[1:0] bits are initialized by a USB bus reset.

(1) PIPEnCTR (n = 1 to 5, 9, A to F)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BSTS	INBUFM	CSCLR	CSSTS	—	ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Initial value:	0	0	0	0	—	0	0	0	0	0	0	—	—	—	0	0
R/W:	R	R	R/W*2	R	R	R/W	R/W	R/W*1	R/W*1	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	BSTS	0	R	Buffer Status Indicates whether or not the FIFO buffer allocated for the pertinent pipe can be accessed by the CPU. 0: Buffer access by the CPU is disabled. 1: Buffer access by the CPU is enabled. The meaning of this bit depends on the settings of the DIR, BFRE, and DCLRM bits as shown in Table 28.12.
14	INBUFM	0	R	Transmission Buffer Monitor Indicates the pertinent FIFO buffer status when the pertinent pipe is in the transmitting direction. 0: There is no transmissible data in the buffer memory. 1: There is transmissible data in the buffer memory. When the pertinent pipe is in the transmitting direction (DIR = 1), this module sets this bit to 1 when writing data to at least one FIFO buffer plane is completed. This module sets this bit to 0 when this module completes transmitting the data from the FIFO buffer plane to which all the data has been written. In double buffer mode (DBLB = 1), this module sets this bit to 0 after this module has completed transmitting the data from both FIFO buffer planes but before it has completed writing data to a single FIFO buffer plane. This bit indicates the same value as the BSTS bit when the pertinent pipe is in the receiving direction (DIR = 0).
13	CSCLR	0	R/W*2	C-SPLIT Status Clear Bit When the host controller mode is selected, setting this bit to 1 allows this module to clear the CSSTS bit to 0. 0: Writing invalid 1: Clears the CSSTS bit to 0. For the transfer using the split transaction, to restart the next transfer with the S-SPLIT forcibly, set this bit to 1. However, for the normal split transaction, this module automatically clears the CSSTS bit to 0 upon completion of the C-SPLIT; therefore, clearing the CSSTS bit is not necessary. Controlling the CSSTS bit through this bit must be done while UACT is 0 and thus communication is halted or while no transfer is being performed with bus disconnection detected. Setting this bit to 1 while CSSTS is 0 has no effect. When the function controller mode is selected, be sure to write 0 to this bit.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
12	CSSTS	0	R	CSSTS Status Bit Indicates the C-SPLIT status of the split transaction when the host controller mode is selected. 0: START-SPLIT (S-SPLIT) transaction being processed or the transfer not using the split transaction in progress 1: C-SPLIT transaction being processed This module sets this bit to 1 upon start of the C-SPLIT and clears this bit to 0 upon detection of C-SPLIT completion. Indicates the valid value only when the host controller mode is selected.
11	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
10	ATREPM	0	R/W	Auto Response Mode Enables or disables auto response mode for the pertinent pipe. 0: Auto response disabled 1: Auto response enabled When the function controller mode is selected and the pertinent pipe is for bulk transfer, this bit can be set to 1. When this bit is set to 1, this module responds to the token from the USB host as described below. (1) When the pertinent pipe is for bulk IN transfer (TYPE = 01 and DIR = 1) When ATREPM = 1 and PID = BUF, this module transmits a zero-length packet in response to the IN token. This module updates (toggles) the sequence toggle bit (DATA-PID) each time this module receives the ACK from the USB host (in a single transaction, IN token is received, zero-length packet is transmitted, and then ACK is received.). In this case, this module does not generate the BRDY or BEMP interrupt. (2) When the pertinent pipe is for bulk OUT transfer (TYPE = 01 and DIR = 0) When ATREPM = 1 and PID = BUF, this module returns NAK in response to the OUT token (or PING token) and generates the NRDY interrupt. Modify this bit while CSSTS is 0 and PID is NAK. Before modifying this bit after modifying the PID bits for the corresponding pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked. For USB communication in auto response mode, set this bit to 1 while the FIFO buffer is empty. Do not write to the FIFO buffer during USB communication in auto response mode. When the pertinent pipe is for isochronous transfer, be sure to set this bit to 0. When the host controller mode is selected, set this bit to 0.
9	ACLRM	0	R/W	Auto Buffer Clear Mode Enables or disables automatic buffer clear mode for the pertinent pipe. 0: Disabled 1: Enabled (all buffers are initialized) To delete the contents in the FIFO buffer assigned to the pertinent pipe completely, write 1 and then 0 to this bit continuously. Table 28.13 shows the contents cleared by writing 1 and 0 to this bit continuously and the cases in which clearing the contents is necessary. Modify this bit while CSSTS is 0, PID is NAK, and the pertinent pipe is not selected by the CURPIPE bits. Before modifying this bit after modifying the PID bits for the corresponding pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
8	SQCLR	0	R/W*1	Toggle Bit Clear This bit should be set to 1 to clear the expected value (to set DATA0 as the expected value) of the sequence toggle bit for the next transaction of the pertinent pipe. 0: Writing invalid 1: Specifies DATA0. Setting this bit to 1 allows this module to set DATA0 as the expected value of the sequence toggle bit of the pertinent pipe. This bit always indicates 0. When the host controller mode is selected, setting this bit to 1 for the pipe for bulk OUT transfer, this module starts the next transfer of the pertinent pipe with the PING token. Set the SQCLR bit to 1 while CSSTS is 0 and PID is NAK. Before modifying this bit after modifying the PID bits for the corresponding pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked.
7	SQSET	0	R/W*1	Toggle Bit Set This bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the pertinent pipe. 0: Writing invalid 1: Specifies DATA1. Setting this bit to 1 allows this module to set DATA1 as the expected value of the sequence toggle bit of the pertinent pipe. This module always sets this bit to 0. Set the SQSET bit to 1 while CSSTS is 0 and PID is NAK. Before modifying this bit after modifying the PID bits for the corresponding pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked.
6	SQMON	0	R	Toggle Bit Confirmation Indicates the expected value of the sequence toggle bit for the next transaction of the pertinent pipe. 0: DATA0 1: DATA1 When the pertinent pipe is not for the isochronous transfer, this bit is toggled upon normal completion of the transaction. However, this bit is not toggled when a DATA-PID disagreement occurs during the receiving transfer.
5	PBUSY	0	R	Pipe Busy This bit indicates whether or not the pertinent pipe is being currently used for the transaction. 0: The pertinent pipe is not being currently used for the transaction. 1: The pertinent pipe is being currently used for the transaction. This module modifies this bit from 0 to 1 upon start of the USB transaction for the pertinent pipe, and modifies the bit from 1 to 0 upon completion of one transaction. Reading this bit after PID has been set to NAK allows checking that modification of the pipe settings is possible.
4 to 2	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PID[1:0]	00	R/W	Response PID Specifies the response type for the next transaction of the pertinent pipe. 00: NAK response 01: BUF response (depending on the buffer state) 10: STALL response 11: STALL response The default setting of these bits is NAK. Modify the setting to BUF to use the pertinent pipe for USB transfer. Table 28.14 and Table 28.15 show the basic operation (operation when there are no errors in the transmitted and received packets) of this module depending on the PID bit setting. Some registers require these bits to be set to NAK before their settings can be changed by software. Set these bits to NAK by software when changing such kind of registers. To confirm which registers apply to this, reference the descriptions of bits. After modifying the setting of these bits from BUF to NAK during USB communication using the pertinent pipe, check that PBUSY is 0 to see if USB communication using the pertinent pipe has actually entered the NAK state. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked. Even if the PID bits are modified to NAK after S-SPLIT of the split transaction has been issued for the pertinent pipe (while CSSTS indicates 1), this module continues the transaction until C-SPLIT completes. • This module modifies the setting of these bits as follows. This module sets PID to NAK on recognizing the completion of the transfer when the pertinent pipe is in the receiving direction and the SHTNAK bit for the selected pipe has been set to 1. • This module sets PID to STALL (11) on receiving the data packet with the payload exceeding the maximum packet size of the pertinent pipe. • This module sets PID to NAK on detecting a USB bus reset when the function controller mode is selected. • This module sets PID to NAK on detecting a reception error such as a CRC error three consecutive times when the host controller mode is selected. • This module sets PID to STALL (11) on receiving the STALL handshake when the host controller mode is selected. To specify each response type, set these bits as follows. • To make a transition from NAK (00) to STALL, set 10. • To make a transition from BUF (01) to STALL, set 11. • To make a transition from STALL (11) to NAK, set 10 and then 00. • To make a transition from STALL to BUF, set 00 (NAK) and then 01 (BUF).

Note 1. Only 0 can be read and 1 can be written to.
Note 2. Only 1 can be written to.

chanhvinh.com

Table 28.12 Meaning of BSTS Bit

DIR Bit	BFRE Bit	DCLRM Bit	Meaning of BSTS Bit
0	0	0	1: The received data can be read from the FIFO buffer. 0: The received data has been completely read from the FIFO buffer.
		1	Setting prohibited
	1	0	1: The received data can be read from the FIFO buffer. 0: The BCLR bit has been set to 1 after the received data has been completely read from the FIFO buffer.
		1	1: The received data can be read from the FIFO buffer. 0: The received data has been completely read from the FIFO buffer.
1	0	0	1: The transmit data can be written to the FIFO buffer. 0: The transmit data has been completely written to the FIFO buffer.
		1	Setting prohibited
	1	0	Setting prohibited
		1	Setting prohibited

Table 28.13 Contents Cleared by This Module by Setting ACLRM = 1

No.	Contents Cleared by ACLRM Bit Manipulation	Cases in which Clearing the Contents is Necessary
1	All the contents in the FIFO buffer assigned to the pertinent pipe (all the information in two FIFO buffer planes in double buffer mode)	
2	The interval count value when the pertinent pipe is for isochronous transfer	When the interval count value is to be reset
3	Values of the internal flags related to the BFRE bit	When the BFRE setting is modified
4	FIFO buffer toggle control	When the DBLB setting is modified
5	Values of the internal flags related to the transaction count	When the transaction count function is forcibly terminated

Table 28.14 Operation of This Module Depending on PID Setting (when Host Controller Mode is Selected)

PID	Transfer Type (TYPE Bits)	Transfer Direction (DIR Bit)	Operation of This Module
00 (NAK)	Operation does not depend on the setting.	Operation does not depend on the setting.	Does not issue tokens.
01 (BUF)	Bulk (TYPE = 01) or interrupt (TYPE = 10)	Operation does not depend on the setting.	Issues tokens while UACT is 1 and the FIFO buffer corresponding to the pertinent pipe is ready for transmission and reception. Does not issue tokens while UACT is 0 or the FIFO buffer corresponding to the pertinent pipe is not ready for transmission or reception.
	Isochronous (TYPE = 11)	Operation does not depend on the setting.	Issues tokens irrespective of the status of the FIFO buffer corresponding to the pertinent pipe when the UACT bit is set to 1. Does not issue tokens when the UACT bit is set to 0.
10 (STALL) or 11 (STALL)	Operation does not depend on the setting.	Operation does not depend on the setting.	Does not issue tokens.

chanhvinh.com

Table 28.15 Operation of This Module Depending on PID Setting (when Function Controller Mode is Selected)

PID	Transfer Type (TYPE Bits)	Transfer Direction (DIR Bit)	Operation of This Module
00 (NAK)	Bulk (TYPE = 01) or interrupt (TYPE = 10)	Operation does not depend on the setting.	Returns NAK in response to the token from the USB host.
	Isochronous (TYPE = 11)	Operation does not depend on the setting.	Returns nothing in response to the token from the USB host.
01 (BUF)	Bulk (TYPE = 01)	Receiving direction (DIR = 0)	Receives data and returns ACK or NYET in response to the OUT token from the USB host if the FIFO buffer corresponding to the pertinent pipe is ready for reception. Returns NAK if not ready. Returns ACK in response to the PING token from the USB host if the FIFO buffer corresponding to the pertinent pipe is ready for reception. Returns NAK if not ready.
	Interrupt (TYPE = 10)	Receiving direction (DIR = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the pertinent pipe is ready for reception. Returns NAK if not ready.
	Bulk (TYPE = 01) or interrupt (TYPE = 10)	Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Returns NAK if not ready.
	Isochronous (TYPE = 11)	Receiving direction (DIR = 0)	Receives data in response to the OUT token from the USB host if the FIFO buffer corresponding to the pertinent pipe is ready for reception. Discards data if not ready.
		Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Transmits the zero-length packet if not ready.
10 (STALL) or 11 (STALL)	Bulk (TYPE = 01) or interrupt (TYPE = 10)	Operation does not depend on the setting.	Returns STALL in response to the token from the USB host.
	Isochronous (TYPE = 11)	Operation does not depend on the setting	Returns nothing in response to the token from the USB host.

chanhvinh.com

(2) PIPEnCTR (n = 6 to 8)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BSTS	—	CSCLR	CSSTS	—	—	ACLRM	SQCLR	SQSET	SQ MON	PBUSY	—	—	—	PID[1:0]	
Initial value:	0	—	0	0	—	—	0	0	0	0	0	—	—	—	0	0
R/W:	R	R	R/W*1	R/W	R	R	R/W	R/W*1	R/W*1	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	BSTS	0	R	Buffer Status Indicates whether or not the FIFO buffer allocated for the pertinent pipe can be accessed by the CPU. 0: Buffer access is disabled. 1: Buffer access is enabled. The meaning of this bit depends on the settings of the DIR, BFRE, and DCLRM bits as shown in Table 28.12.
14	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
13	CSCLR	0	R/W*1	C-SPLIT Status Clear Bit Setting this bit to 1 allows this module to clear the CSSTS bit of the pertinent pipe to 0. 0: Writing invalid 1: Clears the CSSTS bit to 0. For the transfer using the split transaction, to restart the next transfer with the S-SPLIT forcibly, set this bit to 1. However, for the normal split transaction, this module automatically clears the CSSTS bit to 0 upon completion of the C-SPLIT; therefore, clearing the CSSTS bit is not necessary. Controlling the CSSTS bit through this bit must be done while UACT is 0 thus communication is halted or while no transfer is being performed with bus disconnection detected. Setting this bit to 1 while CSSTS is 0 has no effect. When the function controller mode is selected, be sure to write 0 to this bit.
12	CSSTS	0	R/W	CSSTS Status Bit Indicates the C-SPLIT status of the split transaction when the host controller mode is selected. 0: START-SPLIT (S-SPLIT) transaction being processed or the transfer not using the split transaction in progress 1: C-SPLIT transaction being processed This module sets this bit to 1 upon start of the C-SPLIT and clears this bit to 0 upon detection of C-SPLIT completion. Indicates the valid value only when the host controller mode is selected.
11, 10	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
9	ACLRM	0	R/W	Auto Buffer Clear Mode Enables or disables automatic buffer clear mode for the pertinent pipe. 0: Disabled 1: Enabled (all buffers are initialized) To delete the contents in the FIFO buffer assigned to the pertinent pipe completely, write 1 and then 0 to this bit continuously. Table 28.16 shows the contents cleared by writing 1 and 0 to this bit continuously and the cases in which clearing the contents is necessary. Modify this bit while CSSTS is 0, PID is NAK, and the pertinent pipe is not selected by the CURPIPE bits. Before modifying this bit after modifying the PID bits for the corresponding pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
8	SQCLR	0	R/W*1	Toggle Bit Clear This bit should be set to 1 to clear the expected value (to set DATA0 as the expected value) of the sequence toggle bit for the next transaction of the pertinent pipe. 0: Writing invalid 1: Specifies DATA0. Setting this bit to 1 allows this module to set DATA0 as the expected value of the sequence toggle bit of the pertinent pipe. This bit always indicates 0. When the host controller mode is selected, setting this bit to 1 for the pipe for bulk OUT transfer, this module starts the next transfer of the pertinent pipe with the PING token. Set the SQCLR bit to 1 while CSSTS is 0 and PID is NAK. Before modifying this bit after modifying the PID bits for the corresponding pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked.
7	SQSET	0	R/W*1	Toggle Bit Set This bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the pertinent pipe. 0: Writing invalid 1: Specifies DATA1. Setting this bit to 1 allows this module to set DATA1 as the expected value of the sequence toggle bit of the pertinent pipe. This module always sets this bit to 0. Set the SQSET bit to 1 while CSSTS is 0 and PID is NAK. Before modifying this bit after modifying the PID bits for the corresponding pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked.
6	SQMON	0	R	Toggle Bit Confirmation Indicates the expected value of the sequence toggle bit for the next transaction of the pertinent pipe. 0: DATA0 1: DATA1 When the pertinent pipe is not for the isochronous transfer, this bit is toggled upon normal completion of the transaction. However, this bit is not toggled when a DATA-PID disagreement occurs during the receiving transfer.
5	PBUSY	0	R	Pipe Busy This bit indicates whether or not the pertinent pipe is being currently used for the USB bus. 0: The pertinent pipe is not being currently used for the USB bus. 1: The pertinent pipe is being currently used for the USB bus. This module modifies this bit from 0 to 1 upon start of the USB transaction for the pertinent pipe, and modifies the bit from 1 to 0 upon completion of one transaction. Reading this bit after PID has been set to NAK allows checking that modification of the pipe settings is possible.
4 to 2	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PID[1:0]	00	R/W	Response PID Specifies the response type for the next transaction of the pertinent pipe. 00: NAK response 01: BUF response (depending on the buffer state) 10: STALL response 11: STALL response The default setting of these bits is NAK. Modify the setting to BUF to use the pertinent pipe for USB transfer. Table 28.14 and Table 28.15 show the basic operation (operation when there are no errors in the transmitted and received packets) of this module depending on the PID bit setting. Some registers require these bits to be set to NAK before their settings can be changed by software. Set these bits to NAK by software when changing such kind of registers. To confirm which registers apply to this, reference the descriptions of bits. After modifying the setting of these bits from BUF to NAK during USB communication using the pertinent pipe, check that PBUSY is 0 to see if USB communication using the pertinent pipe has actually entered the NAK state. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked. Even if the PID bits are modified to NAK after S-SPLIT of the split transaction has been issued for the pertinent pipe (while CSSTS indicates 1), this module continues the transaction until C-SPLIT completes. This module modifies the setting of these bits as follows. • This module sets PID to NAK on recognizing the completion of the transfer when the pertinent pipe is in the receiving direction and the SHTNAK bit for the selected pipe has been set to 1. • This module sets PID to STALL (11) on receiving the data packet with the payload exceeding the maximum packet size of the pertinent pipe. • This module sets PID to NAK on detecting a USB bus reset when the function controller mode is selected. • This module sets PID to NAK on detecting a reception error such as a CRC error three consecutive times when the host controller mode is selected. • This module sets PID to STALL (11) on receiving the STALL handshake when the host controller mode is selected. To specify each response type, set these bits as follows. • To make a transition from NAK (00) to STALL, set 10. • To make a transition from BUF (01) to STALL, set 11. • To make a transition from STALL (11) to NAK, set 10 and then 00. • To make a transition from STALL to BUF, set 00 (NAK) and then 01 (BUF).

Note 1. Only 0 can be read and 1 can be written to.

Table 28.16 Contents Cleared by This Module by Setting ACLRM = 1

No.	Contents Cleared by ACLRM Bit Manipulation	Cases in which Clearing the Contents is Necessary
1	All the contents in the FIFO buffer assigned to the selected pipe	
2	When the host controller mode is selected, the interval count value when the selected pipe is for interrupt transfer	When the interval count value is to be reset
3	Values of the internal flags related to the BFRE bit	When the BFRE setting is modified
4	Values of the internal flags related to the transaction count	When the transaction count function is forcibly terminated

chanhvinh.com

28.3.37 PIPEn Transaction Counter Enable Registers (PIPEnTRE) (n = 1 to 5, 9, and A to F)

PIPEnTRE is a register that enables or disables the transaction counter corresponding to PIPE1 to PIPE5, and clears the transaction counter.

These registers are initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TRENB	TRCLR	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	0	0	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R/W	R/W*	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
9	TRENB	0	R/W	Transaction Counter Enable Enables or disables the transaction counter. 0: The transaction counter is disabled. 1: The transaction counter is enabled. For the pipe in the receiving direction, setting this bit to 1 after setting the total number of the packets to be received in the TRNCNT bits allows this module to control hardware as described below on having received the number of packets equal to the set value in the TRNCNT bits. • In continuous transmission/reception mode (CNTMD = 1), this module switches the FIFO buffer to the CPU side even if the FIFO buffer is not full on completion of reception. • While SHTNAK is 1, this module modifies the PID bits to NAK for the corresponding pipe on having received the number of packets equal to the set value in the TRNCNT bits. • While BFRE is 1, this module asserts the BRDY interrupt on having received the number of packets equal to the set value in the TRNCNT bits and then reading out the last received data. For the pipe in the transmitting direction, set this bit to 0. When the transaction counter is not used, set this bit to 0. When the transaction counter is used, set the TRNCNT bits before setting this bit to 1. Set this bit to 1 before receiving the first packet to be counted by the transaction counter.
8	TRCLR	0	R/W*	Transaction Counter Clear When this bit is set to 1, this module clears the current counter value of the transaction counter corresponding to the pertinent pipe and then sets this bit to 0. 0: Invalid 1: The current counter value is cleared.
7 to 0	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.

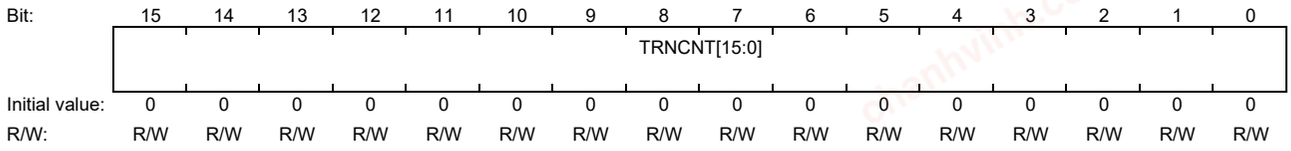
Note 1. Modify each bit in this register while CSSTS is 0 and PID is NAK. Before modifying each bit after modifying the PID bits for the corresponding pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked.

* Only 0 can be read and 1 can be written to.

chanhvinh.com

28.3.38 PIPEn Transaction Counter Registers (PIPEnTRN) (n = 1 to 5, 9, and A to F)

PIPEnTRN is a transaction counter corresponding to PIPE1 to PIPE5, 9, and A to F.
These registers are initialized by a power-on reset, but retain the set value by a USB bus reset.



Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TRNCNT[15:0]	H'0000	R/W	Transaction Counter When written to: Specifies the total number of packets to be received by the pertinent pipe (number of transactions). When read from: Indicates the specified number of transactions if TRENB is 0. Indicates the number of currently counted transaction if TRENB is 1. This module increments the value of these bits by one when all of the following conditions are satisfied on receiving the packet. • TRENB is 1. • (TRNCNT setting ≠ current counter value + 1) on receiving the packet. • The payload of the received packet agrees with the set value in the MXPS bits. This module clears the value of these bits to 0 when any of the following conditions are satisfied. • All the following conditions are satisfied. TRENB is 1. (TRNCNT setting = current counter value + 1) on receiving the packet. The payload of the received packet agrees with the set value in the MXPS bits. • All the following conditions are satisfied. TRENB is 1. This module has received a short packet. • The following condition is satisfied. TRCLR is 1. For the pipe in the transmitting direction, set these bits to 0. When the transaction counter is not used, set these bits to 0. Modify these bits while CSSTS is 0, PID is NAK, and TRENB is 0. Before modifying these bits after modifying the PID bits for the corresponding pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, the PBUSY bit does not have to be checked. To modify the value of these bits, set TRCLR to 1 before setting TRENB to 1.

chanhvinh.com

28.3.39 Device Address n Configuration Registers (DEVADDn) (n = 0 to 9, and A)

DEVADDn is a register that specifies the USB address and port number of the hub to which the communication target peripheral device is connected and also specifies the USB transfer speed of the communication target peripheral device. These registers are initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	UPPHUB[3:0]				HUBPORT[2:0]			USBSPD[1:0]		—	—	—	—	—	—
Initial value:	—	0	0	0	0	0	0	0	0	0	—	—	—	—	—	—
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
14 to 11	UPPHUB[3:0]	0000	R/W	Address of Hub to which Communication Target is Connected Specifies the USB address of the hub to which the communication target peripheral device is connected. 0000: The peripheral device is directly connected to the port of this module. 0001 to 1010: USB address of the hub 1011 to 1111: Setting prohibited
10 to 8	HUBPORT[2:0]	000	R/W	Port Number of Hub to which Communication Target is Connected Specifies the port number of the hub to which the communication target peripheral device is connected. 000: The peripheral device is directly connected to the port of this module. 001 to 111: Port number of the hub
7, 6	USBSPD[1:0]	00	R/W	Transfer Speed of the Communication Target Device Specifies the USB transfer speed of the communication target peripheral device. 00: DEVADDn register is not used. 01: Low speed 10: Full speed 11: High speed
5 to 0	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.

Note 1. When the host controller mode is selected, the bits in this register should be set before starting communication using each pipe.
(1) When the host controller mode is selected, this module refers to the settings of the UPPHUB bits and HUBPORT bits to generate packets for split transactions.
(2) When the host controller mode is selected, this module refers to the setting of the USBSPD bits to generate packets.
Note 2. The bits in this register should be modified while no valid pipes are using the settings of this register. Valid pipes refer to the ones satisfying both of conditions (1) and (2) below.
(1) This register is selected by the DEVSEL bits as the communication target.
(2) The PID bits are set to BUF for the pertinent pipe or the pertinent pipe is the DCP with SUREQ being 1.
Note 3. When the function controller mode is selected, set all the bits in this register to 0.

chanhvinh.com

28.3.40 UTMI Suspend Mode Register (SUSPMODE)

SUSPMODE is a register that specifies the SuspendM signal to be sent to the UTMI.
This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	SUSPM	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	0	—	0	—	—	—	0	—	—	—	—	0	—	0	0
R/W:	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
14	SUSPM	0	R/W	SuspendM Control Enables or disables the clock to be supplied to this module. 0: The clock supplied to this module is disabled. 1: The clock supplied to this module is enabled. This module controls clock output by using the SuspendM signal. The clock to the LINK is stopped, when the SuspendM signal is low. Writing to the registers of this module is impossible when the SUSPM bit is set to 0 (the clock to this module is stopped). Reading the registers is possible. Note that writing to the following registers is possible even when the SUSPM bit is set to 0. - SYSCFG0 - BUSWAIT - INTENB1* - SUSPMODE Note: * Writing to bit 0 in INTENB1 is only possible when the SUSPM bit is set to 0. When setting this bit to 1, wait for at least 1 ms after setting the UPLLE bit to 1. Set this bit to 0 when this module is to enter software standby or USB standby mode.
13	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
12	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
11 to 9	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
8	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
7 to 4	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

28.3.41 D0FIFO Continuous Transfer Port Register n (D0FIFOBn)
D1FIFO Continuous Transfer Port Register n (D1FIFOBn) (n = 0 to 7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FIFOPORT[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FIFOPORT[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	FIFOPORT [31:0]	H'0000 0000	R/W	FIFO Port Accessing these bits allows reading the received data from the FIFO buffer or writing the transmit data to the FIFO buffer. When the DFACC bits are set to 01 (16-byte continuous access mode) or 10 (32-byte continuous access mode), use the DnFIFO continuous transfer port registers to access the DnFIFO buffer.

chanhvinh.com

28.4 Operation

28.4.1 System Control and Oscillation Control

This section describes the register operations that are necessary to the initial settings of this module, and the registers necessary for power consumption control.

(1) Resets

Table 28.17 lists the types of resets for this module. For the initialized states of the registers following the reset operations, see section 28.3, Register Descriptions.

Table 28.17 Types of Reset

Name	Operation
Power-on reset	Low level input from the $\overline{\text{RES}}$ pin
USB bus reset	Automatically detected by this module from the D+ and D– lines when the function controller mode is selected

(2) Controller Mode Selection

This module can select the host controller mode or function controller mode using the DCFM bit in SYSCFG0. Changing the DCFM bit should be done in the initial settings immediately after a power-on reset or in the D+ pull-up disabled (DPRPU = 0) and D+ /D– pull-down disabled (DRPD = 0) state.

(3) USB Data Bus Resistor Control

This module controls switching between a pull-up resistor for the D+ signal and a pull-down resistor for the D+ and D- signals for the Renesas USB 2.0 PHY port. The DPRPU and DRPD bits of the SYSCFG0 register are used to make the pull-up and pull-down resistor settings.
When the function controller mode is selected, set the DPRPU bit of the SYSCFG0 register to 1, which pulls up the D+ signal, after connection to a USB host is recognized.
When disconnection from the USB host is recognized, set the DPRPU and DCFM bits according to the procedure below.

1. Set the DPRPU bit to 0.
2. Wait for at least 1 µs (1000 ns).
3. Set the DCFM bit to 1.
4. Wait for at least 200 ns.
5. Set the DCFM bit to 0.

This module incorporates the terminating resistor for the D+ and D- signals during high-speed operation and the output resistor for the signals during full-speed operation. This module automatically switches the resistor after connection with the host controller or peripheral device upon the detection of a reset handshake, suspend, and resume event.
When the function controller mode is selected and the DPRPU bit in SYSCFG0 is cleared to 0 during communication with the host controller, the pull-up resistor (or the terminating resistor) of the USB data line is disabled, making it possible to notify the USB host of the device disconnection.

(4) Selecting the Input Clock

This module can select the signal on USB_X1 or EXTAL as an input clock. The UCKSEL bit of the SYSCFG0 register for channel 0 is used to select the input clock. The UCKSEL bit should be set while supply of the clock signal to the USB module is stopped (SUSPM = 0 for channel 0 and channel 1).

chanhvinh.com

(5) Setting the Clock Supply for the USB Module

Set the UCKSEL bit of the SYSCFG0 register to the target input clock and then set clock supply by following the appropriate procedure below.

Example 1: When the clock supply is enabled by the initial setting immediately after a power-on reset,

1. set the UPLLE bit to 1,
2. wait for 1 ms, and
3. set the SUSPM bit to 1.

Example 2: When the clock supply is stopped in the suspended state,

1. set the SUSPM bit to 0 and
2. set the UPLLE bit to 0.

Example 3: When the clock supply is enabled after recovery from the suspended state,

1. set the UPLLE bit to 1,
2. wait for 1 ms, and
3. set the SUSPM bit to 1.

Note: • During high-speed operation when the function controller mode is selected, set the SUSPM bit to 1 within 2.5 ms of this module resuming operation from the suspended state due to a USB reset.

chanhvinh.com

28.4.2 Interrupt Functions

(1) Overview of Interrupt Functions

Table 28.18 lists the interrupt generation conditions for this module.

Table 28.18 Interrupt Generation Conditions

Bit	Interrupt Name	Cause of Interrupt	Mode That Generates the Interrupt	Related Status
VBINT	VBUS interrupt	• When a change in the state of the VBUS input pin has been detected (low to high or high to low)	Host, function	VBSTS
RESM	Resume interrupt	• When a change in the state of the USB bus has been detected in the suspended state (J-state to K-state or J-state to SE0)	Function	—
SOFR	Frame number update interrupt	When the host controller mode is selected: • When an SOF packet with a different frame number has been transmitted When the function controller mode is selected: • If SOFRM = 0, reception of an SOF packet with a different frame number • If SOFRM = 1, reception of an SOF packet with the microframe number 0 was not possible because it was corrupted.	Host, function	—
DVST	Device state transition interrupt	• When a device state transition is detected A USB bus reset detected The suspended state detected SET_ADDRESS request received SET_CONFIGURATION request received	Function	DVSQ
CTRT	Control transfer stage transition interrupt	• When a stage transition is detected in control transfer Setup stage completed Control write transfer status stage transition Control read transfer status stage transition Control transfer completed A control transfer sequence error occurred	Function	CTSQ
BEMP	Buffer empty interrupt	• When transmission of all of the data in the buffer memory has been completed and the buffer has become empty • When an excessive maximum packet size error has been detected	Host, function	PIPEBEMP
NRDY	Buffer not ready interrupt	When the host controller mode is selected: • When STALL is received from the peripheral side for the issued token • When a response cannot be received correctly from the peripheral side for the issued token (No response is returned three consecutive times or a packet reception error occurred three consecutive times.) • When an overrun/underrun occurred during isochronous transfer When the function controller mode is selected: • Reception of a token while PID = BUF and the buffer memory is not ready for transmission or reception • When a CRC error or a bit stuffing error occurred during data reception in isochronous transfer • When an interval error occurred during data reception in isochronous transfer	Host, function	PIPENRDY
BRDY	Buffer ready interrupt	• When the buffer is ready (reading or writing is enabled)	Host, function	PIPEBRDY
BCHG	Bus change interrupt	• When a change of USB bus state is detected	Host	—
DTCH	Device disconnection detection	• When disconnection of a peripheral device connected to this LSI's USB port is detected.	Host	—
ATTCH	Device connection detection	• When J-state or K-state is detected on the USB port for 2.5 µs. Used for checking whether a peripheral device is connected.	Host	—

chanhvinh.com

Table 28.18 Interrupt Generation Conditions

Bit	Interrupt Name	Cause of Interrupt	Mode That Generates the Interrupt	Related Status
EOFERR	EOF error detection	• When EOF error of a peripheral device is detected	Host	—
SACK	Normal setup operation	• When the normal response (ACK) for the setup transaction is received	Host	—
SIGN	Setup error	• When a setup transaction error (no response or ACK packet corruption) is detected three consecutive times.	Host	—

chanhvinh.com

Figure 28.1 shows a diagram relating to interrupts of this module.

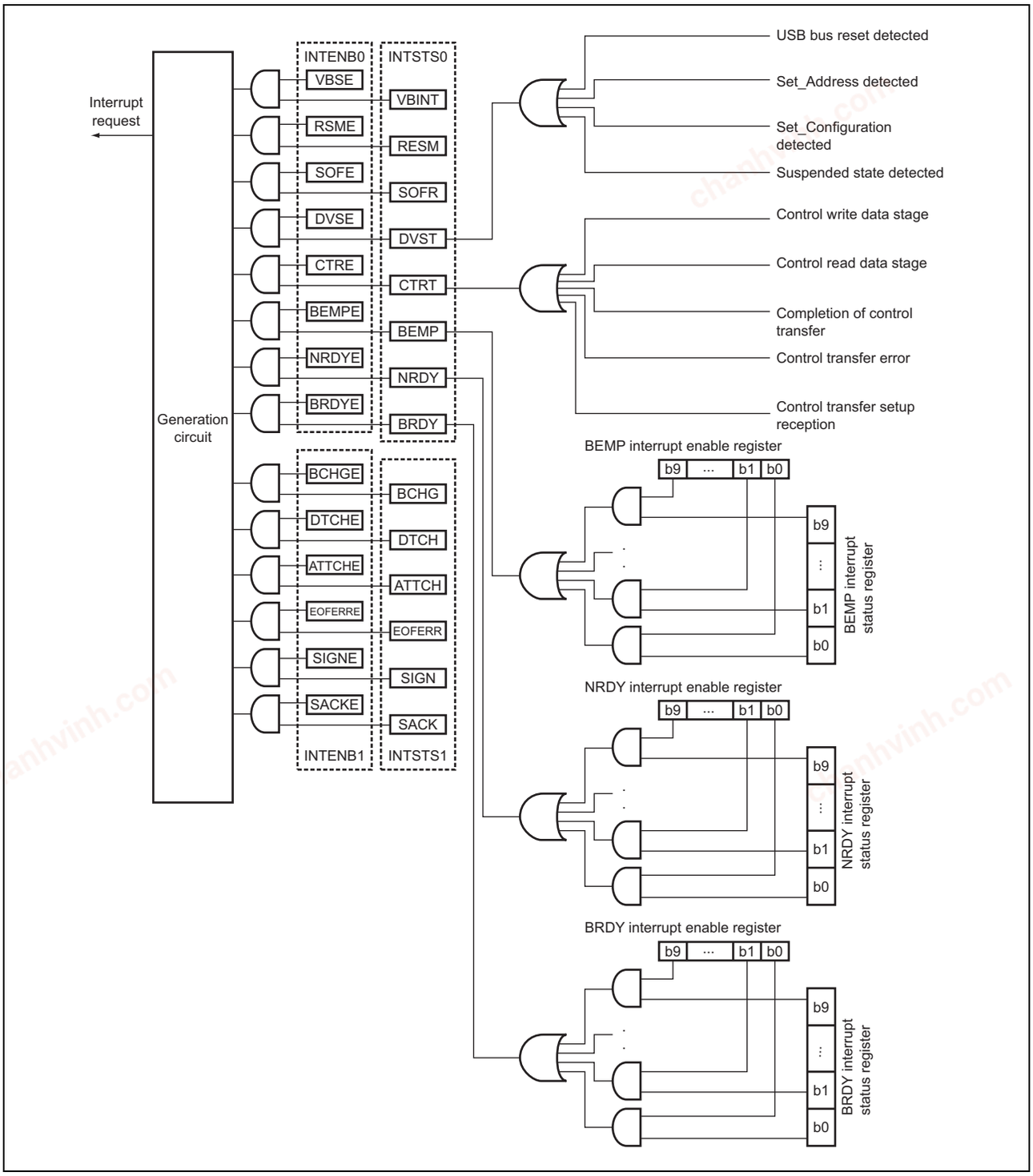


Figure 28.1 Items Relating to Interrupts

chanhvinh.com

(2) BRDY Interrupt

The BRDY interrupt is generated when either of the host controller mode or function controller mode is selected. When each pipe satisfies the following conditions, this module sets 1 to a corresponding bit in BRDYSTS. Under this condition, if the PIPEBRDYE bit in BRDYENB that corresponds to the pipe is set to 1 and the BRDYE bit in INTENB0 is set to 1, this module sets the BRDY bit in INTSTS0 to 1, allowing the BRDY interrupt to be generated.
The conditions for generating and clearing the BRDY interrupt depend on the settings of the BRDYM bit and BFRE bit for the pertinent pipe as described below.

(a) When BRDYM Bit is 0 and BFRE Bit is 0

With these settings, the BRDY interrupt indicates that the FIFO port is accessible.
On any of the following conditions, this module generates the internal BRDY interrupt request trigger and sets 1 to the PIPEBRDY bit corresponding to the pertinent pipe.

1. For the pipe in the transmitting direction
   - When the DIR bit is changed from 0 to 1.
   - When packet transmission is completed using the pertinent pipe when write-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS bit is read as 0).
     In continuous transmission/reception mode, the request trigger is generated on completion of transmitting data of one plane of the FIFO buffer.
   - When one FIFO buffer is empty on completion of writing data to the other FIFO buffer in double buffer mode.
     The request trigger is not generated until completion of writing data to the currently-written FIFO buffer plane even if transmission to the other FIFO buffer is completed.
   - When the hardware flushes the buffer of the pipe for isochronous transfers.
   - When 1 is written to the ACLRM bit, which causes the FIFO buffer to make transition from the write-disabled to write-enabled state.

The request trigger is not generated for the DCP (that is, during data transmission for control transfers).

2. For the pipe in the receiving direction
   - When packet reception is completed successfully thus enabling the FIFO buffer to be read when read-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS bit is read as 0).
     The request trigger is not generated for the transaction in which DATA-PID disagreement occurs.
     In continuous transmission/reception mode, the request trigger is not generated when the data is of the specified maximum packet size and the buffer has available space.
     When a short packet is received, the request trigger is generated even if the FIFO buffer has available space.
   - When the transaction counter is used, the request trigger is generated on receiving the specified number of packets.
     In this case, the request trigger is generated even if the FIFO buffer has available space.
     When one FIFO buffer is read-enabled on completion of reading data from the other FIFO buffer in double buffer mode.
     The request trigger is not generated until completion of reading data from the currently-read FIFO buffer plane even if reception by the other FIFO buffer is completed.

When the function controller mode is selected, the BRDY interrupt is not generated in the status stage of control transfers.
The PIPEBRDY interrupt status of the pertinent pipe can be cleared to 0 by writing 0 to the corresponding PIPEBRDY interrupt status bit in the BRDYSTS register. In this case, 1s should be written to the PIPEBRDY interrupt status bits for the other pipes.
Be sure to clear the BRDY status before accessing the FIFO buffer.

chanhvinh.com

(b) When BRDYM Bit is 0 and BFRE Bit is 1

With these settings, this module generates the BRDY interrupt on completion of reading all the data for a single transfer using the pipe in the receiving direction, and sets 1 to the PIPEBRDY bit corresponding to the pertinent pipe.
On any of the following conditions, this module determines that the last data for a single transfer has been received.

- When a short packet including a zero-length packet is received.
- When the transaction counter register (TRNCNT bits) is used and the number of packets specified by the TRNCNT bits is completely received.

When the pertinent data is completely read out after any of the above determination conditions has been satisfied, this module determines that all the data for a single transfer has been completely read out.
When a zero-length packet is received when the FIFO buffer is empty, this module determines that all the data for a single transfer has been completely read out upon the FRDY and DTLN bits of the FIFO port control register being set to 1 and 0, respectively. In this case, to start the next transfer, write 1 to the BCLR bit in the corresponding FIFOCTR register.
With these settings, this module does not detect the BRDY interrupt for the pipe in the transmitting direction.
The PIPEBRDY interrupt status of the pertinent pipe can be cleared to 0 by writing 0 to the corresponding PIPEBRDY interrupt status bit. In this case, 1s should be written to the PIPEBRDY interrupt status bits for the other pipes.
In this mode, the BFRE bit setting should not be modified until all the data for a single transfer has been processed. When it is necessary to modify the BFRE bit before completion of processing, all the FIFO buffers for the pertinent pipe should be cleared using the ACLRM bit.

(c) When the BRDYM bit is 1 and the BFRE bit is 0

With these settings, the PIPEBRDY values are linked to the BSTS bit settings for each pipe. In other words, the BRDY interrupt status bits (PIPEBRDY) are set to 1 or 0 by this module depending on the FIFO buffer status.

1. For the pipe in the transmitting direction
   The BRDY interrupt status bits are set to 1 when the FIFO buffer is write-enabled and are set to 0 when write-disabled.
   However, the BRDY interrupt is not generated if the DCP in the transmitting direction is write-enabled.
2. For the pipe in the receiving direction
   The BRDY interrupt status bits are set to 1 when the FIFO buffer is read-enabled and are set to 0 when all the data have been read (read-disabled).
   When a zero-length packet is received when the FIFO buffer is empty, the pertinent bit is set to 1 and the BRDY interrupt continues to be effective until BCLR = 1 is written.

With this setting, the PIPEBRDY bit cannot be cleared to 0. When BRDYM is set to 1, all of the BFRE bits (for all pipes) should be cleared to 0.

chanhvinh.com

(3) NRDY Interrupt

On generating the internal NRDY interrupt request for the pipe whose PID bits are set to BUF, this module sets the corresponding PIPENRDY bit in NRDYSTS to 1. If the corresponding bit in NRDYENB is set to 1, this module sets the NRDY bit in INTSTS0 to 1, allowing the NRDY interrupt to be generated.
The following describes the conditions on which this module generates the internal NRDY interrupt request for each pipe.
However, the internal NRDY interrupt request is not generated during setup transaction execution when the host controller mode is selected. During setup transactions when the host controller mode is selected, the SACK or SIGN interrupt is detected.
The internal NRDY interrupt request is not generated during status stage execution of the control transfer when the function controller mode is selected.

(a) If Host Controller Mode is Selected when Connection is Used in which No Split Transactions Occur

1. For the pipe in the transmitting direction
   On any of the following conditions, this module detects the NRDY interrupt.
   - For the pipe for isochronous transfers, when the time to issue an OUT token comes in a state in which there is no data to be transmitted in the FIFO buffer.
     In this case, this module transmits a zero-length packet following the OUT token, setting the corresponding PIPENRDY bit and the OVRN bit to 1.
   - During communications other than setup transactions using the pipe for the transfers other than isochronous transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device (when timeout is detected before detection of the handshake packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.
     In this case, this module sets the corresponding PIPENRDY bit to 1 and modifies the setting of the PID bits of the corresponding pipe to NAK.
   - During communications other than setup transactions, when the STALL handshake is received from the peripheral device (including the STALL handshake in response to PING in addition to the STALL handshake in response to OUT).
     In this case, this module sets the corresponding PIPENRDY bit to 1 and modifies the setting of the PID bits of the corresponding pipe to STALL (11).

2. For the pipe in the receiving direction
   - For the pipe for isochronous transfers, when the time to issue an IN token comes in a state in which there is no space available in the FIFO buffer.
     In this case, this module discards the received data for the IN token, setting the PIPENRDY bit of the corresponding pipe and the OVRN bit to 1.
     When a packet error is detected in the received data for the IN token, this module also sets the CRCE bit to 1.
   - For the pipe for the transfers other than isochronous transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device for the IN token issued by this module (when timeout is detected before detection of the DATA packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.
     In this case, this module sets the corresponding PIPENRDY bit to 1 and modifies the setting of the PID bits of the corresponding pipe to NAK.
   - For the pipe for isochronous transfers, when no response is returned from the peripheral device for the IN token (when timeout is detected before detection of the DATA packet from the peripheral device) or an error is detected in the packet from the peripheral device.
     In this case, this module sets the corresponding PIPENRDY bit to 1. (The setting of the PID bits of the corresponding pipe is not modified.)

chanhvinh.com

- For the pipe for isochronous transfers, when a CRC error or a bit stuffing error is detected in the received data packet.
  In this case, this module sets the corresponding PIPENRDY bit and CRCE bit to 1.
- When the STALL handshake is received.
  In this case, this module sets the corresponding PIPENRDY bit to 1 and modifies the setting of the PID bits of the corresponding pipe to STALL.

(b) If Host Controller Mode is Selected when Connection is Used in which Split Transactions Occur

1. For the pipe in the transmitting direction
   - For the pipe for isochronous transfers, when the time to issue an OUT token comes in a state in which there is no data to be transmitted in the FIFO buffer.
     In this case, this module transmits a zero-length packet following the OUT token, setting the corresponding PIPENRDY bit and the OVRN bit to 1 at the issuance of the start-split transaction (S-SPLIT).
   - For the pipe for the transfers other than isochronous transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the HUB for the S-SPLIT or complete-split transaction (C-SPLIT) (when timeout is detected before detection of the handshake packet from the HUB) and 2) an error is detected in the packet from the HUB.
     In this case, this module sets the PIPENRDY bit of the corresponding pipe to 1 and modifies the setting of the PID bits of the corresponding pipe to NAK.
     If the NRDY interrupt is detected when the C-SPLIT is issued, this module clears the CSSTS bit to 0.
   - When the STALL handshake is received in response to the C-SPLIT.
     In this case, this module sets the corresponding PIPENRDY bit to 1, modifies the setting of the PID bits of the corresponding pipe to STALL (11) and clears the CSSTS bit to 0.
     This interrupt is not detected for setup transactions.
   - For the pipe for interrupt transfers when the NYET is received in response to the C-SPLIT and the microframe number = 4.
     In this case, this module sets the corresponding PIPENRDY bit to 1 and clears the CSSTS bit to 0 (does not modify the setting of the PID bits for the corresponding pipe).

2. For the pipe in the receiving direction
   - For the pipe for isochronous transfers, when the time to issue an IN token comes in a state in which there is no space available in the FIFO buffer.
     In this case, this module discards the received data for the IN token, setting the corresponding PIPENRDY bit and the OVRN bit to 1 at the issuance of the S-SPLIT.
   - For the pipes for bulk transfers, or the transfers other than setup transactions with the DCP, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the HUB for the IN token issued by this module at the issuance of S-SPLIT or C-SPLIT (when timeout is detected before detection of the DATA packet from the HUB) and 2) an error is detected in the packet from the HUB.
     In this case, this module sets the corresponding PIPENRDY bit to 1 and modifies the setting of the PID bits of the corresponding pipe to NAK. When the condition is generated during the C-SPLIT transaction, this module clears the CSSTS bit to 0.
   - During the C-SPLIT transaction for the pipe for isochronous transfers or interrupt transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the HUB for the IN token issued by this module (when timeout is detected before detection of the DATA packet from the HUB) and 2) an error is detected in the packet from the HUB.
     On generating this condition for the pipe for interrupt transfers, this module sets the corresponding PIPENRDY bit to 1, modifies the setting of the PID bits of the corresponding pipe to NAK and clears the CSSTS bit to 0.
     On generating this condition for the pipe for isochronous transfers, this module sets the corresponding PIPENRDY

chanhvinh.com

bit to 1 and CRCE bit to 1, and clears the CSSTS bit to 0 (does not modify the setting of the PID bits for the corresponding pipe).

- During the C-SPLIT transaction, when the STALL handshake is received for the pipe for the transfers other than isochronous transfers.
  In this case, this module sets the corresponding PIPENRDY bit to 1, modifies the setting of the PID bits of the corresponding pipe to STALL (11) and clears the CSSTS bit to 0.
- During the C-SPLIT transaction, when the NYET handshake is received for the pipe for the isochronous transfers or interrupt transfers and the microframe number = 4.
  In this case, this module sets the corresponding PIPENRDY bit for the pipe to 1, sets the CRCE bit to 1, and clears the CSSTS bit to 0 (does not modify the setting of the PID bits for the corresponding pipe).

(c) When Function Controller Mode is Selected

1. For the pipe in the transmitting direction
   - On receiving an IN token when the PID bits are set to 01 (BUF) for the pertinent pipe and there is no data to be transmitted in the FIFO buffer.
     In this case, this module generates an NRDY interrupt request at the reception of the IN token, setting the PIPENRDY bit to 1. For the pipe for the isochronous transfers in which an interrupt is generated, this module transmits a zero-length packet, setting the OVRN bit to 1.

2. For the pipe in the receiving direction
   - On receiving an OUT token when the PID bits are set to 01 (BUF) for the pertinent pipe and there is no space available in the FIFO buffer.
     For the pipe for the isochronous transfers in which an interrupt is generated, this module generates an NRDY interrupt request at the reception of the OUT token, setting the PIPENRDY bit to 1 and OVRN bit to 1.
     For the pipe for the transfers other than isochronous transfers in which an interrupt is generated, this module generates an NRDY interrupt request when an NAK handshake is transferred after the data following the OUT token was received, setting the PIPENRDY bit to 1.
     However, during re-transmission (due to DATA-PID disagreement), the NRDY interrupt request is not generated.
     In addition, if an error occurs in the DATA packet, the NRDY interrupt request is not generated.
   - On receiving a PING token when the PID bits are set to 01 (BUF) for the pertinent pipe and there is no space available in the FIFO buffer.
     In this case, this module generates an NRDY interrupt request at the reception of the PING token, setting the PIPENRDY bit to 1.
   - For the pipe for isochronous transfers, when the PID bits are set to 01 (BUF) for the pertinent pipe and a token is not received normally within an interval frame.
     In this case, this module generates an NRDY interrupt request at the reception of an SOF, and sets the PIPENRDY bit to 1.

chanhvinh.com

(4) BEMP Interrupt

On detecting the BEMP interrupt for the pipe whose PID bits are set to BUF, this module sets the corresponding PIPEBEMP bit in BEMPSTS to 1. If the corresponding bit in BEMPENB is set to 1, this module sets the BEMP bit in INTSTS0 to 1, allowing the USB interrupt to be generated.

The following describes the conditions on which this module generates the internal BEMP interrupt request.

1. For the pipe in the transmitting direction, when the FIFO buffer of the corresponding pipe is empty on completion of transmission (including zero-length packet transmission). In single buffer mode, the internal BEMP interrupt request is generated simultaneously with the BRDY interrupt for the pipe other than DCP. However, the internal BEMP interrupt request is not generated on any of the following conditions.
   - When writing data to the FIFO buffer on the CPU side is started on completion of transmitting data of one plane in double buffer mode.
   - When the buffer is cleared (emptied) by setting the ACLRM or BCLR bit to 1.
   - When IN transfer (zero-length packet transmission) is performed during the control transfer status stage in function controller mode.

2. For the pipe in the receiving direction
   When the successfully-received data packet size exceeds the specified maximum packet size. In this case, this module generates the BEMP interrupt request, setting the corresponding PIPEBEMP bit to 1, and discards the received data and modifies the setting of the PID bits of the corresponding pipe to STALL (11).
   Here, this module returns no response when used as the host controller, and returns STALL response when used as the function controller.
   However, the internal BEMP interrupt request is not generated on any of the following conditions.
   - When a CRC error or bit stuffing error is detected in the received data.
   - When a setup transaction is being performed.

Writing 0 to the PIPEBEMP bit clears the status; writing 1 to the PIPEBEMP bit has no effect.

chanhvinh.com

(5) Device State Transition Interrupt (Function Controller Mode)

Figure 28.2 shows a diagram of how this module handles the device state transitions. This module monitors device states and generates device state transition interrupts. However, recovery from the suspended state (resume signal detection) is detected by means of the resume interrupt. The device state transition interrupts can be enabled or disabled individually by using INTENB0. The device state after a transition can be confirmed by using the DVSQ bits in INTSTS0.

When making a transition to the default state, the device state transition interrupt is generated after the reset handshake protocol has been completed.

Device state can be monitored only when the function controller mode is selected. Also, the device state transition interrupts can be generated only when the function controller mode is selected.

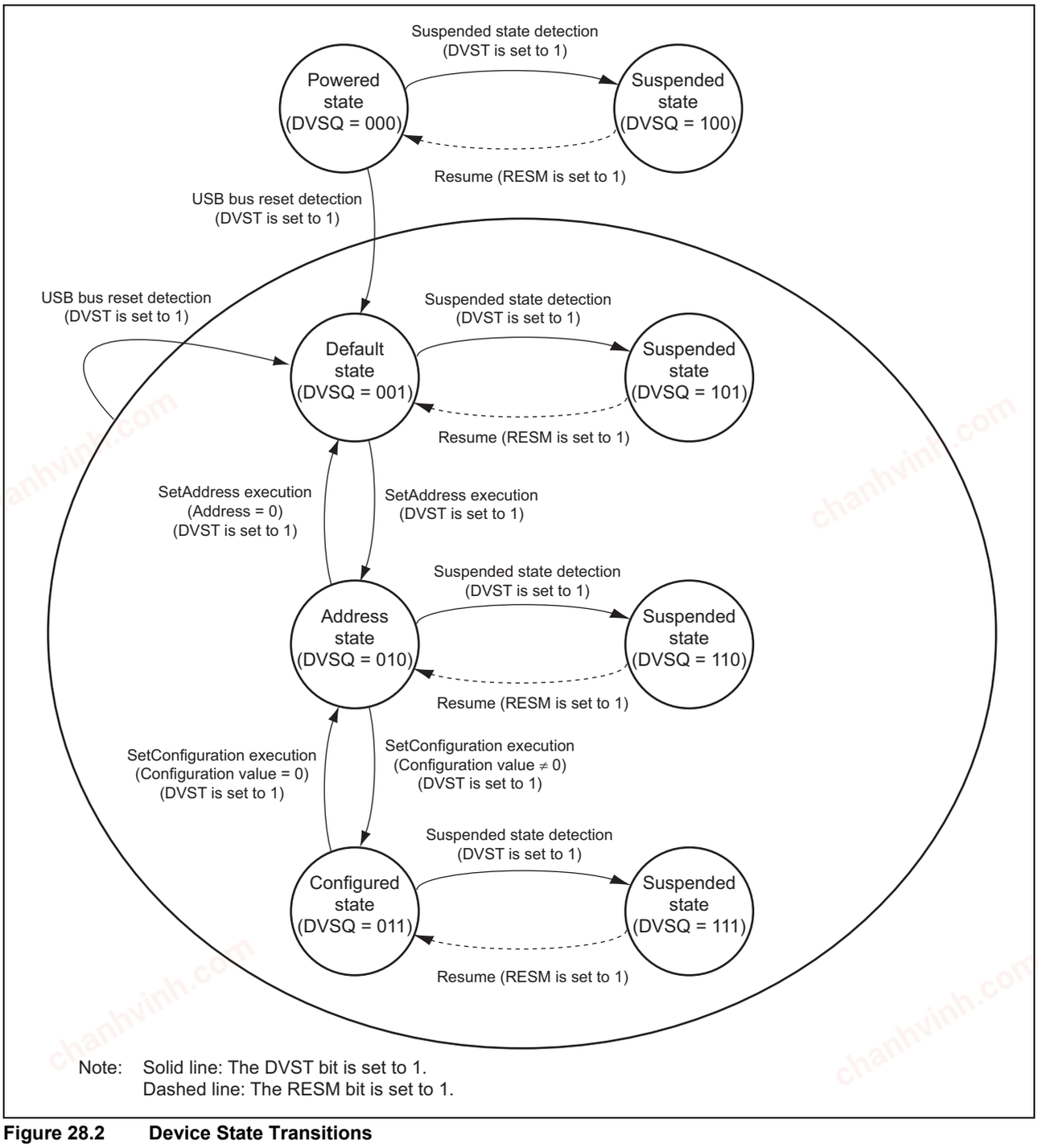


Figure 28.2 Device State Transitions

chanhvinh.com

(6) Control Transfer Stage Transition Interrupt (Function Controller Mode)

Figure 28.3 shows a diagram of how this module handles the control transfer stage transition. This module monitors the control transfer sequence and generates control transfer stage transition interrupts. Control transfer stage transition interrupts can be enabled or disabled individually using INTENB0. The control transfer stage after a transition can be confirmed using the CTSQ bits in INTSTS0.

The control transfer stage transition interrupts are generated only when the function controller mode is selected.

The control transfer sequence errors are described below. If an error occurs, the PID bits in DCPCTR are set to B'1x (STALL).

1. During control read transfers
   - At the IN token of the data stage, an OUT or PING token is received when there have been no data transfers at all.
   - An IN token is received at the status stage
   - A packet is received at the status stage for which the data packet is DATAPID = DATA0
2. During control write transfers
   - At the OUT token of the data stage, an IN token is received when there have been no ACK response at all
   - A packet is received at the data stage for which the first data packet is DATAPID = DATA0
   - At the status stage, an OUT or PING token is received
3. During control write no-data transfers
   - At the status stage, an OUT or PING token is received

At the control write transfer data stage, if the number of receive data exceeds the wLength value of the USB request, it cannot be recognized as a control transfer sequence error. At the control read transfer status stage, packets other than zero-length packets are received by an ACK response and the transfer ends normally.

When a CTRT interrupt occurs in response to a sequence error (SERR = 1), the CTSQ = 110 value is retained until CTRT = 0 is written from the system (the interrupt status is cleared). Therefore, while CTSQ = 110 is being retained, the CTRT interrupt that indicates completion of the setup stage will not be generated even if a further USB request is received (this module retains the indication of completion of the setup stage, and after the interrupt status flag has been cleared, a CTRT interrupt is generated).

chanhvinh.com

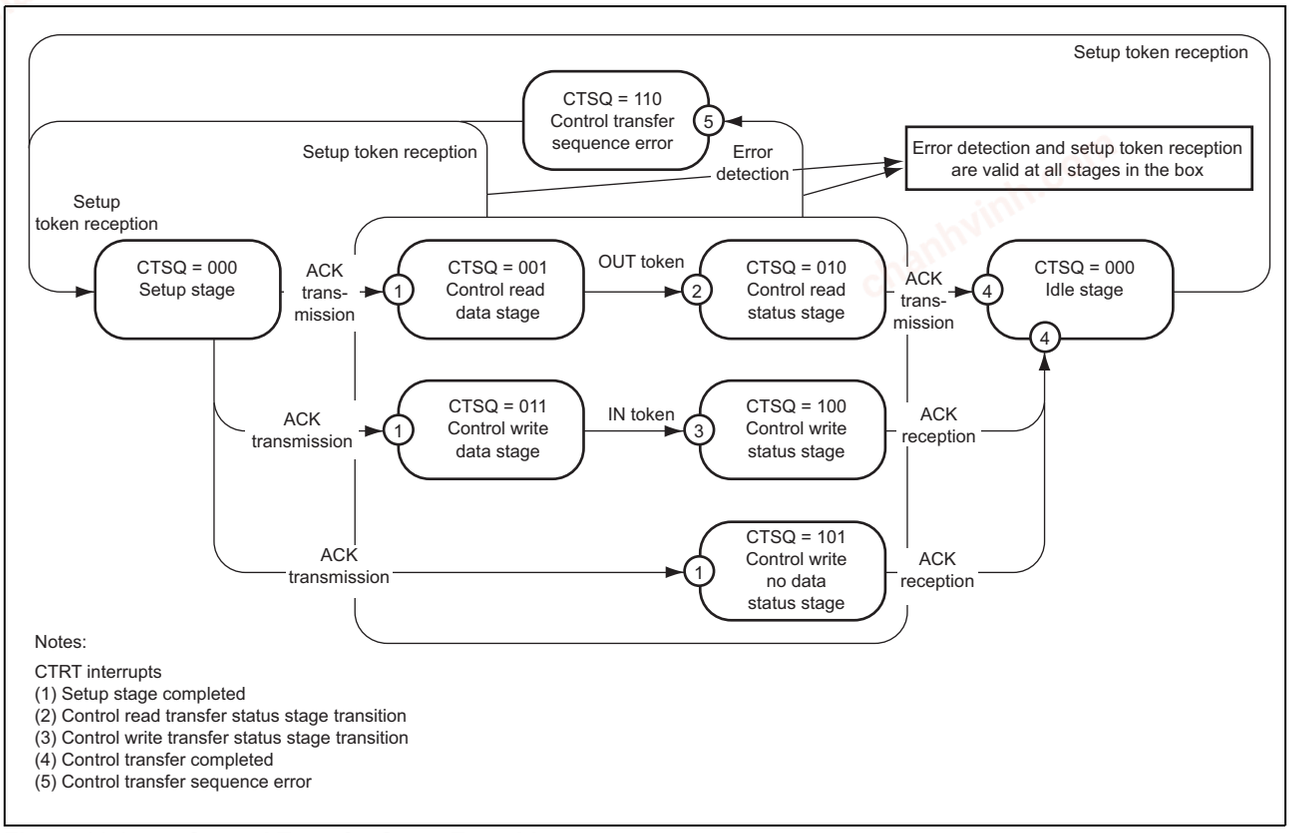


Figure 28.3 Control Transfer Stage Transitions

chanhvinh.com

28.4.3 Pipe Control

Table 28.19 lists the pipe setting items of this module. With USB data transfer, data transmission has to be carried out using the logic pipe called the endpoint. This module has 16 pipes that are used for data transfer. Settings should be entered for each of the pipes in conjunction with the specifications of the system.

Table 28.19 Pipe Setting Items

Register Name	Bit Name	Setting Contents	Remarks
DCPCFG PIPECFG	TYPE	Specifies the transfer type	PIPE1 to PIPE15: Can be set
	BFRE	Selects the BRDY interrupt mode	PIPE1 to PIPE5, PIPE11 to PIPE15: Can be set
	DBLB	Selects a double buffer	PIPE1 to PIPE5, PIPE11 to PIPE15: Can be set PIPE9 and PIPE10: Can be set (only when bulk transfer has been selected).
	CNTMD	Selects continuous transfer or non-continuous transfer	DCP: Can be set. PIPE1 and PIPE2, PIPE9 and PIPE10: Can be set (only when bulk transfer has been selected). PIPE3 to PIPE5, PIPE11 to PIPE15: Can be set
	DIR	Selects transfer direction	IN or OUT can be set
	EPNUM	Endpoint number	PIPE1 to PIPE15: Can be set A value other than 0000 should be set when the pipe is used.
	SHTNAK	Selects disabled state for pipe when transfer ends	DCP: Can be set. PIPE1 and PIPE2, PIPE9 and PIPE10: Can be set (only when bulk transfer has been selected) PIPE3 to PIPE5, PIPE11 to PIPE15: Can be set
PIPEBUF	BUFSIZE	Buffer memory size	DCP: Cannot be set (fixed at 256 bytes) PIPE1 to PIPE5, PIPE9 to PIPE15: Can be set (a maximum of 2 Kbytes can be specified) PIPE6 to PIPE8: Cannot be set (fixed at 64 bytes)
	BUFNMB	Buffer memory number	DCP: Cannot be set (areas fixed at H'0 to H'3) PIPE1 to PIPE5, PIPE9 to PIPE15: Can be set (can be specified in areas H'7 to H'7F) PIPE6 to PIPE8: Cannot be set (areas fixed at H'4 to H'6)
DCPMAXP PIPEMAXP	DEVSEL	Selects a device	Referenced only when the host controller mode is selected.
	MXPS	Maximum packet size	Compliant with the USB standard.
PIPEPERI	IFIS	Buffer flush	PIPE1 and PIPE2: Can be set (only when isochronous transfer has been selected) PIPE3 to PIPE15: Cannot be set
	IITV	Interval counter	PIPE1 and PIPE2: Can be set (only when isochronous transfer has been selected) PIPE3 to PIPE5: Cannot be set PIPE6 to PIPE9: Can be set (only when the host controller mode has been selected) PIPE10 to PIPE15: Can be set

chanhvinh.com

Table 28.19 Pipe Setting Items

Register Name	Bit Name	Setting Contents	Remarks
DCPCTR PIPEnCTR	BSTS	Buffer status	For the DCP, receive buffer status and transmit buffer status are switched with the ISEL bit.
	INBUFM	IN buffer monitor	Mounted for only PIPE3 to PIPE5 and PIPE9 to PIPE15.
	SUREQ	Setup request	Can be set only for the DCP. Can be controlled only when the host controller mode has been selected.
	SUREQCLR	SUREQ clear	Can be set only for the DCP. Can be controlled only when the host controller mode has been selected.
	CSCLR	CSSTS clear	Can be controlled only when the host controller mode has been selected.
	CSSTS	SPLIT status indication	Can be referenced only when the host controller mode has been selected.
	ATREPM	Auto response mode	PIPE1 to PIPE5, PIPE9 to PIPE15: Can be set Can be set only when the function controller mode has been selected.
	ACLRM	Auto buffer clear	PIPE1 to PIPE15: Can be set
	SQCLR	Sequence clear	Clears the data toggle bit
	SQSET	Sequence set	Sets the data toggle bit
	SQMON	Sequence monitor	Monitors the data toggle bit
	PBUSY	Pipe busy confirmation	
	PID	Response PID	
PIPEnTRE	TRENB	Transaction counter enable	PIPE1 to PIPE5, PIPE9 to PIPE15: Can be set
	TRCLR	Current transaction counter clear	PIPE1 to PIPE5, PIPE9 to PIPE15: Can be set
PIPEnTRN	TRNCNT	Transaction counter	PIPE1 to PIPE5, PIPE9 to PIPE15: Can be set

chanhvinh.com

(1) Pipe Control Register Switching Procedures

The following bits in the pipe control registers are only modifiable when USB communication is disabled (PID = NAK): Figure 28.4 shows the procedure for modifying the pipe control registers from the USB communication enabled (PID = BUF) state.

[Bits that should not be set in the USB communication enabled (PID = BUF) state]

- All bits in DCPCFG and DCPMAXP
- The SQCLR and SQSET bits in DCPCTR
- All bits in PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI
- The ATREPM, ACLRM, SQCLR, and SQSET bits in PIPEnCTR
- All bits in PIPEnTRE and PIPEnTRN
- All bits in DEVADDn

In addition to the settings described for the CSCLR bit and all bits in DEVADDn, the settings for each bit described in section 28.3, Register Descriptions must also be complied with.

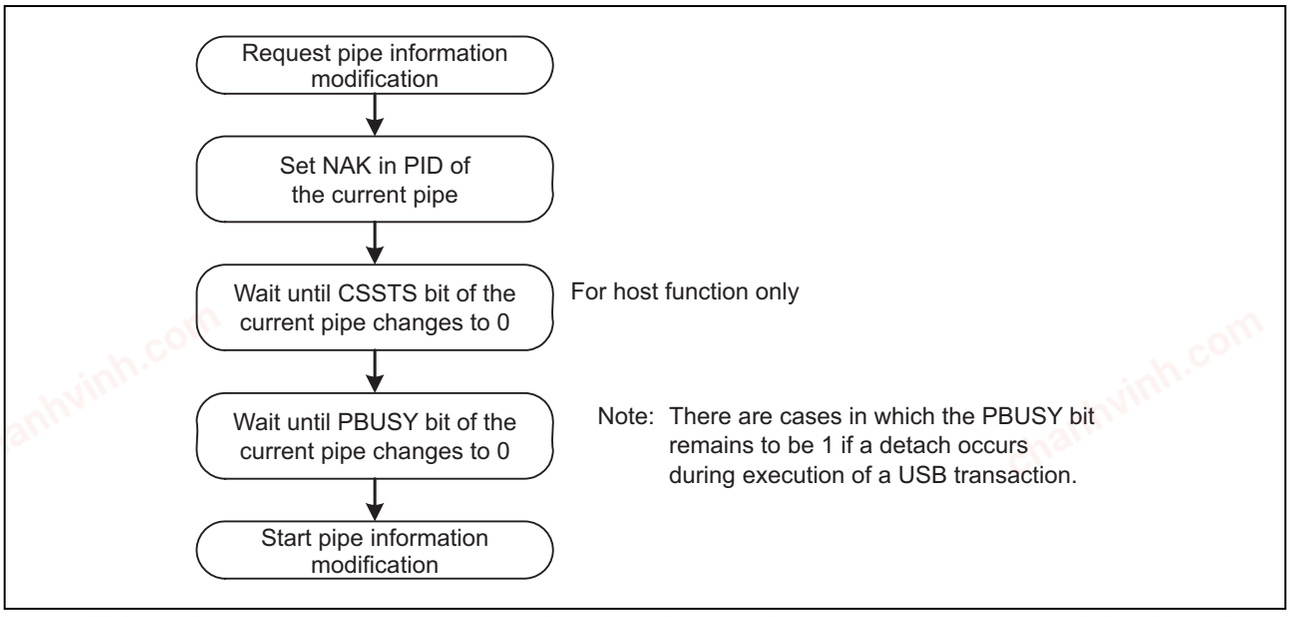


Figure 28.4 Procedure for Modifying Pipe Information from USB Communication Enabled (PID = BUF) State

The following bits in the pipe control registers are only modifiable when the pertinent pipe has not been specified by the CURPIPE bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

[Bits that should not be set when the pertinent pipe is specified by the CURPIPE bits in FIFO port select registers]

- All bits in DCPCFG and DCPMAXP
- All bits in PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI
- The ACLRM bit in PIPEnCTR

In order to modify pipe information, the CURPIPE bits should be set to the pipes other than the pipe to be modified. For the DCP, the buffer should be cleared using BCLR after the pipe information is modified.

chanhvinh.com

(2) Maximum Packet Size Setting

The MXPS bits in DCPMAXP and PIPEMAXP are used to specify the maximum packet size for each pipe. DCP, PIPE1 to PIPE5, and PIPE11 to PIPE15 can be set to any of the maximum packet sizes defined by the USB Specification. For PIPE6 to PIPE10, 64 bytes are the upper limit of the maximum packet size. The maximum packet size should be set before beginning the transfer (PID = BUF).

- DCP: 64 should be set when using high-speed operation.
- DCP: Select and set 8, 16, 32, or 64 when using full-speed operation.
- PIPE1 to PIPE5: 512 should be set when using high-speed bulk transfer.
- PIPE1 to PIPE5: Select and set 8, 16, 32, or 64 when using full-speed bulk transfer.
- PIPE1 and PIPE2: Set a value between 1 and 1024 when using high-speed isochronous transfer.
- PIPE1 and PIPE2: Set a value between 1 and 1023 when using full-speed isochronous transfer.
  For details, see section 28.4.9, Isochronous Transfers (PIPE1 and PIPE2).
- PIPE6 to PIPE8: Set a value between 1 and 64.
- PIPE9: 64 should be set when using interrupt transfer (only when the host controller mode has been selected).
  512 should be set when using high-speed bulk transfer (only when the function controller mode has been selected).
  Select and set 8, 16, 32, or 64 when using full-speed bulk transfer (only when the function controller mode has been selected).
- PIPE10 to PIPE15: 512 should be set when using high-speed bulk transfer (only when the function controller mode has been selected).
- PIPE10 to PIPE15: Select and set 8, 16, 32, or 64 when using full-speed bulk transfer (only when the function controller mode has been selected).

The high bandwidth transfers used with interrupt transfers and isochronous transfers are not supported.

chanhvinh.com

(3) Response PID

The PID bits in DCPCTR and PIPEnCTR are used to set the response PID for each pipe.
The following shows this module operation with various response PID settings:

1. Response PID settings when the host controller mode is selected
   The response PID is used to specify the execution of transactions.
   - NAK setting: Using pipes is disabled. No transaction is executed.
   - BUF setting: Transactions are executed based on the status of the buffer memory.
     For OUT direction: If there are transmit data in the buffer memory, an OUT token is issued.
     For IN direction: If there is an area to receive data in the buffer memory, an IN token is issued.
   - STALL setting: Using pipes is disabled. No transaction is executed.

Note: • Setup transactions for the DCP are set with the SUREQ bit.

2. Response PID settings when the function controller mode is selected
   The response PID is used to specify the response to transactions from the host.
   - NAK setting: The NAK response is always returned in response to the generated transaction.
   - BUF setting: Responses are made to transactions based on the status of the buffer memory.
   - STALL setting: The STALL response is always returned in response to the generated transaction.

Note: • For setup transactions, an ACK response is always returned, regardless of the PID setting, and the USB request is stored in registers USBREQ, USBVAL, USBINDX, and USBLENG.

This module may carry out writing to the PID bits, depending on the results of the transaction. Writing to the PID bits by this module is carried out in the following cases.

1. When the host controller mode has been selected and the response PID is set by this module
   - NAK setting: In the following cases, PID = NAK is set and issuing of tokens is automatically stopped:
     - For transfer that is not isochronous, any combination of the following two items occurring three consecutive times in response to transmitted tokens
       1) no response being returned or 2) a reception error such as a CRC error or a bit stuffing error
     - For isochronous transfer, a reception error such as a CRC error or a bit stuffing error occurring three consecutive times in response to transmitted tokens
     - Reception of a short packet at the stage of control read transfer data when the setting of the SHTNAK bit in PIPECFG is 1
     - If a short packet is received when the SHTNAK bit in PIPECFG has been set to 1 for bulk transfer.
     - If counting by the transaction counter ends while the SHTNAK bit in PIPECFG is set to 1 during bulk transfer.
   - BUF setting: There is no BUF writing by this module.
   - STALL setting: In the following cases, PID = STALL is set and issuing of tokens is automatically stopped:
     When STALL is received in response to the transmitted token.
     When the size of the receive data packet exceeds the maximum packet size.

2. When the function controller mode has been selected and the response PID is set by this module
   - NAK setting: When the setup token is received normally (DCP only).
     If counting by the transaction counter ends or a short packet is received while the SHTNAK bit in PIPECFG is set to 1 during bulk transfer.
   - BUF setting: There is no BUF writing by this module.
   - STALL setting: When a maximum packet size exceeded error has been detected in the receive data packet.
     When a control transfer sequence error has been detected.

chanhvinh.com

(4) Data PID Sequence Bit

This module automatically toggles the sequence bit in the data PID when data is transferred normally in the control transfer data stage, bulk transfer and interrupt transfer. The sequence bit of the data PID that is to be transmitted for the next transaction can be confirmed with the SQMON bit in DCPCTR or PIPEnCTR. When data is transmitted, the sequence bit switches at the timing at which the ACK handshake is received. When data is received, the sequence bit switches at the timing at which the ACK handshake is transmitted. The SQCLR and SQSET bits in DCPCTR or PIPEnCTR can be used to change the data PID sequence bit.

When the function controller mode has been selected and control transfer is used, this module automatically sets the sequence bit when a stage transition is made. The data PID sequence bit becomes DATA1 when the setup stage ends and this module does not reference the sequence bit and responds with PID = DATA1 in the status stage. Therefore, settings are not required. However, when the host controller mode has been selected and control transfer is used, the sequence bit should be set at the stage transition.

For the ClearFeature request transmission or reception, the data PID sequence bit should be set, regardless of whether the host controller mode or function controller mode is selected.

With pipes for which isochronous transfer has been set, sequence bit operation cannot be carried out using the SQSET bit.

chanhvinh.com

28.4.4 FIFO Buffer Memory

This section describes the operation of the FIFO buffers incorporated in this module. Unless specifically specified, the buffer operation is the same regardless of whether the host controller mode or function controller mode is selected.

(1) FIFO Buffer Memory Allocation

Figure 28.5 shows an example of a FIFO buffer memory map for this module. The FIFO buffer memory is an area shared by the CPU and this module. In the FIFO buffer memory status, there are times when the access right to the buffer memory is allocated to the user system (CPU side), and times when it is allocated to this module (SIE side).
Independent FIFO buffer memory areas should be set for each pipe. Each memory area can be set using the first block number and the number of blocks (specified using the BUFNMB and BUFSIZE bits in PIPEBUF), where one block comprises 64 bytes.
When continuous transfer mode has been selected using the CNTMD bit in PIPECFG, the BUFSIZE bits should be set so that the buffer memory size should be an integral multiple of the maximum packet size. When double buffer mode has been selected using the DBLB bit in PIPECFG, two planes of the memory area specified using the BUFSIZE bits in PIPEBUF can be assigned to a single pipe.
Moreover, three FIFO ports are used for access to the FIFO buffer memory (reading and writing data). A pipe is assigned to the FIFO port by specifying the pipe number using the CURPIPE bits in CFIFOSEL/DnFIFOSEL.
The FIFO buffer status of each pipe can be confirmed using the BSTS bit in DCPCTR or the BSTS and INBUFM bits in PIPEnCTR. Also, the access right of the FIFO port can be confirmed using the FRDY bit in CFIFOCTR or DnFIFOCTR.

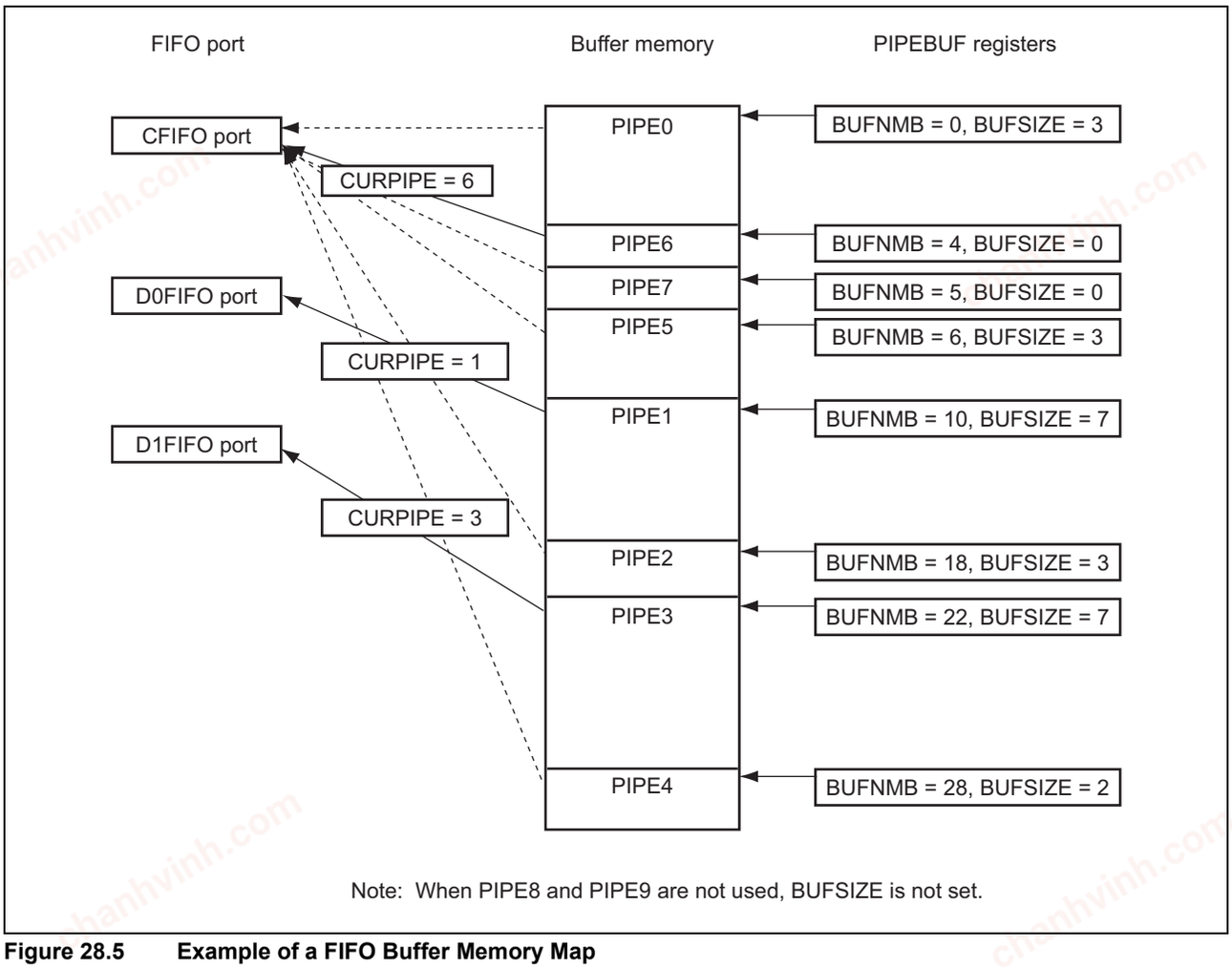


Figure 28.5 Example of a FIFO Buffer Memory Map

chanhvinh.com

(a) FIFO Buffer Clearing

Table 28.20 shows the clearing of the FIFO buffer memory by this module. The FIFO buffer memory can be cleared using the BCLR, DCLRM, and ACLRM bits.

Table 28.20 List of FIFO Buffer Clearing Methods

Bit Name	BCLR	DCLRM	ACLRM
Register	CFIFOCTR DnFIFOCTR	DnFIFOSEL	PIPEnCTR
Function	Clears the FIFO buffer memory on the CPU side	In this mode, after the data of the specified pipe has been read, the FIFO buffer memory is cleared automatically.	This is the auto buffer clear mode, in which all of the received packets are discarded.
Clearing method	Cleared by writing 1	1: Mode valid 0: Mode invalid	1: Mode valid 0: Mode invalid

chanhvinh.com

28.4.5 FIFO Port Functions

This section describes the FIFO port functions. Table 28.21 shows the settings for the FIFO port functions of this module. In write access, writing data until the buffer is full (or the maximum packet size for non-continuous transfers) automatically enables sending of the data to the USB bus. To enable sending of data before the buffer is full (or before the maximum packet size for non-continuous transfers), the BVAL bit in CFIFOCTR/DnFIFOCTR must be set to end the writing (TEND signal for DMA transfers). Also, to send a zero-length packet, the BCLR bit in the same register must be used to clear the buffer and then the BVAL bit is set in order to end the writing.
In read access, reception of new packets is automatically enabled if all of the data has been read. Data cannot be read when a zero-length packet has been received (DTLN = 0), so the BCLR bit in the same register must be used to clear the buffer. The length of the data being received can be confirmed using the DTLN bit in CFIFOCTR/DnFIFOCTR.

Table 28.21 FIFO Port Function Settings

Register Name	Bit Name	Function	Note
CFIFOSEL/ DnFIFOSEL	RCNT	Selects DTLN read mode	
	REW	Buffer memory rewind (re-read, rewrite)	
	DCLRM	Automatically clears data received for a specified pipe after the data has been read	For DnFIFO only
	DREQE	Enables a DMA transfer request	For DnFIFO only
	MBW	FIFO port access bit width	
	BIGEND	Selects FIFO port endian	
	ISEL	FIFO port access direction	For DCP only
	CURPIPE	Selects the current pipe	
CFIFOCTR/ DnFIFOCTR	BVAL	Ends writing to the buffer memory	
	BCLR	Clears the buffer memory on the CPU side	
	FRDY	Monitors whether the FIFO port is ready for access	
	DTLN	Checks the length of received data	

(a) FIFO Port Selection

Table 28.22 shows the pipes that can be selected with each FIFO port. The pipe to be accessed is selected using the CURPIPE bits in CFIFOSEL/DnFIFOSEL. After the pipe is selected, whether the CURPIPE value for the pipe which was written last can be correctly read should be checked. (If the previous pipe number is read, it indicates that the pipe switching is being executed by this module.) Then, the FIFO port can be accessed after FRDY = 1 is checked. Figure 28.6 shows the procedure for pipe switching during access to the FIFO port.
Also, the bus width to be accessed should be selected using the MBW bit. The buffer memory access direction conforms to the DIR bit in PIPECFG. The ISEL bit determines this only for the DCP.

Table 28.22 FIFO Port Access Categorized by Pipe

Pipe	Access Method	Port that can be Used
DCP	CPU access	CFIFO port register
PIPE1 to PIPE15	CPU access	CFIFO port register
	DMA access	D0FIFO/D1FIFO port register

chanhvinh.com

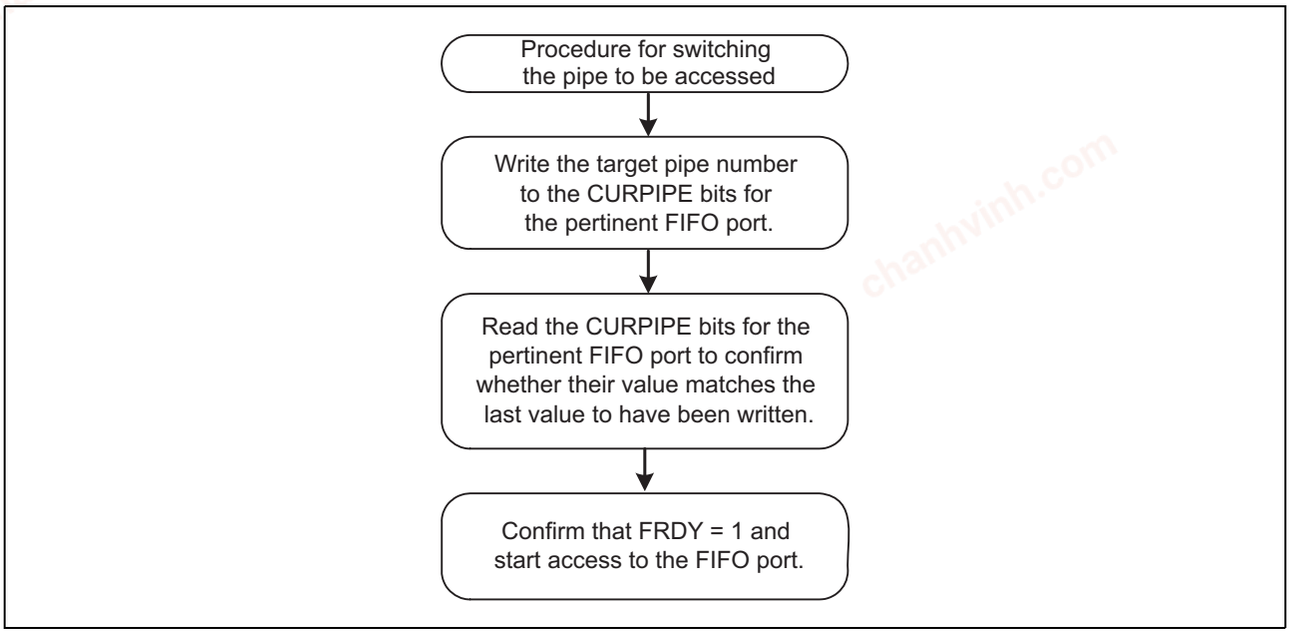


Figure 28.6 Procedure for Pipe Switching during Access to the FIFO Port

(b) DnFIFO Auto Clear Mode (D0FIFO/D1FIFO Port Reading Direction)

If 1 is set for the DCLRM bit in DnFIFOSEL, the module automatically clears the buffer memory of the selected pipe when reading of the data from the buffer memory has been completed.

Table 28.23 shows the packet reception and buffer memory clearing processing for each of the various settings. As shown in Table 28.23, the buffer clear conditions depend on the value set to the BFRE bit. Using the DCLRM bit eliminates the need for the buffer to be cleared even if a situation occurs that necessitates clearing of the buffer. This makes it possible to carry out DMA transfers without involving software.

This function can be set only in the buffer memory reading direction.

Table 28.23 Packet Reception and Buffer Memory Clearing Processing

Buffer Status when Packet is Received	Register Setting			
	DCLRM = 0		DCLRM = 1	
	BFRE = 0	BFRE = 1	BFRE = 0	BFRE = 1
Buffer full	Doesn't need to be cleared	Doesn't need to be cleared	Doesn't need to be cleared	Doesn't need to be cleared
Zero-length packet reception	Needs to be cleared	Needs to be cleared	Doesn't need to be cleared	Doesn't need to be cleared
Normal short packet reception	Doesn't need to be cleared	Needs to be cleared	Doesn't need to be cleared	Doesn't need to be cleared
Transaction count ended	Doesn't need to be cleared	Needs to be cleared	Doesn't need to be cleared	Doesn't need to be cleared

chanhvinh.com

(c) Timing Selection Function for BRDY Interrupts

The BFRE bit of the PIPECFG register can be used to select non-generation of the BRDY interrupt when a data packet with the maximum packet size has been received.

For DMA transfer, this function enables the generation of an interrupt only when the last of the data have been received. Reception of the last of the data indicates either that a short packet was received or completion of counting by the transaction counter. If BFRE = 1, a BRDY interrupt is generated after the received data have been read. The length of the data packet that was last to have been received before the BRDY interrupt was generated can be confirmed by reading the DTLN bits of the DnFIFOCTR register.

Table 28.24 shows the times at which a BRDY interrupt is generated by this module.

Table 28.24 Times BRDY Interrupts are Generated

Buffer Status when Packet is Received \ Register Setting	BFRE = 0	BFRE = 1
Buffer full (normal packet reception)	When a packet is received	No interrupt is generated.
Zero-length packet reception	When a packet is received	When a packet is received
Normal short packet reception	When a packet is received	When received data have been read from buffer memory
Transaction count ended	When a packet is received	When received data have been read from buffer memory

The BFRE bit function is only valid for the reading of buffer memory. In the case of writing, the value of the BFRE bit should always be 0.

chanhvinh.com

28.4.6 Control Transfers (DCP)

Data transfers of the data stage of control transfers are done using the default control pipe (DCP). The DCP buffer memory is a 64-byte single buffer, and is a fixed area that is shared for both control reading and control writing. The buffer memory can be accessed through the CFIFO port.

(1) Control Transfers when the Host Controller Mode is Selected

(a) Setup Stage

USBREQ, USBVAL, USBINDX, and USBLENG are the registers that are used to transmit a USB request for setup transactions. Writing setup packet data to the registers and writing 1 to the SUREQ bit in DCPCTR transmits the specified data for setup transactions. Upon completion of transactions, the SUREQ bit is cleared to 0 by this module. The above USB request registers should not be modified while SUREQ = 1. The device address for setup transactions is specified using the DEVSEL bits in DCPMAXP.

When the data for setup transactions has been sent, a SIGN or SACK interrupt request is generated according to the response received from the peripheral device (SIGN or SACK bit in INTSTS1), by means of which the result of the setup transactions can be confirmed.

A data packet of DATA0 (USB request) is transmitted as the data packet for the setup transactions regardless of the setting of the SQMON bit in DCPCTR.

(b) Data Stage

Data transfers are done using the DCP buffer memory.

The access direction of the DCP buffer memory should be specified using the ISEL bit in CFIFOSEL. The transfer direction should be specified using the DIR bit in DCPCFG.

For the first data packet of the data stage, the data PID must be transferred as DATA1. Accordingly, transaction should be done by setting the data PID to DATA1 by using the SQSET bit in DCPCTR, and setting the response PID to BUF by using the PID bits in DCPCTR. Completion of data transfer is detected using the BRDY and BEMP interrupts.

The data in multiple packets can be transferred in continuous transfer mode. For continuous transfer in the receiving direction, however, note that a BRDY interrupt is only generated when the buffer memory is full or a short packet is received (when the number of bytes of data is an integer multiple of the maximum packet size and no greater than 256 bytes).

For control write transfers, when the number of data bytes to be sent is the integral multiple of the maximum packet size, a zero-length packet must be sent at the end.

(c) Status Stage

Zero-length packet data transfers are done in the direction opposite to that in the data stage. As with the data stage, data transfers are done using the DCP buffer memory. Transactions are done in the same manner as the data stage.

For the data packets of the status stage, the data PID must be transferred as DATA1. The data PID should be set to DATA1 using the SQSET bit in DCPCTR.

For reception of a zero-length packet, the received data length must be confirmed using the DTLN bits in CFIFOCTR after the BRDY interrupt is generated, and the buffer memory must then be cleared using the BCLR bit.

chanhvinh.com

(2) Control Transfers when the Function Controller Mode is Selected

(a) Setup Stage

This module always sends an ACK response in response to a setup packet that is normal with respect to this module. The operation of this module in the setup stage is noted below.

(i) When a new setup packet is received, this module sets the following registers:

- Set the VALID bit in INTSTS0 to 1.
- Set the PID bits in DCPCTR to NAK.
- Set the CCPL bit in DCPCTR to 0.

(ii) When a data packet is received right after the setup packet, the USB request parameters are stored in USBREQ, USBVAL, USBINDX, and USBLENG.

Response processing with respect to the control transfer should always be carried out after first setting VALID = 0. In the VALID = 1 state, PID = BUF cannot be set, and the data stage cannot be terminated.

Using the function of the VALID bit, this module is able to interrupt the processing of a request currently being processed if a new USB request is received during a control transfer, and can send a response in response to the newest request.

Also, this module automatically judges the direction bit (bit 8 of the bmRequestType) and the request data length (wLength) of the USB request that was received, and then distinguishes between control read transfers, control write transfers, and control write no-data transfers, and monitors the stage transition. For a wrong sequence, the sequence error of the control transfer stage transition interrupt is generated. For information on the stage control of this module, see Figure 28.3.

(b) Data Stage

Data transfers corresponding to USB requests that have been received should be done using the DCP. Before accessing the DCP buffer memory, the access direction should be specified using the ISEL bit in CFIFOSEL.

Transaction should be done by setting the response PID to BUF by using the PID bits in DCPCTR.

Completion of data transfer is detected using the BRDY interrupt for control write transfers and BEMP interrupt for control read transfers, respectively.

With control write transfers during high-speed operation, the NYET handshake response is carried out based on the state of the buffer memory.

(c) Status Stage

Control transfers are terminated by setting the CCPL bit to 1 with the PID bits in DCPCTR set to BUF.

After the above settings have been entered, this module automatically executes the status stage in accordance with the data transfer direction determined at the setup stage. The specific procedure is as follows.

- For control read transfers
  This module receives a zero-length packet from the USB host controller and transmits an ACK response.
- For control write transfers and no-data control transfers
  This module transmits a zero-length packet and receives an ACK response from the USB host controller.

chanhvinh.com

(d) Control Transfer Auto Response Function

This module automatically responds to a normal SET_ADDRESS request. If any of the following errors occur in the SET_ADDRESS request, a response is necessary.

- bmRequestType ≠ H'00
- wIndex ≠ H'00
- wLength ≠ H'00
- wValue > H'7F
- DVSQ = B'011 (Configured)

For all requests other than the SET_ADDRESS request, corresponding responses are required.

28.4.7 Bulk Transfers (PIPE1 to PIPE5, PIPE9 to PIPE15)

The buffer memory specifications for bulk transfers (single/double buffer setting, or continuous/non-continuous transfer mode setting) can be selected. The maximum size that can be set for the buffer memory is 2 Kbytes. The buffer memory state is controlled by this module, with a response sent automatically for a PING packet/NYET handshake.

(1) PING Packet Control when the Host Controller Mode is Selected

This module automatically sends a PING packet in the OUT direction at a specific timing.
On receiving an ACK handshake in the initial state in which PING packet sending mode is set, this module sends an OUT packet as noted below. Reception of an NAK or NYET handshake returns this module to PING packet sending mode. This control also applies to the control transfers in the data stage and status stage.
<Starts sending an OUT data>

1. Sets OUT data sending mode.
2. Sends a PING packet.
3. Receives an ACK handshake.
4. Sends an OUT data packet.
5. Receives an ACK handshake.
   (Repeats steps 4 and 5.)
6. Sends an OUT data packet.
7. Receives an NAK/NYET handshake.
8. Sends a PING packet.

This module is returned to PING packet sending mode by a power-on reset, receiving an NYET/NAK handshake, clearing the sequence toggle bit (SQCLR), and setting the buffer clear bit (ACLRM) in PIPEnCTR.

(2) NYET Handshake Control when the Function Controller Mode is Selected

Table 28.25 lists responses to tokens received in bulk or control transfer. The NYET response of this module is made when there is only enough space in the buffer memory for one packet when an OUT token for such transfer is received. When a short packet is received, an ACK response will be produced instead of an NYET response even if the above condition holds.

chanhvinh.com

Table 28.25 NYET Handshake Responses

Value Set for PID Bits in DCPCTR	Buffer Memory State	Received Token	Response	Note
NAK/STALL	—	SETUP	ACK	—
	—	IN/OUT/ PING	NAK/STALL	—
BUF	—	SETUP	ACK	—
	RCV-BRDY	OUT/PING	ACK	If an OUT token is received, a data packet is received.*1
	RCV-BRDY	OUT	NYET	A data packet is received.*2
	RCV-BRDY	OUT (Short)	ACK	A data packet is received.*2
	RCV-BRDY	PING	ACK	*2
	RCV-NRDY	OUT/PING	NAK	
	TRN-BRDY	IN	DATA0/DATA1	A data packet is transmitted.
	TRN-NRDY	IN	NAK	

[Legend]
RCV-BRDY*1:When an OUT/PING token is received, there is space in the buffer memory for two or more packets.
RCV-BRDY*2:When an OUT token is received, there is only enough space in the buffer memory for one packet.
RCV-NRDY:When a PING token is received, there is no space in the buffer memory.
TRN-BRDY:When an IN token is received, there is data to be transmitted in the buffer memory.
TRN-NRDY:When an IN token is received, there is no data to be transmitted in the buffer memory.

chanhvinh.com

28.4.8 Interrupt Transfers (PIPE6 to PIPE9, PIPE10)

When the function controller mode is selected, this module carries out interrupt transfers in accordance with the timing controlled by the host controller. For interrupt transfers, PING packets are ignored (no responses are sent), and the ACK, NAK, and STALL responses are carried out without an NYET handshake response being made.
When the host controller mode is selected, this module can set the timing of issuing a token using the interval counter. At this time, this module issues an OUT token even in the OUT direction, without issuing a PING token.
This module does not support high bandwidth transfers of interrupt transfers.

(1) Interval Counter during Interrupt Transfers when the Host Controller Mode is Selected

(a) Operation Outline

For interrupt transfers, intervals between transactions are set in the IITV bits in PIPEPERI. This module issues an interrupt transfer token based on the specified intervals.

(b) Counter Initialization

This module initializes the interval counter under the following conditions.

- Power-on reset:
  The IITV bits are initialized.
- Buffer memory initialization using the ACLRM bit:
  The IITV bits are not initialized but the count value is. Setting the ACLRM bit to 0 starts counting from the value set in the IITV bits.

Note that the interval counter is not initialized in the following case.

- USB bus reset, USB suspended:
  The IITV bits are not initialized. Setting 1 to the UACT bit starts counting from the value before entering the USB bus reset state or USB suspended state.

(c) Operation when Transmission/Reception is Impossible at Token Issuance Timing

This module cannot issue tokens even at token issuance timing in the following cases. In such a case, this module attempts transactions at the subsequent interval.

- When the PID is set to NAK or STALL.
- When the buffer memory is full at the token sending timing in the receiving (IN) direction.
- When there is no data to be sent in the buffer memory at the token sending timing in the sending (OUT) direction.

chanhvinh.com

28.4.9 Isochronous Transfers (PIPE1 and PIPE2)

This module has the following functions pertaining to isochronous transfers.

- Notification of isochronous transfer error information
- Interval counter (specified by the IITV bits)
- Isochronous IN transfer data setup control (IDLY function)
- Isochronous IN transfer buffer flush function (specified by the IFIS bit)
- SOF pulse output function

This module does not support the high bandwidth transfers of isochronous transfers.

(1) Error Detection with Isochronous Transfers

This module has a function for detecting the error information in isochronous transfers noted below. Table 28.26 and Table 28.27 show the priority in which errors are confirmed and the interrupts that are generated.

1. PID errors
   - If the PID of the packet being received is illegal
2. CRC errors and bit stuffing errors
   - If an error occurs in the CRC of the packet being received, or the bit stuffing is illegal
3. Exceeded maximum packet size
   - The data size of the received packet exceeded the specified maximum packet size.
4. Overrun and underrun errors
   - When host controller mode is selected

     When using isochronous IN transfers (reception), the IN token was transmitted but there was not enough space in the buffer memory.

     When using isochronous OUT transfers (transmission), the OUT token was transmitted, but the data was not in the buffer memory.
   - When function controller mode is selected

     When using isochronous IN transfers (transmission), the IN token was received but the data was not in the buffer memory.

     When using isochronous OUT transfers (reception), the OUT token was received, but there was not enough space in the buffer memory.
5. Interval errors

When function controller mode is selected, interval errors occur in following cases.

- During an isochronous IN transfer, the IN token could not be received during the interval frame.
- During an isochronous OUT transfer, the OUT token could not be received during the interval frame.

chanhvinh.com

Table 28.26 Error Detection when a Token is Received

Detection Priority Order	Error	Generated Interrupt and Status
1	PID errors	No interrupts are generated in both cases when the host controller mode is selected and the function controller mode is selected (ignored as a corrupted packet).
2	CRC error and bit stuffing errors	No interrupts are generated in both cases when the host controller mode is selected and the function controller mode is selected (ignored as a corrupted packet).
3	Overrun and underrun errors	An NRDY interrupt is generated and the OVRN bit is set in both host controller mode and function controller mode. When the function controller mode is selected, a zero-length packet is transmitted in response to IN token. However, no data packet is received in response to OUT token.
4	Interval errors	An NRDY interrupt is generated when the function controller mode is selected. It is not generated in the host controller mode.

Table 28.27 Error Detection when a Data Packet is Received

Detection Priority Order	Error	Generated Interrupt and Status
1	PID errors	No interrupts are generated (ignored as a corrupted packet)
2	CRC error and bit stuffing errors	An NRDY interrupt is generated and the CRCE bit is set in both host controller mode and function controller mode.
3	Maximum packet size exceeded error	A BEMP interrupt is generated to set the PID bits to STALL in both cases when the host controller mode is selected and the function controller mode is selected.

(2) DATA-PID

This module does not support high bandwidth transfers. When the function controller mode is selected, this module operates as follows in response to the received PID.

1. IN direction
   - DATA0: Sent as data packet PID
   - DATA1: Not sent
   - DATA2: Not sent
   - mDATA: Not sent

2. OUT direction (when using full-speed operation)
   - DATA0: Received normally as data packet PID
   - DATA1: Received normally as data packet PID
   - DATA2: Packets are ignored
   - mDATA: Packets are ignored

3. OUT direction (when using high-speed operation)
   - DATA0: Received normally as data packet PID
   - DATA1: Received normally as data packet PID
   - DATA2: Received normally as data packet PID
   - mDATA: Received normally as data packet PID

chanhvinh.com

(3) Interval Counter

(a) Operation Outline

The isochronous interval can be set using the IITV bits in PIPEPERI. The interval counter enables the functions shown in Table 28.28 when the function controller mode is selected. When the host controller mode is selected, this module generates the token issuance timing. When the host controller mode is selected, the interval counter operation is the same as the interrupt transfer operation. Refer to section 28.4.8 (1) Interval Counter during Interrupt Transfers when the Host Controller Mode is Selected.

Table 28.28 Functions of the Interval Counter when the Function Controller Mode is Selected

Transfer Direction	Function	Conditions for Detection
IN	Transmission buffer flush function	When an IN token cannot be normally received in the interval frame during an isochronous IN transfer
OUT	Notifies that a token not being received	When an OUT token cannot be normally received in the interval frame during an isochronous OUT transfer

The interval count is carried out when an SOF is received or for interpolated SOFs, so the isochronism can be maintained even if an SOF is damaged. The frame interval that can be set is the 2^{IITV} frames or 2^{IITV} µ frames.

(b) Interval Counter Initialization when the Function Controller Mode is Selected

This module initializes the interval counter under the following conditions.

- Power-on reset
  The IITV bits are initialized.
- Buffer memory clearing using the ACLRM bit
  The IITV bits are not initialized but the count value is.
- USB bus reset

After the interval counter has been initialized, the counter is started under the following condition 1 or 2 when a packet has been transferred normally.

1. An SOF is received following transmission of data in response to an IN token, in the PID = BUF state.
2. An SOF is received after data following an OUT token is received in the PID = BUF state.

The interval counter is not initialized under the conditions below.

1. When the PID bits are set to NAK or STALL
   The interval timer does not stop. This module attempts the transactions at the subsequent interval.
2. The USB bus is reset or USB operations are suspended
   The IITV bits are not initialized. When the SOF has been received, the counter is restarted from the value prior to the reception of the SOF.

(c) Interval Counting and Transfer Control when the Host Controller Mode is Selected

The IITV bits can be set when the selected pipe is for isochronous or interrupt transfers. This module controls the interval between token issuance operations based on the IITV bit setting. Specifically, this module issues a token for the selected pipe once every 2^{IITV} (µ) frames.
This module counts the interval every 1-ms frame for the pipes used for communications with the full-speed or low-speed peripheral devices connected to a high-speed HUB.
This module starts counting the token issuance interval at the (µ) frame following the (µ) frame in which the PID bits are

chanhvinh.com

set to BUF.

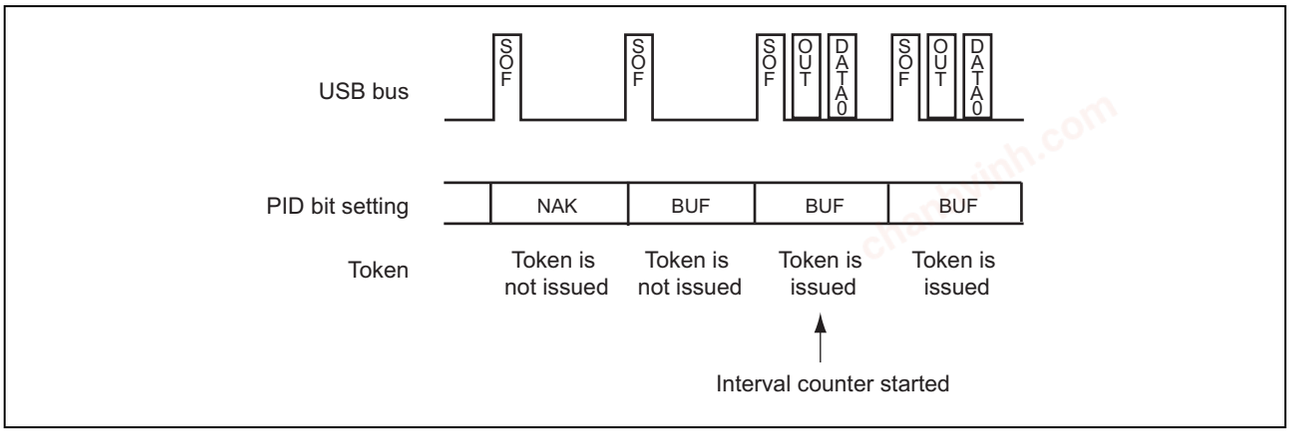


Figure 28.7 Token Issuance when IITV = 0

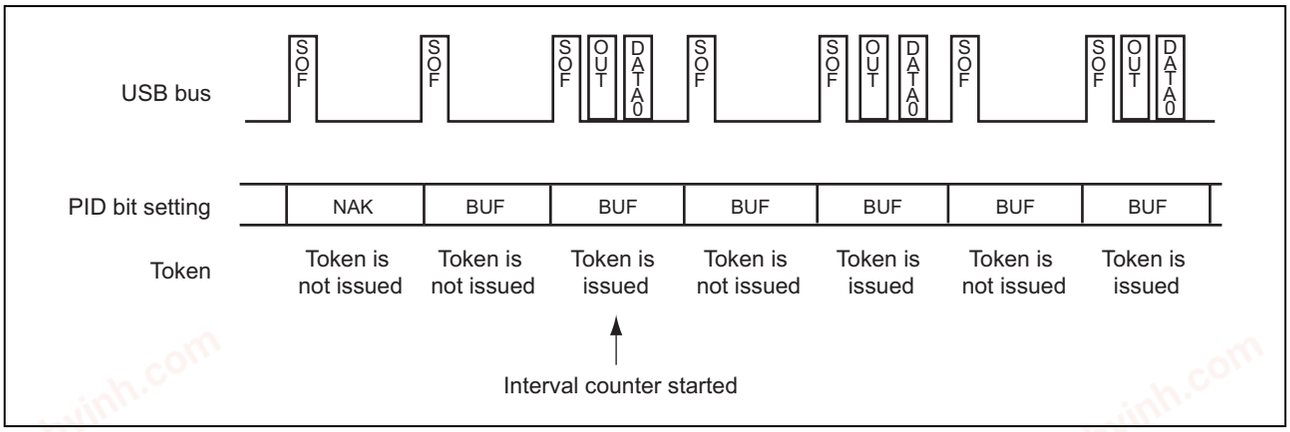


Figure 28.8 Token Issuance when IITV = 1

When the selected pipe is for isochronous transfers, this module carries out the operation below in addition to controlling token issuance interval. This module issues a token even when the NRDY interrupt generation condition is satisfied.

1. When the selected pipe is for isochronous IN transfers
   This module generates the NRDY interrupt when this module issues the IN token but does not receive a packet successfully from a peripheral device (no response or packet error).
   This module sets the OVRN bit to 1 generating the NRDY interrupt when the time to issue an IN token comes in a state in which this module cannot receive data because the FIFO buffer is full (because reading data from the FIFO buffer is slow).
2. When the selected pipe is for isochronous OUT transfers
   This module sets the OVRN bit to 1 generating the NRDY interrupt and transmitting a zero-length packet when the time to issue an OUT token comes in a state in which there is no data to be transmitted in the FIFO buffer (because writing data to the FIFO buffer is slow).
   The token issuance interval is reset by a power-on reset or when the ACLRM bit is set to 1.

(d) Interval Counting and Transfer Control when the Function Controller Mode is Selected

The IITV bits can be set when the selected pipe is for isochronous transfers.

1. When the selected pipe is for isochronous OUT transfers
   This module generates the NRDY interrupt when it fails to receive a data packet within the interval set for (μ) frames by the IITV bits.

chanhvinh.com

This module generates the NRDY interrupt when this module fails to receive a data packet because of a CRC error or other errors contained in the packet, or because of the FIFO buffer being full (because reading data from the FIFO buffer is slow).

This module generates the NRDY interrupt on receiving an SOF packet. Even if the SOF packet is corrupted, the internal interpolation is used and allows the interrupt to be generated at the timing to receive the SOF packet. However, when the IITV bits are set to the value other than 0, this module generates the NRDY interrupt on receiving an SOF packet for every interval after starting interval counting operation. When the PID bits are set to NAK after starting the interval timer, this module does not generate the NRDY interrupt on receiving an SOF packet.

The interval counting starts at the different timing depending on the IITV bit setting as follows.

- When IITV = 0: The interval counting starts at the (μ) frame following the (μ) frame in which the PID bits for the selected pipe are set to BUF.

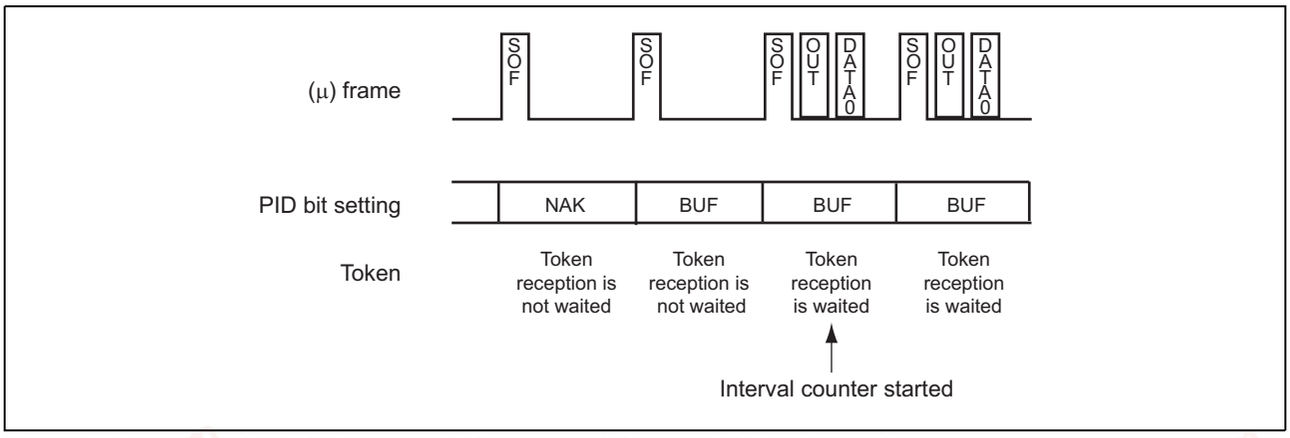


Figure 28.9 Relationship between (μ) Frames and Expected Token Reception when IITV = 0

- When IITV ≠ 0: The interval counting starts on completion of successful reception of the first data packet after the PID bits for the selected pipe have been modified to BUF.

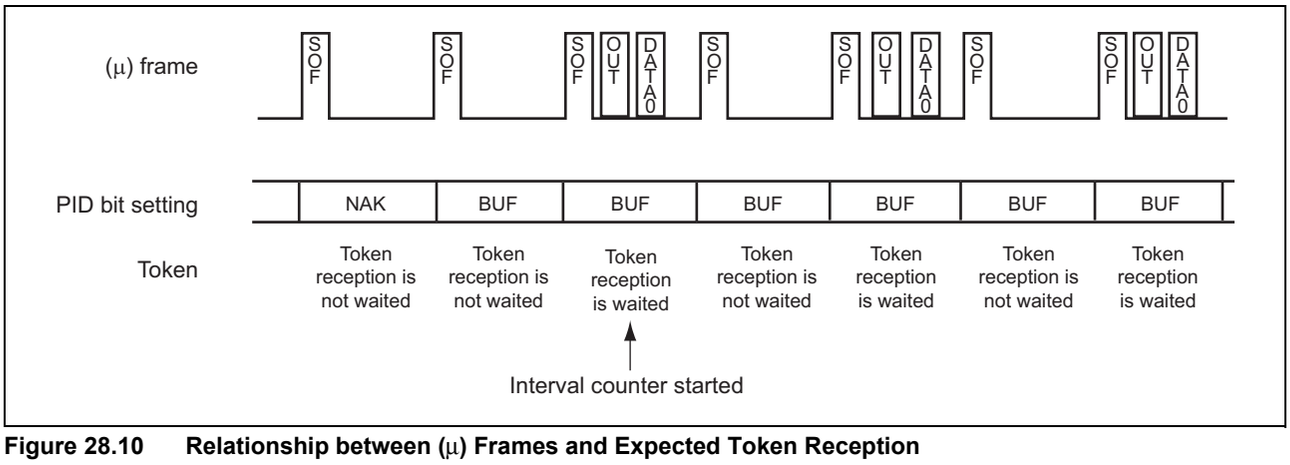


Figure 28.10 Relationship between (μ) Frames and Expected Token Reception when IITV = 1

2. When the selected pipe is for isochronous IN transfers

The IFIS bit should be 1 for this use. When IFIS = 0, this module transmits a data packet in response to the received IN token irrespective of the IITV bit setting.

When IFIS = 1, this module clears the FIFO buffer when this module fails to receive an IN token within the interval set for (μ) frames by the IITV bits in a state in which there is data to be transmitted in the FIFO buffer.

This module also clears the FIFO buffer when this module fails to receive an IN token successfully because of a bus error such as a CRC error contained in the token.

chanhvinh.com

This module clears the FIFO buffer on receiving an SOF packet. Even if the SOF packet is corrupted, the internal interpolation is used and allows the FIFO buffer to be cleared at the timing to receive the SOF packet.
The interval counting starts at the different timing depending on the IITV bit setting (similar to the timing during OUT transfers).
The clearing conditions for the interval counter are any of the following in function controller mode.
- When a power-on-reset is applied to this module (the value set in the IITV bits is also cleared to 0).
- When the ACLRM bit is set to 1.
- When this module detects a USB bus reset.

(4) Setup of Data to be Transmitted using Isochronous Transfer when the Function Controller Mode is Selected

With isochronous data transmission using this module in function controller mode, after data has been written to the buffer memory, a data packet can be sent with the next frame in which an SOF packet is detected. This function is called the isochronous transfer transmission data setup function. This function enables identification of the frame that has started being transmitted.
When a double buffer is in use as the buffer memory, transmission from only one of the two buffers will be possible even after the writing of data to both buffers has been completed; the given buffer memory will be that to which the writing of data was completed first. For this reason, even if multiple IN tokens are received in a single frame, the only buffer memory that can be sent is one packet's worth of data.
When an IN token is received, if the buffer memory is in the transmission enabled state, this module transmits the data as a normal response. If the buffer memory is not in the transmission enabled state, however, a zero-length packet is sent and an underrun error occurs.
Figure 28.11 shows an example of transmission using the isochronous transfer transmission data setup function with this module, when IITV = 0 (every frame) has been set.

chanhvinh.com

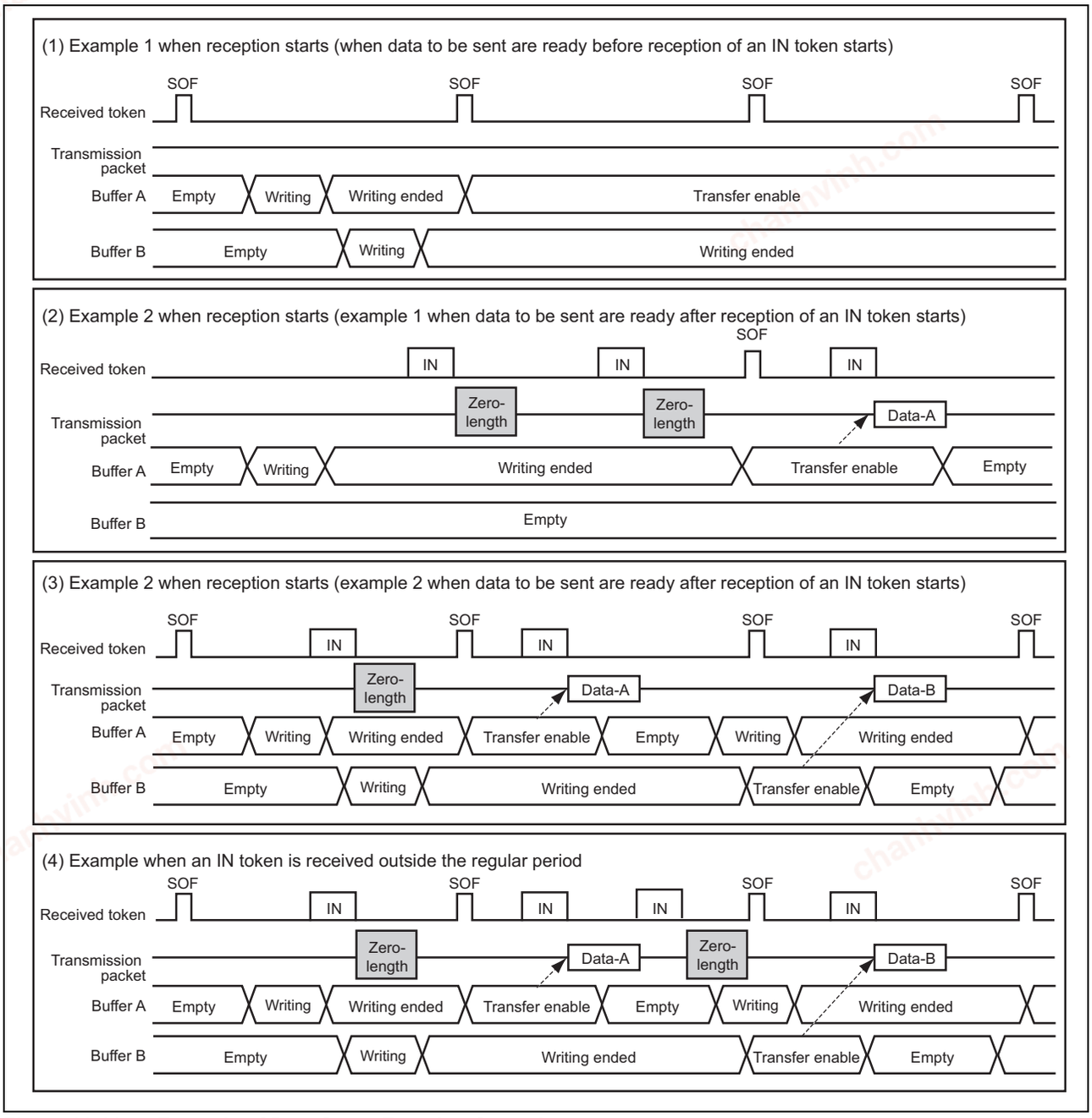


Figure 28.11 Example of Data Setup Function Operation

(5) Isochronous Transfer Transmission Buffer Flush when the Function Controller Mode is Selected

When the function controller mode is selected and an SOF packet or a µSOF packet of the next frame is received without receiving an IN token in the interval frame during isochronous data transmission, this module operates as if a corrupted IN token was received, and clears the buffer for which transmission is enabled, putting that buffer in the writing enabled state.

If a double buffer is being used and writing to both buffers has been completed, data are considered to have been sent from the buffer memory that was cleared in the same interval frame, and transmission is enabled for the buffer memory that is not discarded with SOF or µSOF packets reception.

The timing at which the buffer flush function is activated varies depending on the value set for the IITV bits.

chanhvinh.com

- If IITV = 0
  The buffer flush operation starts from the next frame after the pipe becomes valid.
- In any cases other than IITV = 0
  The buffer flush operation is carried out subsequent to the first normal transaction.

Figure 28.12 shows an example of the buffer flush function of this module. When an unanticipated token is received prior to the interval frame, this module sends the written data or a zero-length packet as an underrun error according to the data setup state.

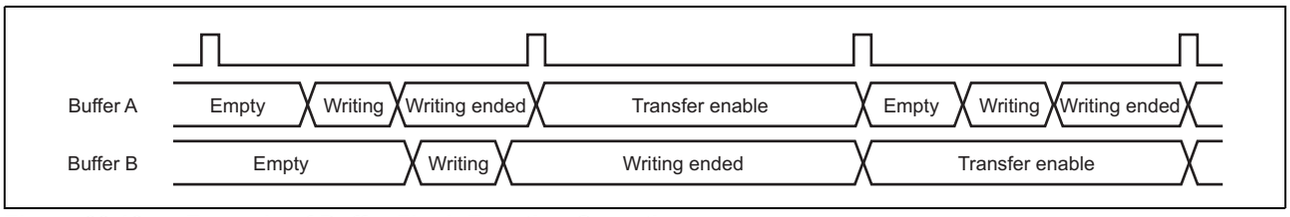


Figure 28.12 Example of Buffer Flush Function Operation

Figure 28.13 shows an example of this module generating an interval error. There are five types of interval errors, as shown below. The interval error is generated at the timing indicated by (1) in the figure, and the buffer flush function is activated.
If an interval error occurs during an IN transfer, the buffer flush function is activated; and if it occurs during an OUT transfer, an NRDY interrupt is generated.
The OVRN bit should be used to distinguish between NRDY interrupts such as received packet errors and overrun errors.
In response to tokens that are shaded in the figure, responses occur based on the buffer memory status.

1. IN direction
   - If the buffer is in the transmission enabled state, the data is transferred as a normal response.
   - If the buffer is in the transmission disabled state, a zero-length packet is sent and an underrun error occurs.
2. OUT direction
   - If the buffer is in the reception enabled state, the data is received as a normal response.
   - If the buffer is in the reception disabled state, the data is discarded and an overrun error occurs.

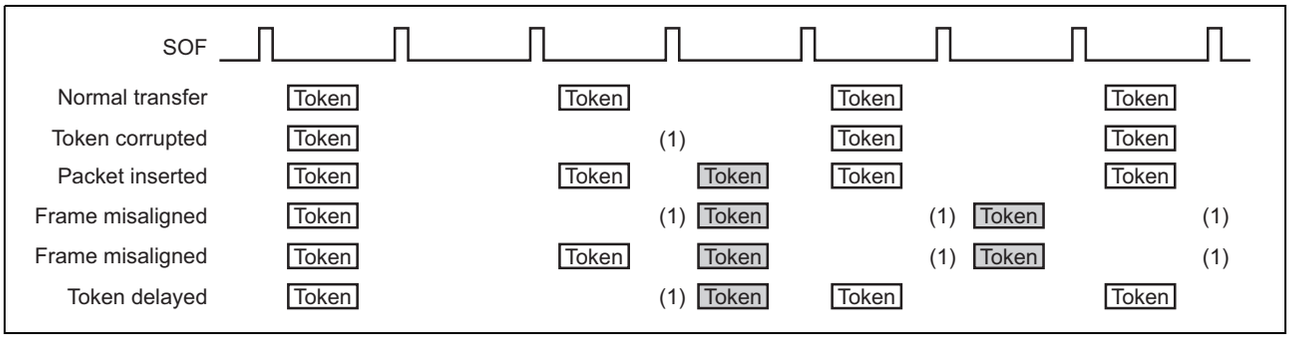


Figure 28.13 Example of an Interval Error being Generated when IITV = 1

chanhvinh.com

28.4.10 SOF Interpolation Function

When the function controller mode is selected and if an SOF packet could not be received at intervals of 1 ms (when using full-speed operation) or 125 µs (when using high-speed operation) because of corruption or missing, this module interpolates the SOF. The SOF interpolation operation begins when both the USBE bit in SYSCFG and the SUSPM bit in SUSPMODE have been set to 1 and an SOF packet is received. The interpolation function is initialized under the following conditions.

- Power-on reset
- USB bus reset
- Suspended state detected

Also, the SOF interpolation operates under the following specifications.

- Frame interval (125 µs or 1 ms) conforms to the results of the reset handshake protocol.
- The interpolation function is not activated until an SOF packet is received.
- After the first SOF packet is received, either 125 µs or 1 ms is counted with an internal clock of 48 MHz, and interpolation is carried out.
- After the second and subsequent SOF packets are received, interpolation is carried out at the previous reception interval.
- Interpolation is not carried out in the suspended state or while a USB bus reset is being received. (With suspended transitions in high-speed operation, interpolation continues for 3 ms after the last packet is received.)

This module supports the following functions based on the SOF reception. These functions also operate normally with SOF interpolation, if the SOF packet was missing.

- Refreshing of the frame number and micro-frame number
- SOFR interrupt and µSOF lock
- SOF pulse output
- Isochronous transfer interval count

If an SOF packet is missing when full-speed operation is being used, the FRNM bits in FRMNUM are not refreshed. If a µSOF packet is missing during high-speed operation, the UFRNM bits in UFRMNUM are refreshed. However, if a µSOF packet for UFRNM = 000 is missing, the FRNM bits are not refreshed. In this case, the FRNM bits are not refreshed even if successive µSOF packets other than UFRNM = 000 are received normally.

chanhvinh.com

29. Video Display Controller 5 (1): Overview

29.1 Features

The video display controller 5 consists of the following six blocks. For the image synthesis, one plane of video image + two graphics planes, or three graphics planes can be selected.

1. Input controller: Input video image selection, sync signal adjustment, horizontal noise reduction, and brightness adjustment, gain adjustment, and YCbCr ↔ GBR conversion using a color matrix
2. Scaler: Scale up, scale down, and rotation of input video images using the frame buffer, and repeated recording of the specified number of fields in the frame buffer
3. Image quality improver: Black stretch, LTI/sharpness, and YCbCr ↔ GBR conversion using a color matrix
4. Image synthesizer: Synthesis of one plane of video image + two graphics planes, or three graphics planes
5. Output controller: Brightness/contrast adjustment, gamma correction, dither processing, output format conversion, control signal output for TFT-LCD panel
6. System controller: Interrupt control, panel clock control, CLUT table select signal status flag output

The functions of video display controller 5 are listed in Table 29.1.

Table 29.1 Features of Video Display Controller 5

Item	Function
Operating frequency	Video input clock: 87 MHz or less (for RGB/YCbCr video image) Panel clock: 87 MHz or less (depends on the panel specifications)
Input video image specification	• 8-bit input conforming to ITU-R BT.656 standard (27 MHz, interlace signal) • 8-bit input conforming to ITU-R BT.656 extended standard (54 MHz, progressive signal) *1 • 8-bit input conforming to ITU-R BT.601 extended standard (27 MHz, interlace signal) *1 • 8-bit input conforming to ITU-R BT.601 extended standard (54 MHz, progressive signal) *1 • 16-bit input conforming to ITU-R BT.601 extended standard (13.5 MHz, interlace signal) *1 • Digital pin input: YCbCr422, YCbCr444, RGB888, RGB666, and RGB565 video image • Digital pin input size: Maximum input video image size to be set *2: 1440 pixels × 1024 lines (horizontal × vertical) Notes:1. The ITU-R BT.656 and 601 standards do not include the description regarding the progressive signal. The ITU-R BT.601 standard does not include the description regarding the connection interface. 2. Depends on the AC characteristics of the connected device. • Examples of input video image size: XGA (1024 × 768), SVGA (800 × 600), WVGA (800 × 480), VGA (640 × 480), WQVGA (480 × 240), QVGA in landscape (320 × 240), QVGA in portrait (240 × 320)
Video image recording function	• Storing the video image in the YCbCr422/YCbCr444/RGB565/RGB888 format at a rate of 1/1, 1/2, 1/4, or 1/8 field. • Maximum video image size to be stored: ×1 size of input video image
Video image quality adjustment function	Contrast adjustment, brightness adjustment, horizontal noise reduction, black stretch, LTI/sharpness
Video image scaling processing	Vertical: ×1/8 to ×8, linear/hold interpolation Horizontal: ×1/8 to ×8, linear/hold interpolation IP conversion can be performed by adjusting the initial phase.
Video image rotation function	• 0/90/180/270 degree rotation in the YCbCr422/RGB565 format • Horizontal mirroring in the YCbCr422/YCbCr444/RGB565/RGB888 format

chanhvinh.com

Table 29.1 Features of Video Display Controller 5

Item	Function
Graphics	• Number of graphic planes: Three planes (graphics 0, graphics 2, and graphics 3) • Supported pixel formats: - RGB565 progressive format (α: none, R: 5 bits, G: 6 bits, B: 5 bits; 16 bits in total) - RGB888 progressive format (α: none, R: 8 bits, G: 8 bits, B: 8 bits; 24 bits in total) - α RGB1555 progressive format (α: 1 bit, R: 5 bits, G: 5 bits, B: 5 bits; 16 bits in total) - α RGB4444 progressive format (α: 4 bits, R: 4 bits, G: 4 bits, B: 4 bits; 16 bits in total) - α RGB8888 progressive format (α: 8 bits, R: 8 bits, G: 8 bits, B: 8 bits; 32 bits in total) - RGBα5551 progressive format (R: 5 bits, G: 5 bits, B: 5 bits, α: 1 bit; 16 bits in total) - RGBα8888 progressive format (R: 8 bits, G: 8 bits, B: 8 bits, α: 8 bits; 32 bits in total) - CLUT8 progressive format (CLUT: 8 bits) - CLUT4 progressive format (CLUT: 4 bits) - CLUT1 progressive format (CLUT: 1 bits) - YCbCr422 progressive format (Y: 8 bits, Cb/Cr: 8 bits; 16 bits in total) (only for graphics 0) - YCbCr444 progressive format (Y: 8 bits, Cb/Cr: 8 bits; 16 bits in total) (only for graphics 0) • Maximum image size to be read: 1440 pixels × 1440 lines (horizontal × vertical)
Graphics function	Alpha blending in rectangular area: Mixes images according to transparency rate α in the specified area (fade-in and fade-out functions are available.) Chroma-key: Mixes images using the specified RGB color and CLUT value according to transparency rate α. Alpha blending in one pixel units: Mixes images according to transparency rate α when the target graphics image is in the αRGB1555, αRGB4444, αRGB8888, RGBα5551, RGBα8888, or CLUT8/4/1 format.
	For each dot, the priority among the α values of the above functions is as follows: Alpha blending in rectangular area > Chroma-key > Alpha blending in one pixel units
Output video image size	Maximum output video image size to be set*: 1999 pixels × 2035 lines (horizontal × vertical) Note: * Depends on the AC characteristics of the display panel. Examples of output video image size: • XGA (1024 × 768) • SVGA (800 × 600), WVGA (800 × 480), • VGA (640 × 480), WQVGA (480 × 240), • QVGA size in landscape (320 × 240) • QVGA size in portrait (240 × 320)
Output video image format	• RGB888 progressive video output (24-bit parallel output) • RGB666 progressive video output (18-bit parallel output) • RGB565 progressive video output (16-bit parallel output) • RGB888 progressive video output (8-bit serial output)
Panel output adjustment	Panel brightness/contrast adjustment, RGB gamma correction, dither processing, output format conversion
Sync signal output	Control signal output for the TFT-LCD panel
Interrupt output	• Vsync signal for video image input/output • Line interrupt output (can be output on a desired line.) • Erroneous Vsync cycle detection signal for video input • Field write completion signal • Overflow/underflow signal for the internal buffer

chanhvinh.com

29.2 Block Diagram

Figure 29.1 to Figure 29.3 show the entire block diagram of this module.

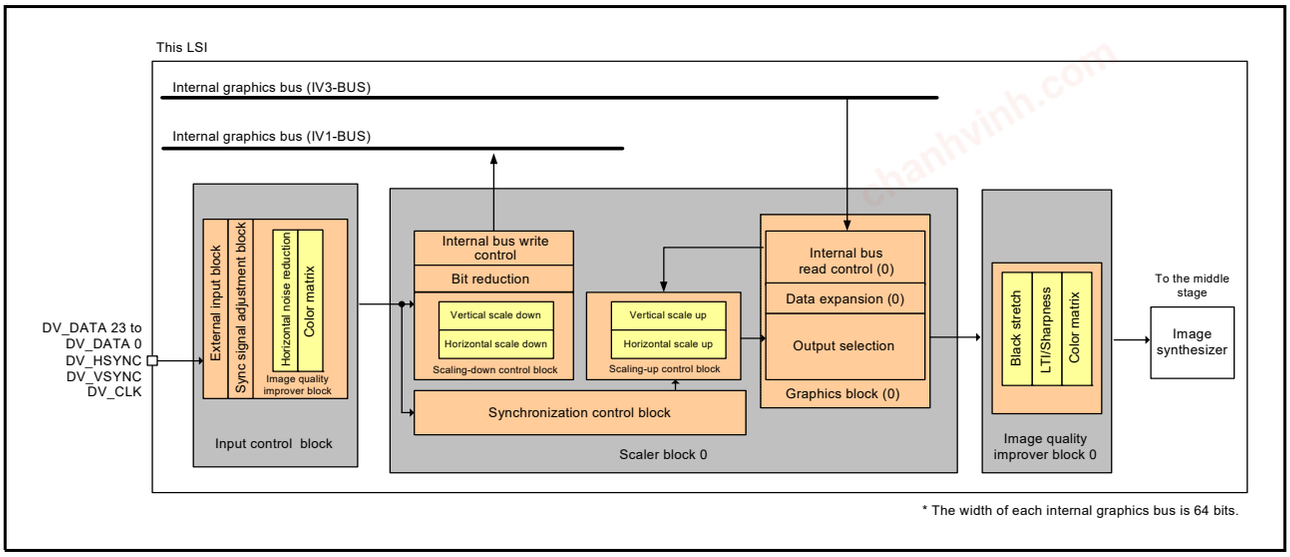


Figure 29.1 Video Display Controller 5 Former Stage Block Diagram

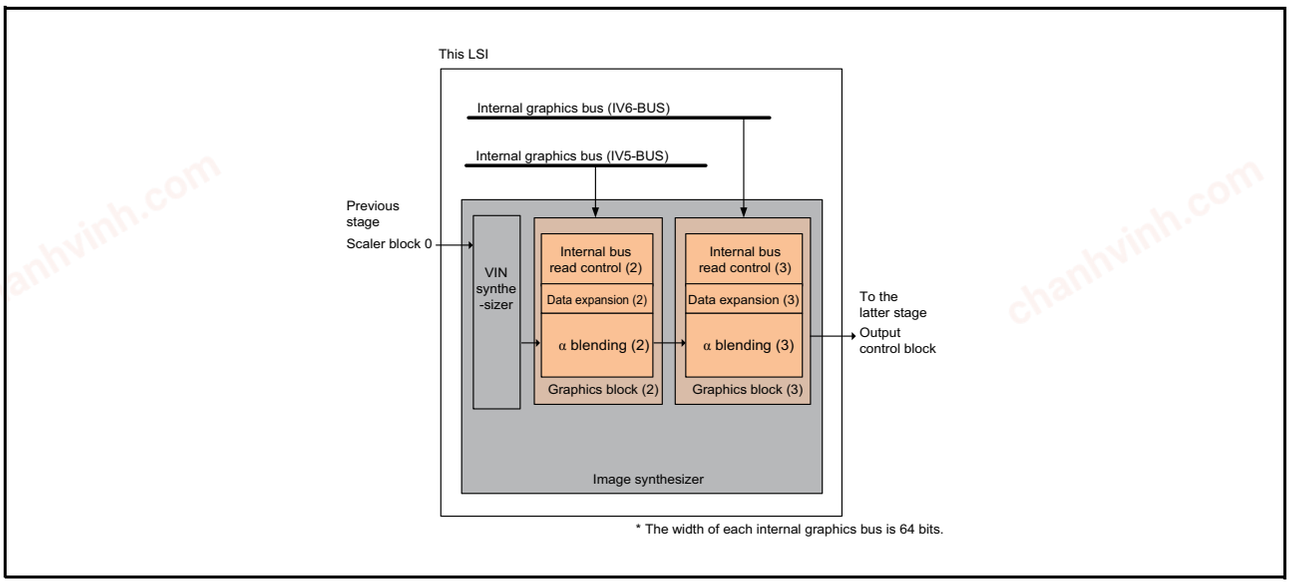


Figure 29.2 Video Display Controller 5 Middle Stage Block Diagram

chanhvinh.com

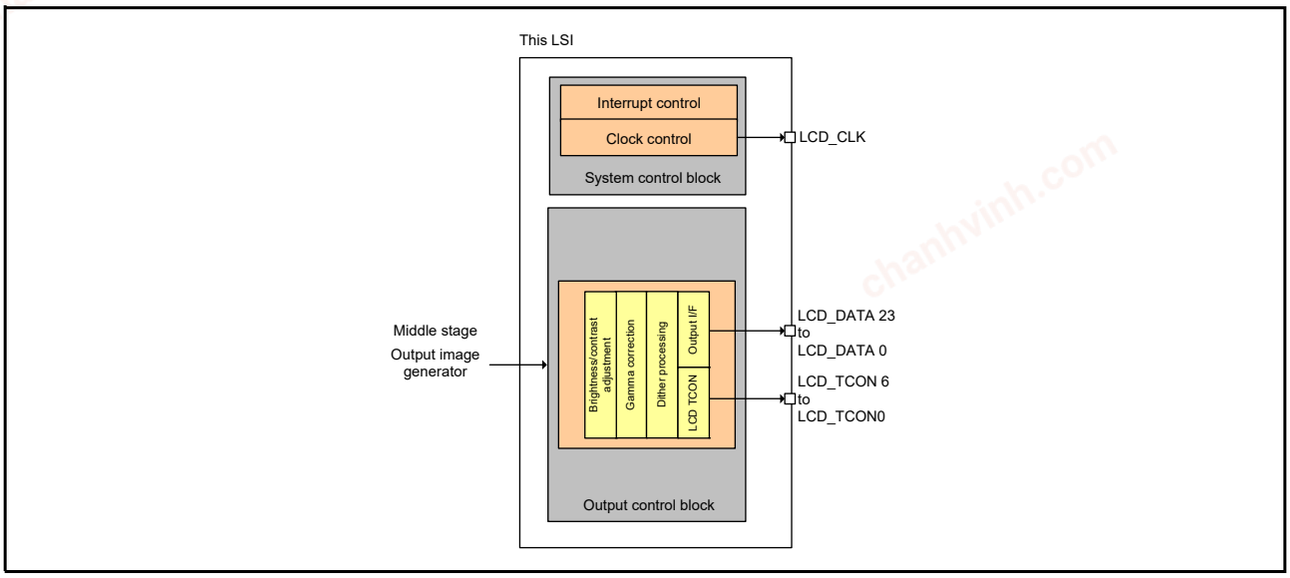


Figure 29.3 Video Display Controller 5 Latter Stage Block Diagram

29.3 Input/Output Pins

Table 29.2 shows the pin configuration.

Table 29.2 Input/Output Pins

Symbol	I/O	Pin Name	Function
DV0_CLK	Input	External input clock 0	External input 0 clock pin
DV0_VSYNC	Input	External input Vsync 0	External input 0 Vsync signal pin
DV0_HSYNC	Input	External input Hsync 0	External input 0 Hsync signal pin
DV0_DATA 23 to DV0_DATA 0	Input	External input video image data 0	External input 0 video image data pin
LCD0_CLK	Output	Panel clock 0	Panel output 0 clock pin
LCD0_DATA 23 to LCD0_DATA 0	Output	Video image data 0 for panel	Panel output 0 video image data pin
LCD0_TCON 6 to LCD0_TCON 0	Output	Control signal 0 for panel	Panel output 0 timing control pin
LCD0_EXTCLK	Input	Panel clock source 0	Panel clock source 0 input pin

29.4 Clocks

There are two clocks to be mainly used by the video display controller 5: the video image clock and pixel clock.

The video image clock is used while the video image is processed in the input controller, passed to the scale-down control block in the scaler, and then written to the buffer (internal bus write control). The DV_CLK clock is used as the video image clock.

The pixel clock is used in graphics read-out processing by the scaler (internal bus read controller) through output controller processing. When the parallel RGB output is selected in the output controller, the frequencies of the pixel clock and panel clock (LCD_CLK) are the same. The panel clock can be selected from the video clock, LCD_EXTCLK, and peripheral bus clock 1 (P1φ) with SYSCNT_PANEL_CLK.PANEL_ICKSEL[1:0] of the system controller. When the serial RGB output (3/4 speed mode) is selected in the output controller, the pixel clock frequency is 1/3 or 1/4 the panel clock (LCD_CLK) frequency.

chanhvinh.com

29.5 Hsync and Vsync Signals

Hsync and Vsync signals to be used in the logic stage following the scale-up control block of the scaler are generated by the synchronization control block of the scaler. Since the Hsync and Vsync signals are used as the reference signals for the LCD TCON, which generates various panel driving timings, they are also the reference signals for the control signals (LCD_TCON6 to LCD_TCON0 pins) passed to the panel.

The output Hsync signal always operates at a free-running frequency, and the horizontal period is set with SC_SCL0_FRC4.SC_RES_FH[10:0]. On the other hand, the output Vsync signal is selected from the external input and free-running Vsync signals with SC_SCL0_FRC3.SC_RES_VS_SEL of the scaler.

29.5.1 External Input Vsync

(1) Operation Outline

In this mode, the output Vsync signal is generated according to an external input Vsync signal. When displaying video image input from a digital pin on the panel and the pointer buffer is not in use, always use this mode. However, the output Hsync signal is free running even in this mode. Figure 29.4 shows the timing of external input Vsync signal.

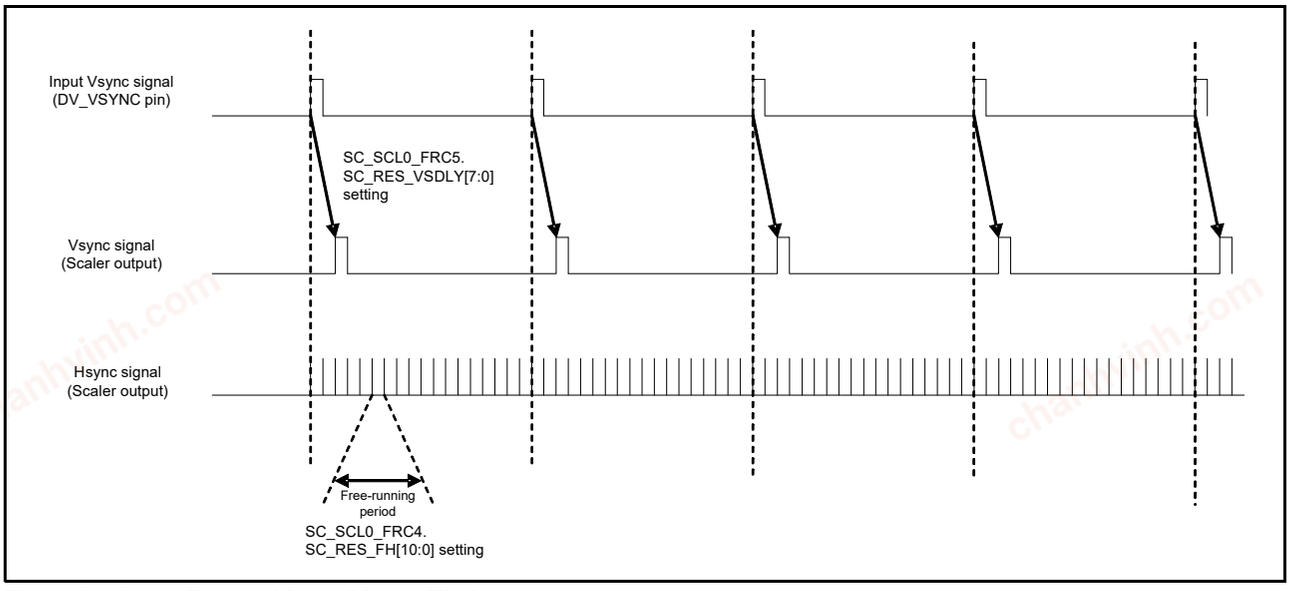


Figure 29.4 External Input Vsync Timing

chanhvinh.com

(2) Notes

When Vsync is externally input, generation of the output Vsync signal is based on the external Vsync signal. That is, the output Vsync signal follows the input Vsync signal, so if an unstable Vsync signal is input, the output Vsync signal will also be unstable.

Since the output Hsync signal is generated according to the frequency generated by a free-running clock and the Vsync signal is generated from the video input as a base, the signals will not be in synchronization. This module adjusts the timing between these signals by adjusting the output Vsync signal so that it stays in time with the output Hsync signal. Therefore, even if the input Vsync signal is stable, the timing of the output Vsync signal may be increased or decreased by up to one line to stay in synchronization with the timing of the output Hsync signal.

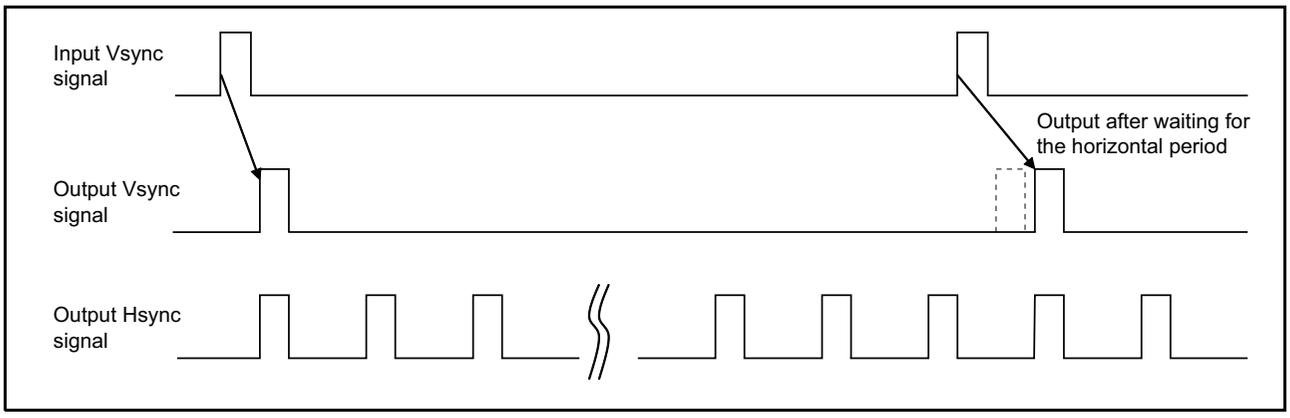


Figure 29.5 Detailed Timing Chart for Generation of the Output Vsync Signal

29.5.2 Free-Running Vsync

(1) Operation Outline

In this mode, the Vsync signal is generated according to the pixel clock (free running). The vertical period is selected with SC_SCL0_FRC4.SC_RES_FV[10:0]. The output Hsync signal is also free running. Figure 29.6 shows the timing.

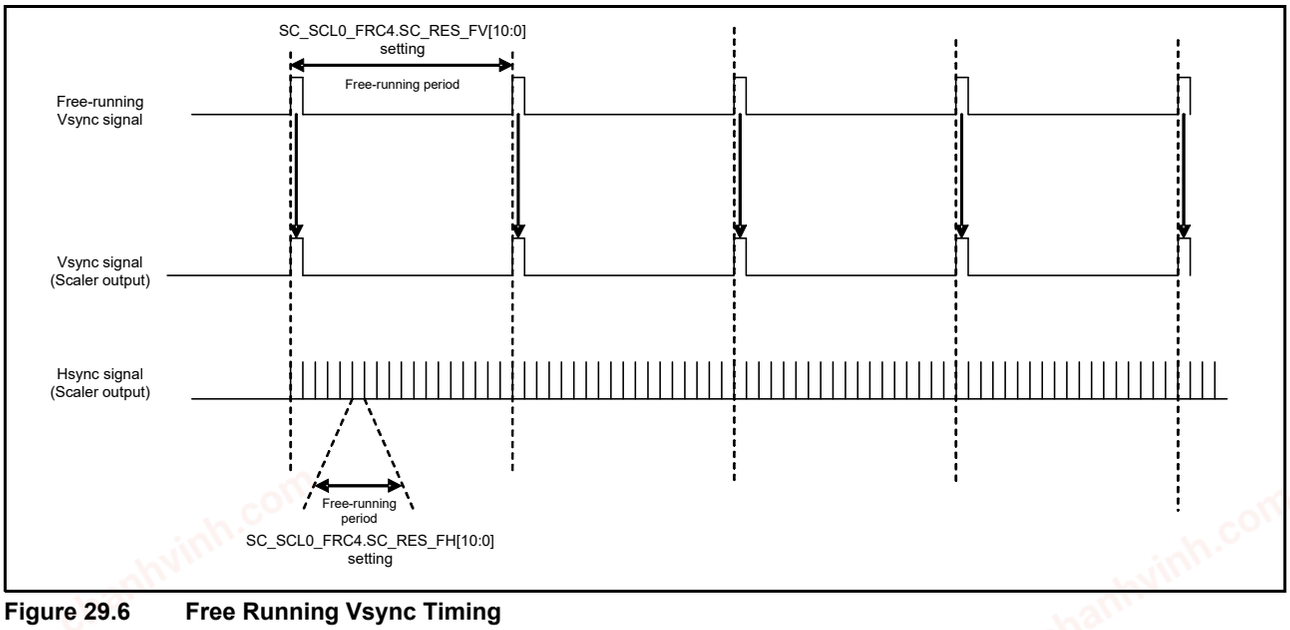


Figure 29.6 Free Running Vsync Timing

chanhvinh.com

(2) Pointer Buffers

In free-running vertical synchronization mode, output of the input video image to a panel may lead to flicker in the output video image. This occurs when the input and output vertical sync signals are not in synchronization. To prevent this, use the pointer buffers to adjust the timing between the input and output video images in frame units. If the input Vsync signal is faster than the output Vsync signal, frames from the input video image are skipped in the output image for display. On the other hand, when the input Vsync signal is slower than the output Vsync signal, frames from the input video image are repeated. However, when the difference in timing between the input and output Vsync signals is too large, the pointer buffers cannot deal with the difference, so flicker may occur. As more buffers are used, the pointer buffers can deal with larger differences in frequency.

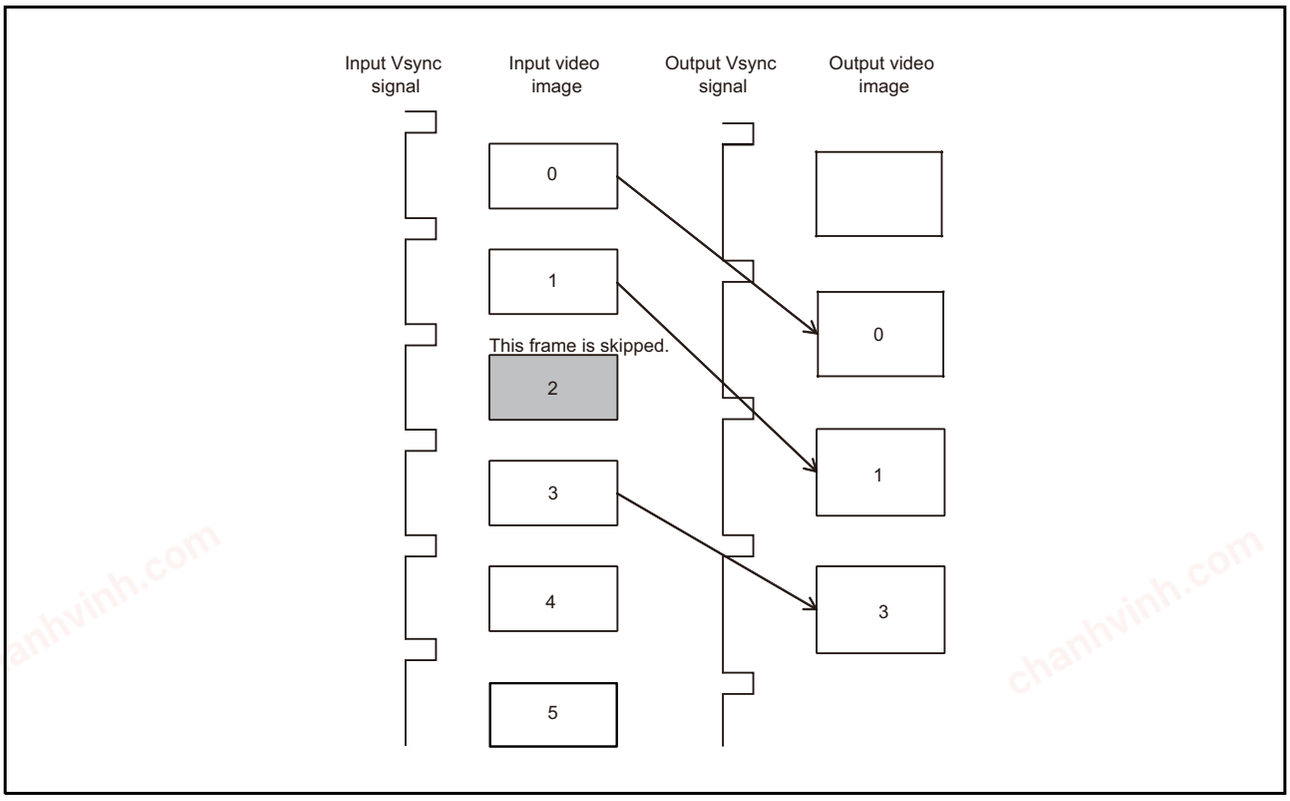


Figure 29.7 Timing when the Input Vsync Signal is Faster than the Output Vsync Signal

chanhvinh.com

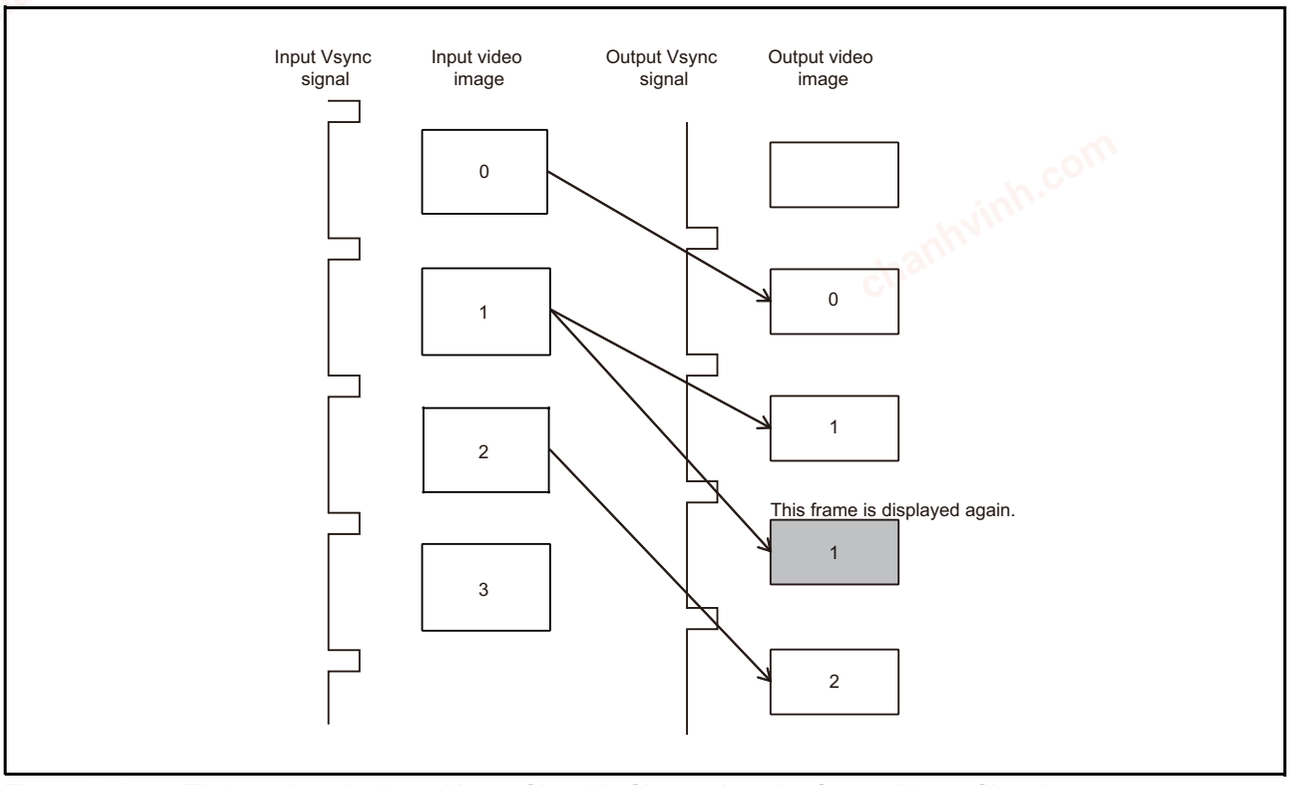


Figure 29.8 Timing when the Input Vsync Signal is Slower than the Output Vsync Signal

29.5.3 Usage Note on Changing Vsync Signal Selections

When the Vsync signal selection is changed, the output Vsync signal is discontinuous, resulting in disordered panel display. In this case, perform the mute processing according to the panel specification as necessary and change the Vsync signal selection.

chanhvinh.com

30. Video Display Controller 5 (2): Input Controller

30.1 Input Controller Functions

30.1.1 Overview of Functions

The input controller performs the on/off control of input supplied via the external input pins, and subjects the signals to synchronization adjustment, horizontal noise reduction, and brightness adjustment, gain adjustment, and YCbCr ↔ GBR conversion using a color matrix.

The functional block diagram of the input controller is shown below.

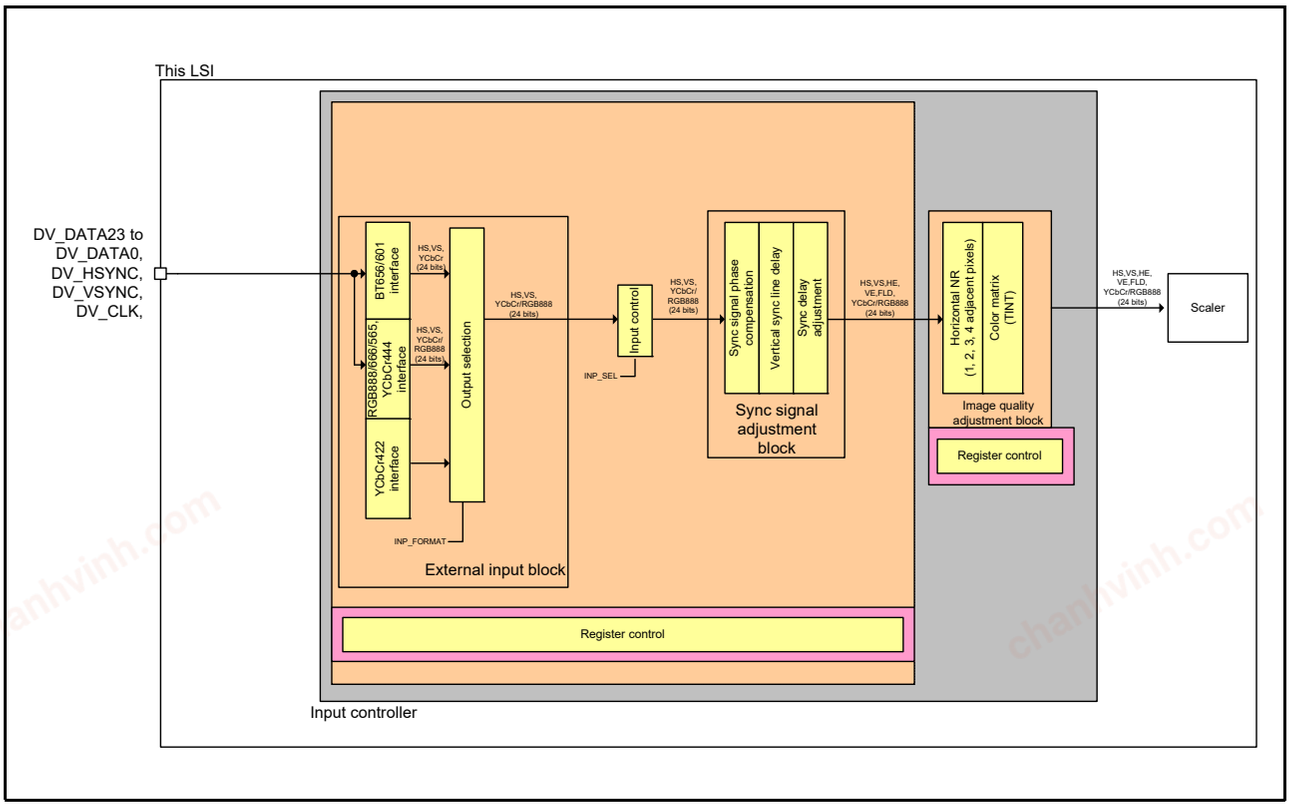


Figure 30.1 Functional Block Diagram of Input Controller

chanhvinh.com

30.1.2 Updating Registers of External Signal Input Block and Sync Signal Adjustment Block

The control registers of the external input block and sync signal adjustment block are updated by setting the relevant update control bit to 1.

For the control registers other than the IMGCNT_DRC_REG register of the image quality adjustment block, the update timing is controlled using the Vsync signal.

After 1 is set to the bits in the update control register, the contents of the relevant registers are actually modified at the rising edge of the Vsync signal, when the update control register is automatically cleared to 0.

Table 30.1 Register Update Control

Register Name	Bit Name	Initial Value	Description
INP_UPDATE	INP_EXT_UPDATE	0	External Input Block Register Update 0: Registers are not updated. 1: Registers are updated.
INP_UPDATE	INP_IMG_UPDATE	0	Sync Signal Adjustment Block Register Update 0: Registers are not updated. 1: Registers are updated.
IMGCNT_UPDATE	IMGCNT_VEN	0	Image Quality Adjustment Block Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync signal.

30.1.3 Controlling Input

The input controller performs the on/off control of input supplied via the external input pins.

Table 30.2 Input Control

Register Name	Bit Name	Initial Value	Description
INP_SEL_CNT	INP_SEL	0	On/Off Control of Input Supplied via External Input Pin 0: Input supplied via the external input pins is off. 1: Input supplied via the external input pins is on.

chanhvinh.com

30.1.4 Controlling Externally Input Video Signals

The externally input video image signals in the YCbCr444, RGB888, RGB666, RGB565, BT656 (extended), BT601 (extended), YCbCr422 (the 16-bit data format of the BT601 standard) formats can be handled.

The BT656 signals can be used for the 525-line and 59.94-Hz (27.0-MHz) or the 625-line and 50.00-Hz (27.0-MHz) interlace signals and for the 525-line and 59.94-Hz (54.0-MHz) or the 625-line and 50.00-Hz (54.0-MHz) BT656-extended progressive signals.

The BT601 signals can be used for the 8-bit data line 525-line and 59.94 Hz (27.0-MHz) or the 625-line and 50.00-Hz (27.0-MHz) interlace signals and for the 525-line and 59.94 Hz (54.0-MHz) or the 625-line and 50.00-Hz (54.0-MHz) extended progressive signals.

The YCbCr422 signals can be used for the 16-bit data line 525-line and 59.94-Hz (13.5-MHz) or the 625-line and 50.00-Hz (13.5-MHz) BT601 interlace signals.

The above signals can be selected by the INP_FORMAT[2:0] bits. Bit endian change and B/R signal swap are controlled by setting the INP_ENDIAN_ON and INP_SWAP_ON bits.

Table 30.3 Externally Input Video Signal Control

Register Name	Bit Name	Initial Value	Description
INP_SEL_CNT	INP_FORMAT[2:0]	000	External Input Format Select 0: YCbCr444, RGB888 1: RGB666 2: RGB565 3: BT656 4: BT601 5: YCbCr422 6, 7: Setting prohibited
INP_EXT_ SYNC_CNT	INP_ENDIAN_ON	0	External Input Bit Endian Change On/Off Control 0: Off 1: On
INP_EXT_ SYNC_CNT	INP_SWAP_ON	0	External Input B/R Signal Swap On/Off Control 0: Off 1: On

chanhvinh.com

30.1.5 Selecting Clock Edge for Externally Input Signals

The clock edge for receiving the video image signals, Vsync signals, and Hsync signals is individually selected with the INP_PXD_EDGE, INP_VS_EDGE, INP_HS_EDGE bits.

Table 30.4 Externally Input Clock Edge Selection

Register Name	Bit Name	Initial Value	Description
INP_SEL_CNT	INP_PXD_EDGE	0	Clock Edge Select for Capturing External Input Video Image Signals DV_DATA23 to DV_DATA0 0: Rising edge 1: Falling edge
INP_SEL_CNT	INP_VS_EDGE	0	Clock Edge Select for Capturing External Input Vsync Signal DV_VSYNC 0: Rising edge 1: Falling edge
INP_SEL_CNT	INP_HS_EDGE	0	Clock Edge Select for Capturing External Input Hsync Signal DV_HSYNC 0: Rising edge 1: Falling edge

Figure 30.2 shows the typical input timing of externally input signals.

The input signals can be received at the rising edge of the clock signal DV_CLK when the INP_PXD_EDGE, INP_VS_EDGE, and INP_ES_EDGE bits are 0.

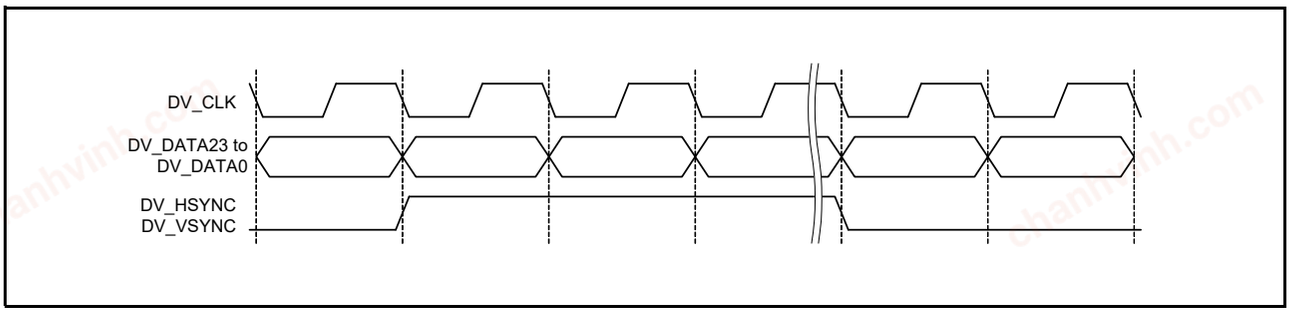


Figure 30.2 Typical Input Timing of Externally Input Signals (Clock Phase)

30.1.6 Externally Input Sync Signal Inversion Control

Inversion of polarity of the Vsync and Hsync signals can be controlled by the INP_VS_INV and INP_HS_INV bits.

Table 30.5 Sync Signal Inversion Control

Register Name	Bit Name	Initial Value	Description
INP_EXT_SYNC_CNT	INP_VS_INV	0	External Input Vsync Signal DV_VSYNC Inversion Control 0: Not inverted (positive polarity) 1: Inverted (negative polarity)
INP_EXT_SYNC_CNT	INP_HS_INV	0	External Input Hsync Signal DV_HSYNC Inversion Control 0: Not inverted (positive polarity) 1: Inverted (negative polarity)

chanhvinh.com

30.1.7 Bit Allocation of Externally Input Video Image Signals

Allocation of the externally input video image signal pins DV_DATA to the signals in each format is described below.

(1) YCbCr444/RGB888 Input

When the external input is of YCbCr444/RGB888 format, the video image signal pins DV_DATA are allocated to the internal signals Y/GOUT, Cb/BOUT, Cr/ROUT, as shown in Table 30.6.

Table 30.6 Bit Allocation of DV_DATA Pin Inputs when the External Input is of YCbCr444/RGB888

INP_FORMAT[2:0]	**0**	**0**	**0**	**0**
INP_ENDIAN_ON	**0**	**0**	**1**	**1**
INP_SWAP_ON	**0**	**1**	**0**	**1**
DV_DATA23	Cr/ROUT[7]	Cb/BOUT[7]	Cr/ROUT[0]	Cb/BOUT[0]
DV_DATA22	Cr/ROUT[6]	Cb/BOUT[6]	Cr/ROUT[1]	Cb/BOUT[1]
DV_DATA21	Cr/ROUT[5]	Cb/BOUT[5]	Cr/ROUT[2]	Cb/BOUT[2]
DV_DATA20	Cr/ROUT[4]	Cb/BOUT[4]	Cr/ROUT[3]	Cb/BOUT[3]
DV_DATA19	Cr/ROUT[3]	Cb/BOUT[3]	Cr/ROUT[4]	Cb/BOUT[4]
DV_DATA18	Cr/ROUT[2]	Cb/BOUT[2]	Cr/ROUT[5]	Cb/BOUT[5]
DV_DATA17	Cr/ROUT[1]	Cb/BOUT[1]	Cr/ROUT[6]	Cb/BOUT[6]
DV_DATA16	Cr/ROUT[0]	Cb/BOUT[0]	Cr/ROUT[7]	Cb/BOUT[7]
DV_DATA15	Y/GOUT[7]	Y/GOUT[7]	Y/GOUT[0]	Y/GOUT[0]
DV_DATA14	Y/GOUT[6]	Y/GOUT[6]	Y/GOUT[1]	Y/GOUT[1]
DV_DATA13	Y/GOUT[5]	Y/GOUT[5]	Y/GOUT[2]	Y/GOUT[2]
DV_DATA12	Y/GOUT[4]	Y/GOUT[4]	Y/GOUT[3]	Y/GOUT[3]
DV_DATA11	Y/GOUT[3]	Y/GOUT[3]	Y/GOUT[4]	Y/GOUT[4]
DV_DATA10	Y/GOUT[2]	Y/GOUT[2]	Y/GOUT[5]	Y/GOUT[5]
DV_DATA9	Y/GOUT[1]	Y/GOUT[1]	Y/GOUT[6]	Y/GOUT[6]
DV_DATA8	Y/GOUT[0]	Y/GOUT[0]	Y/GOUT[7]	Y/GOUT[7]
DV_DATA7	Cb/BOUT[7]	Cr/ROUT[7]	Cb/BOUT[0]	Cr/ROUT[0]
DV_DATA6	Cb/BOUT[6]	Cr/ROUT[6]	Cb/BOUT[1]	Cr/ROUT[1]
DV_DATA5	Cb/BOUT[5]	Cr/ROUT[5]	Cb/BOUT[2]	Cr/ROUT[2]
DV_DATA4	Cb/BOUT[4]	Cr/ROUT[4]	Cb/BOUT[3]	Cr/ROUT[3]
DV_DATA3	Cb/BOUT[3]	Cr/ROUT[3]	Cb/BOUT[4]	Cr/ROUT[4]
DV_DATA2	Cb/BOUT[2]	Cr/ROUT[2]	Cb/BOUT[5]	Cr/ROUT[5]
DV_DATA1	Cb/BOUT[1]	Cr/ROUT[1]	Cb/BOUT[6]	Cr/ROUT[6]
DV_DATA0	Cb/BOUT[0]	Cr/ROUT[0]	Cb/BOUT[7]	Cr/ROUT[7]

chanhvinh.com

(2) RGB666 Input

When the external input is of RGB666 format, the video image signal pins DV_DATA are allocated to the internal signals GOUT, BOUT, ROUT as shown in Table 30.7.

The internal signals GOUT, BOUT, ROUT to which the video image signal pins DV_DATA are allocated are output as a 24-bit video image from the RGB666 interface with the following formulae.

G[7:0] = GOUT[7:2] × 255 ÷ 63

B[7:0] = BOUT[7:2] × 255 ÷ 63

R[7:0] = ROUT[7:2] × 255 ÷ 63

Table 30.7 Bit Allocation of DV_DATA Pin Inputs When the External Input is of RGB666

INP_FORMAT[2:0]	**1**	**1**	**1**	**1**
INP_ENDIAN_ON	**0**	**0**	**1**	**1**
INP_SWAP_ON	**0**	**1**	**0**	**1**
DV_DATA17	ROUT[7]	BOUT[7]	ROUT[2]	BOUT[2]
DV_DATA16	ROUT[6]	BOUT[6]	ROUT[3]	BOUT[3]
DV_DATA15	ROUT[5]	BOUT[5]	ROUT[4]	BOUT[4]
DV_DATA14	ROUT[4]	BOUT[4]	ROUT[5]	BOUT[5]
DV_DATA13	ROUT[3]	BOUT[3]	ROUT[6]	BOUT[6]
DV_DATA12	ROUT[2]	BOUT[2]	ROUT[7]	BOUT[7]
DV_DATA11	GOUT[7]	GOUT[7]	GOUT[2]	GOUT[2]
DV_DATA10	GOUT[6]	GOUT[6]	GOUT[3]	GOUT[3]
DV_DATA9	GOUT[5]	GOUT[5]	GOUT[4]	GOUT[4]
DV_DATA8	GOUT[4]	GOUT[4]	GOUT[5]	GOUT[5]
DV_DATA7	GOUT[3]	GOUT[3]	GOUT[6]	GOUT[6]
DV_DATA6	GOUT[2]	GOUT[2]	GOUT[7]	GOUT[7]
DV_DATA5	BOUT[7]	ROUT[7]	BOUT[2]	ROUT[2]
DV_DATA4	BOUT[6]	ROUT[6]	BOUT[3]	ROUT[3]
DV_DATA3	BOUT[5]	ROUT[5]	BOUT[4]	ROUT[4]
DV_DATA2	BOUT[4]	ROUT[4]	BOUT[5]	ROUT[5]
DV_DATA1	BOUT[3]	ROUT[3]	BOUT[6]	ROUT[6]
DV_DATA0	BOUT[2]	ROUT[2]	BOUT[7]	ROUT[7]

chanhvinh.com

(3) RGB565 Input

When the external input is of RGB565 format, the video image signal pins DV_DATA are allocated to the internal signals GOUT, BOUT, ROUT as shown in Table 30.8.

The internal signals GOUT, BOUT, ROUT to which the video image signal pins DV_DATA are allocated are output as a 24-bit video image from the RGB565 interface with the following formulae.

G[7:0] = GOUT[7:2] × 255 ÷ 63

B[7:0] = BOUT[7:3] × 255 ÷ 31

R[7:0] = ROUT[7:3] × 255 ÷ 31

Table 30.8 Bit Allocation of DV_DATA Pin Inputs When the External Input is of RGB565

INP_FORMAT[2:0]	**2**	**2**	**2**	**2**
INP_ENDIAN_ON	**0**	**0**	**1**	**1**
INP_SWAP_ON	**0**	**1**	**0**	**1**
DV_DATA15	ROUT[7]	BOUT[7]	ROUT[3]	BOUT[3]
DV_DATA14	ROUT[6]	BOUT[6]	ROUT[4]	BOUT[4]
DV_DATA13	ROUT[5]	BOUT[5]	ROUT[5]	BOUT[5]
DV_DATA12	ROUT[4]	BOUT[4]	ROUT[6]	BOUT[6]
DV_DATA11	ROUT[3]	BOUT[3]	ROUT[7]	BOUT[7]
DV_DATA10	GOUT[7]	GOUT[7]	GOUT[2]	GOUT[2]
DV_DATA9	GOUT[6]	GOUT[6]	GOUT[3]	GOUT[3]
DV_DATA8	GOUT[5]	GOUT[5]	GOUT[4]	GOUT[4]
DV_DATA7	GOUT[4]	GOUT[4]	GOUT[5]	GOUT[5]
DV_DATA6	GOUT[3]	GOUT[3]	GOUT[6]	GOUT[6]
DV_DATA5	GOUT[2]	GOUT[2]	GOUT[7]	GOUT[7]
DV_DATA4	BOUT[7]	ROUT[7]	BOUT[3]	ROUT[3]
DV_DATA3	BOUT[6]	ROUT[6]	BOUT[4]	ROUT[4]
DV_DATA2	BOUT[5]	ROUT[5]	BOUT[5]	ROUT[5]
DV_DATA1	BOUT[4]	ROUT[4]	BOUT[6]	ROUT[6]
DV_DATA0	BOUT[3]	ROUT[3]	BOUT[7]	ROUT[7]

chanhvinh.com

(4) BT656/BT601 Input

When the external input is of BT656 or BT601 format, the video image signal pins DV_DATA are allocated to the internal signal BTOUT, as shown in Table 30.9.

The internal signal BTOUT to which the video image signal pins DV_DATA are allocated is expanded to the YCbCr signal.

For expansion to the YCbCr signal, see section 30.1.12, BT656/BT601/YCbCr422 Format Setting.

Table 30.9 Bit Allocation of DV_DATA Pin Inputs When the External Input is of BT656 or BT601

INP_FORMAT[2:0]	3 to 4	3 to 4
INP_ENDIAN_ON	0	1
INP_SWAP_ON	0	0
DV_DATA7	BTOUT[7]	BTOUT[0]
DV_DATA6	BTOUT[6]	BTOUT[1]
DV_DATA5	BTOUT[5]	BTOUT[2]
DV_DATA4	BTOUT[4]	BTOUT[3]
DV_DATA3	BTOUT[3]	BTOUT[4]
DV_DATA2	BTOUT[2]	BTOUT[5]
DV_DATA1	BTOUT[1]	BTOUT[6]
DV_DATA0	BTOUT[0]	BTOUT[7]

(5) YCbCr422 Input

When the external input is of YCbCr422 format, the video image signal pins DV_DATA are allocated to the internal signals Y and Cb/Cr, as shown in Table 30.10.

Table 30.10 Bit Allocation of DV_DATA Pin Inputs When the External Input is of YCbCr422

INP_FORMAT[2:0]	5	5	5	5
INP_ENDIAN_ON	0	0	1	1
INP_SWAP_ON	0	1	0	1
DV_DATA15	Y[7]	Cb/Cr[7]	Y[0]	Cb/Cr[0]
DV_DATA14	Y[6]	Cb/Cr[6]	Y[1]	Cb/Cr[1]
DV_DATA13	Y[5]	Cb/Cr[5]	Y[2]	Cb/Cr[2]
DV_DATA12	Y[4]	Cb/Cr[4]	Y[3]	Cb/Cr[3]
DV_DATA11	Y[3]	Cb/Cr[3]	Y[4]	Cb/Cr[4]
DV_DATA10	Y[2]	Cb/Cr[2]	Y[5]	Cb/Cr[5]
DV_DATA9	Y[1]	Cb/Cr[1]	Y[6]	Cb/Cr[6]
DV_DATA8	Y[0]	Cb/Cr[0]	Y[7]	Cb/Cr[7]
DV_DATA7	Cb/Cr[7]	Y[7]	Cb/Cr[0]	Y[0]
DV_DATA6	Cb/Cr[6]	Y[6]	Cb/Cr[1]	Y[1]
DV_DATA5	Cb/Cr[5]	Y[5]	Cb/Cr[2]	Y[2]
DV_DATA4	Cb/Cr[4]	Y[4]	Cb/Cr[3]	Y[3]
DV_DATA3	Cb/Cr[3]	Y[3]	Cb/Cr[4]	Y[4]
DV_DATA2	Cb/Cr[2]	Y[2]	Cb/Cr[5]	Y[5]
DV_DATA1	Cb/Cr[1]	Y[1]	Cb/Cr[6]	Y[6]
DV_DATA0	Cb/Cr[0]	Y[0]	Cb/Cr[7]	Y[7]

chanhvinh.com

30.1.8 Typical Signal Timing of BT601 Format

Figure 30.3 and Figure 30.4 show the horizontal timings and Figure 30.5 and Figure 30.6 show the vertical timings of the BT601 format.

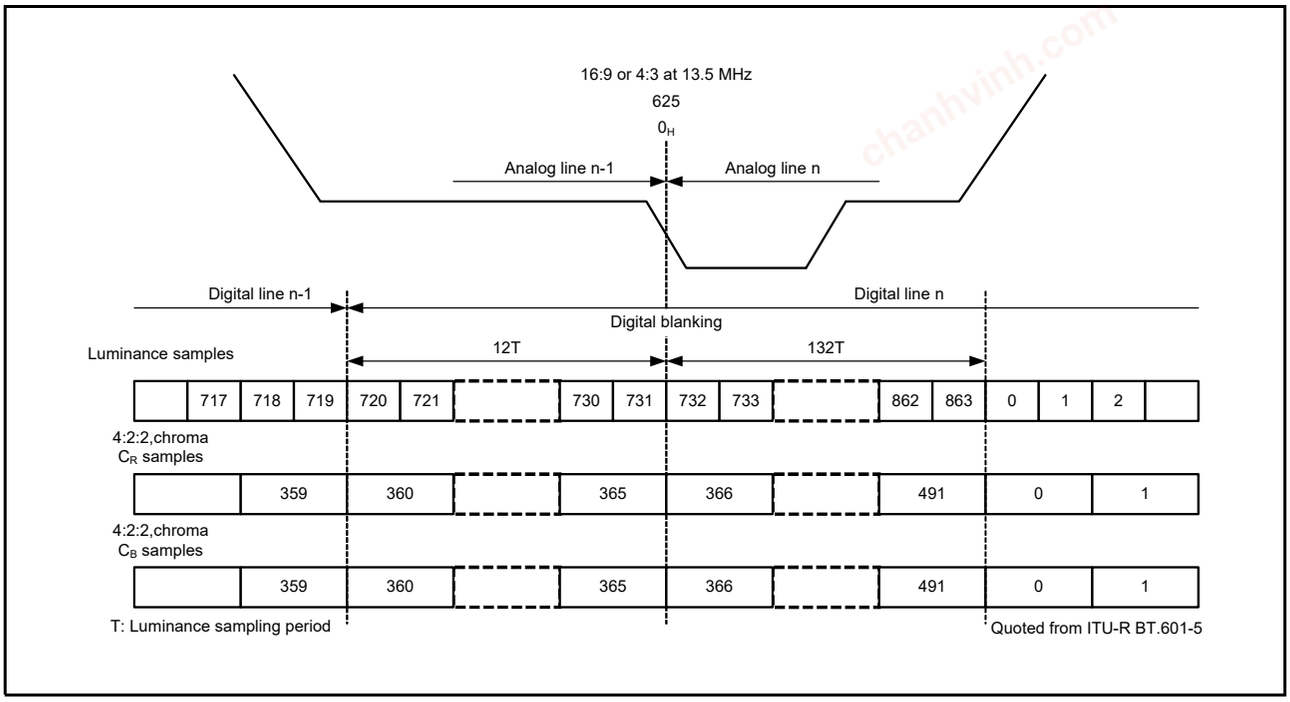


Figure 30.3 BT601 Horizontal Timing (625 Lines/50.00 Hz)

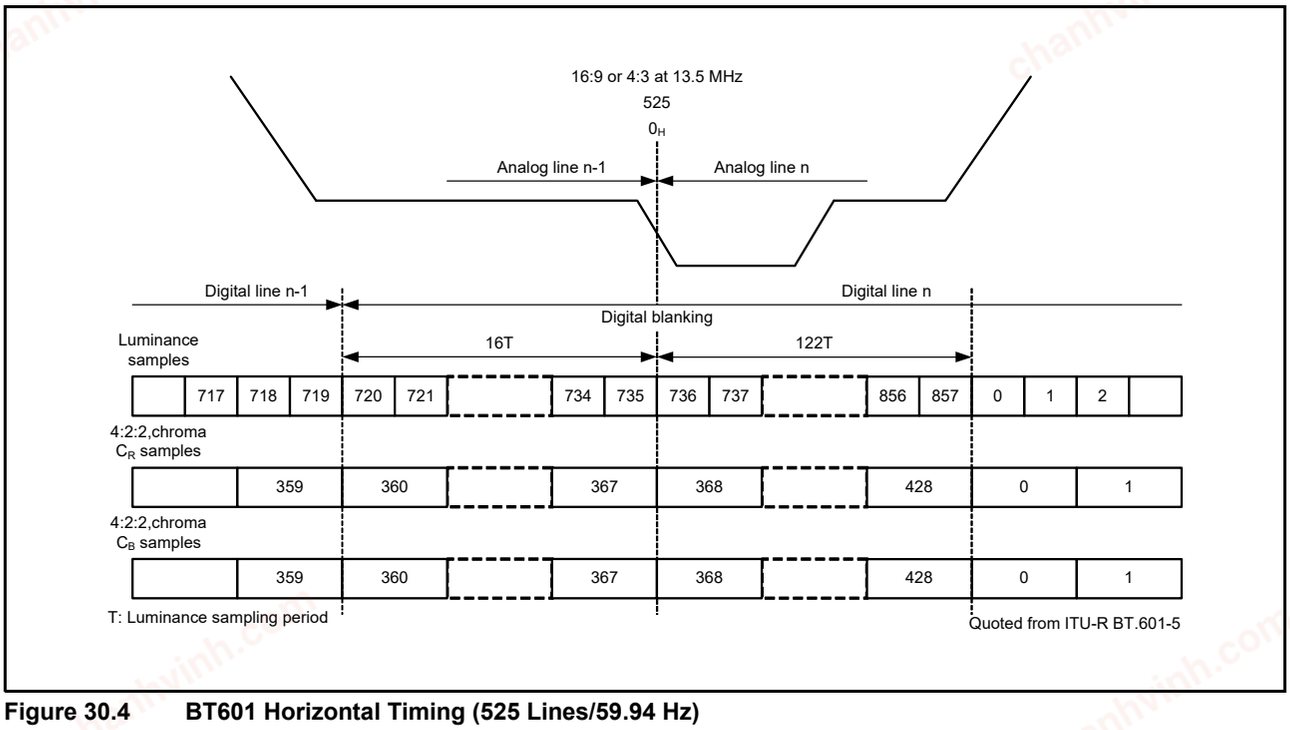


Figure 30.4 BT601 Horizontal Timing (525 Lines/59.94 Hz)

chanhvinh.com

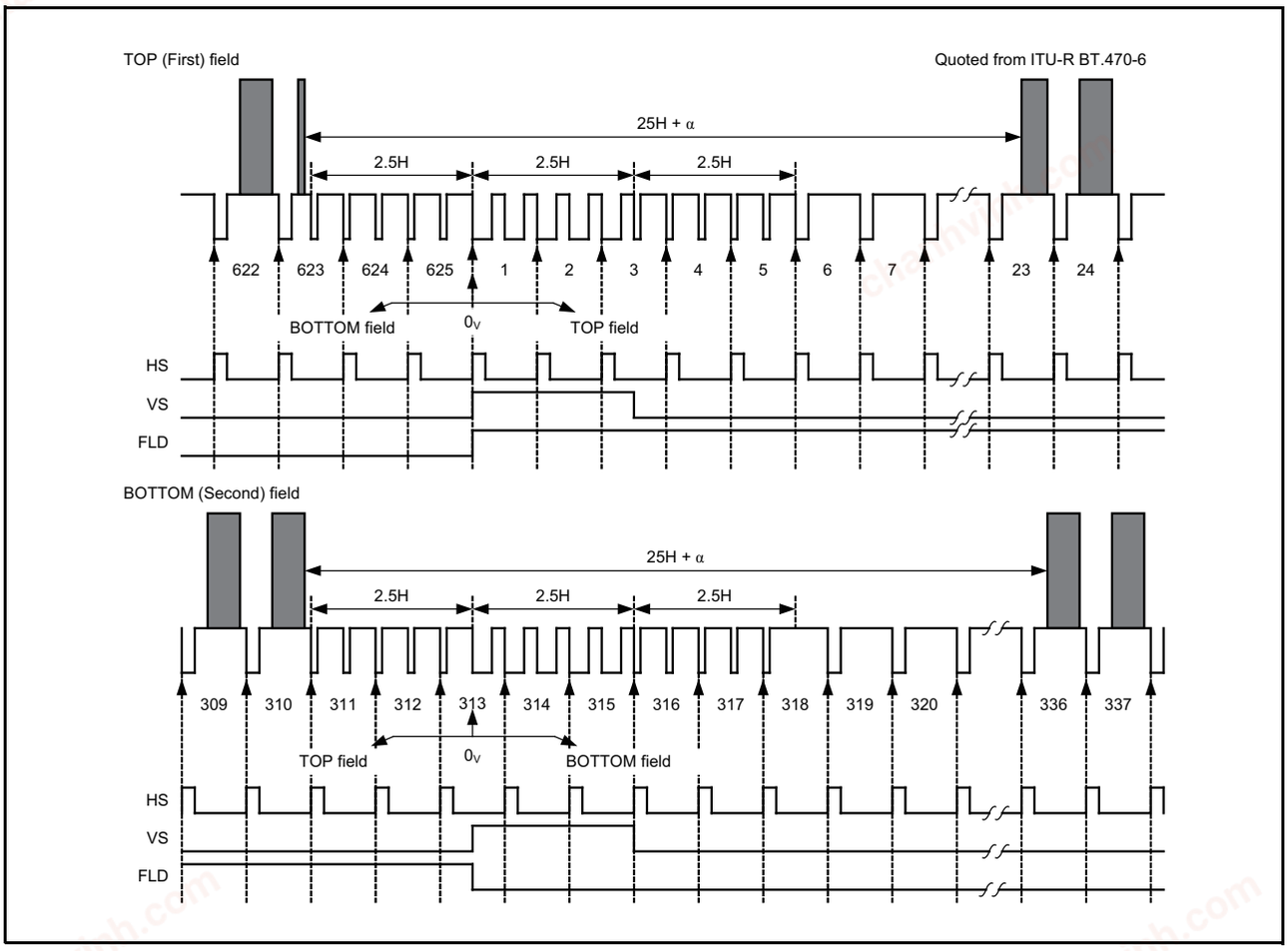


Figure 30.5 BT601 Vertical Timing (625 Lines/50.00 Hz)

chanhvinh.com

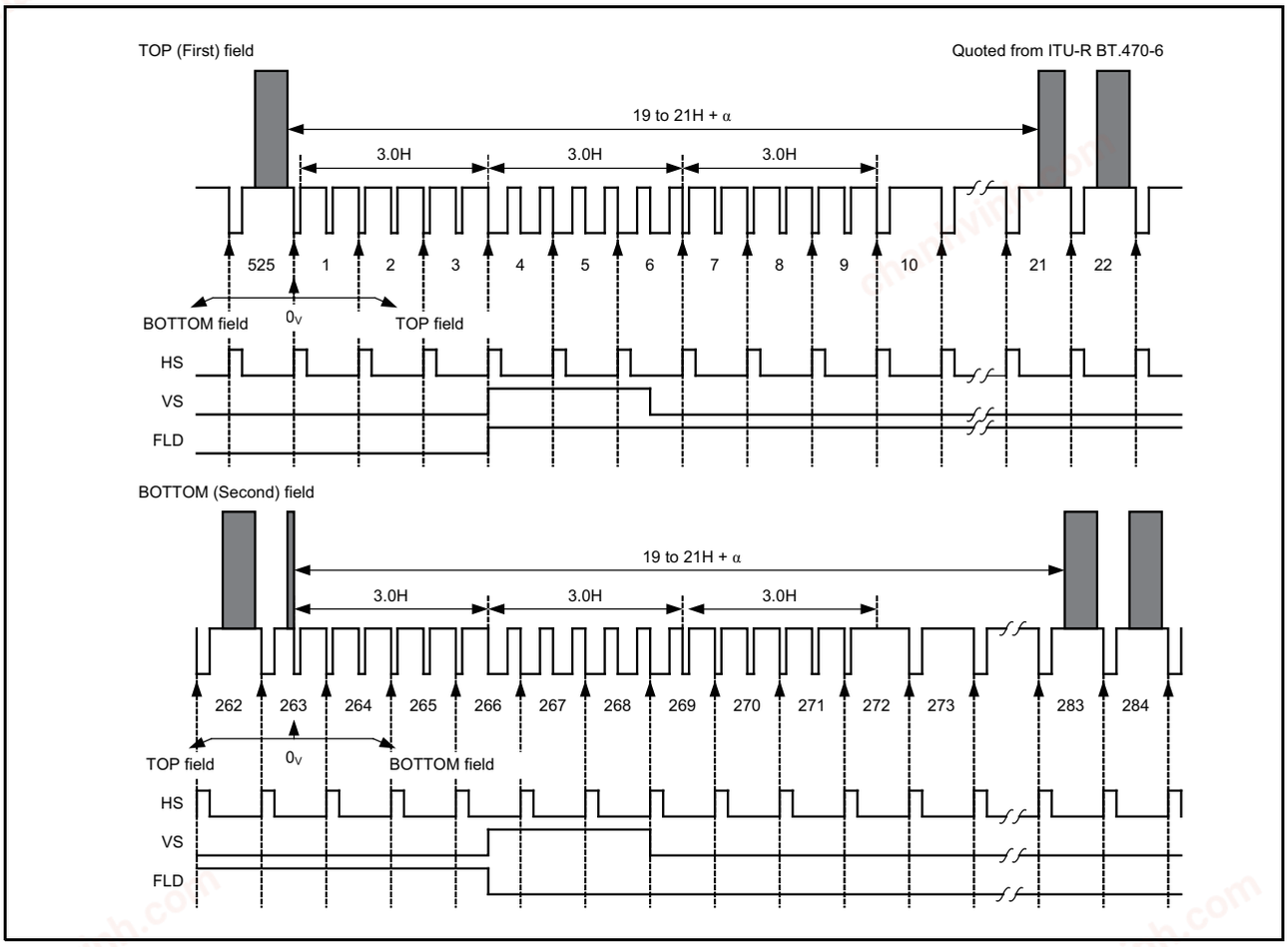


Figure 30.6 BT601 Vertical Timing (525 Lines/59.94 Hz)

chanhvinh.com

30.1.9 Typical Signal Timing of BT656 Format

Figure 30.7 and Figure 30.8 show the horizontal timings of the BT656 format.

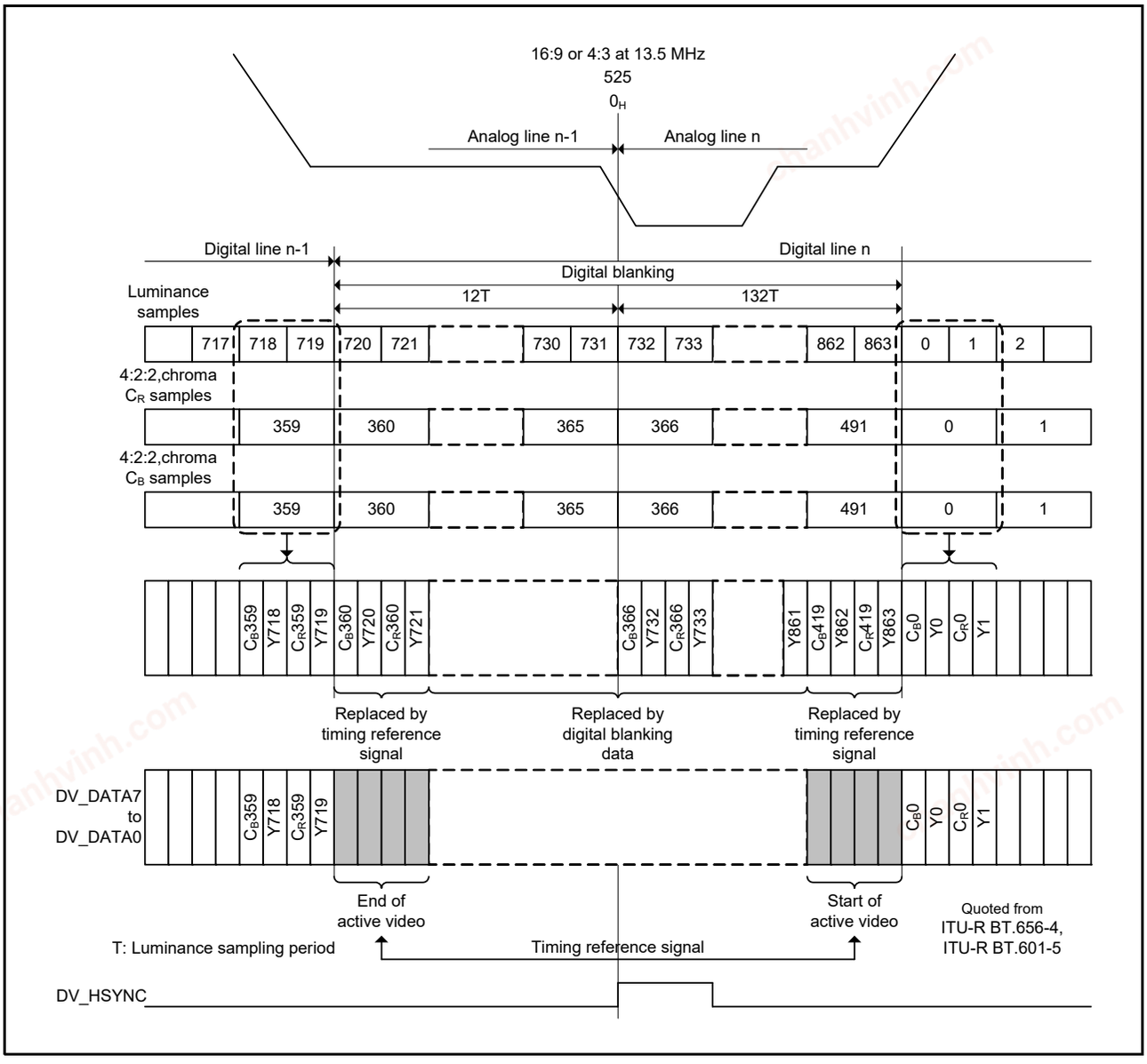


Figure 30.7 BT656 Horizontal Timing (625 Lines/50.00 Hz)

chanhvinh.com

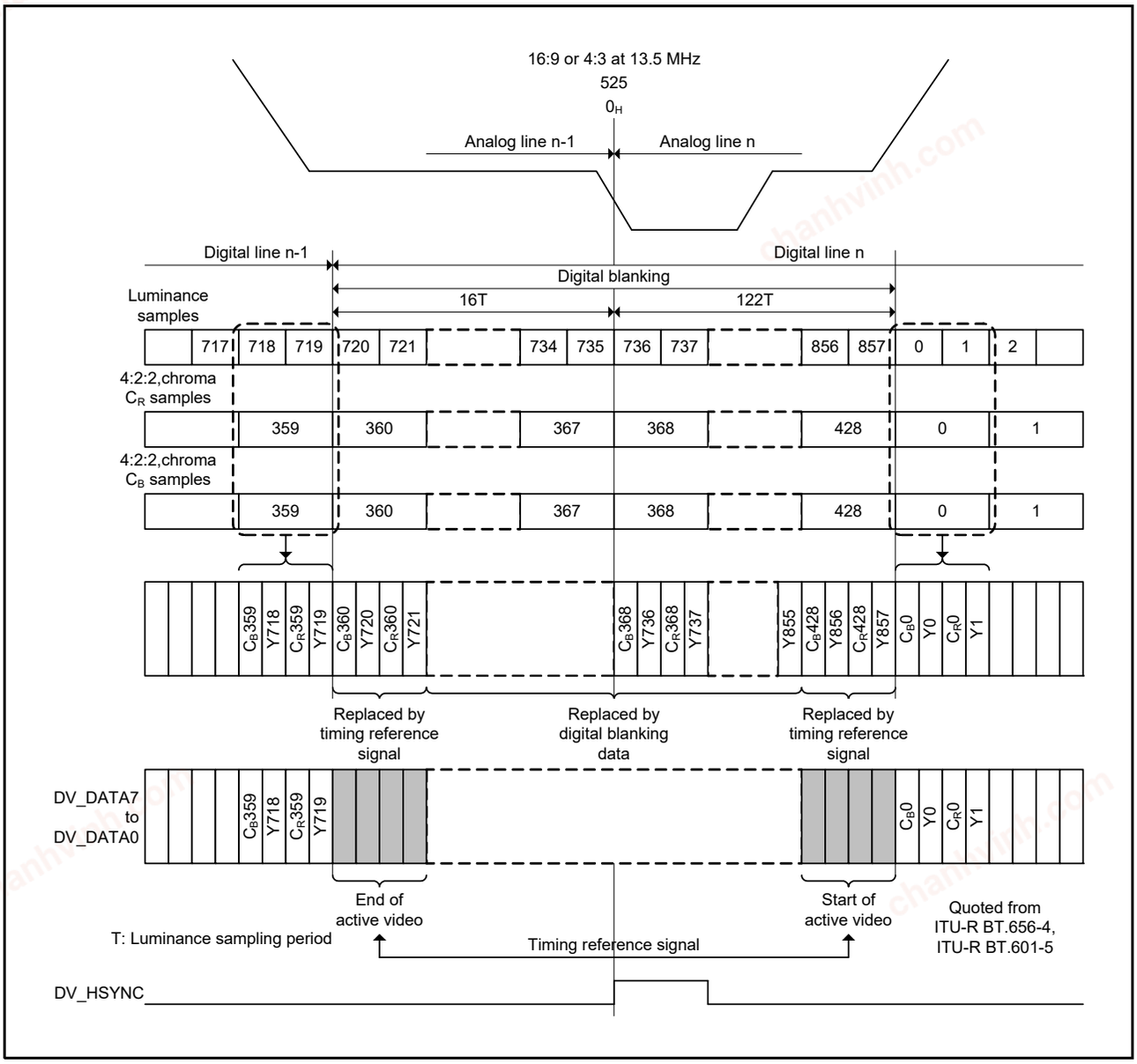


Figure 30.8 BT656 Horizontal Timing (525 Lines/59.94 Hz)

chanhvinh.com

30.1.10 SAV/EAV Code in BT656 Format

Table 30.11 shows the timing of inserting the SAV/EAV code in the BT656 format. Bit information is shown in Table 30.12 and Table 30.13. This module does not refer to the parity bits P3 to P0 shown in Table 30.13.

Table 30.11 SAV/EAV Code Insertion Timing (Line)

		625	525
V-digital field blanking			
Field 1	Start (V = 1)	Line 624	Line 1
	Finish (V = 0)	Line 23	Line 20
Field 2	Start (V = 1)	Line 311	Line 264
	Finish (V = 0)	Line 336	Line 283
V-digital field blanking			
Field 1	F = 0	Line 1	Line 4
Field 2	F = 1	Line 313	Line 266

Table 30.12 SAV/EAV Code Bit Information (1)

Data Bit Number	1st Word (FF)	2nd Word (00)	3rd Word (00)	4th Word (XY)
7 (MSB)	1	0	0	1
6	1	0	0	F
5	1	0	0	V
4	1	0	0	H
3	1	0	0	P3
2	1	0	0	P2
1	1	0	0	P1
0	1	0	0	P0

[Legend]
F = 0 during field 1
F = 1 during field 2
V = 0 elsewhere
V = 1 during field blanking
H = 0 is SAV
H = 1 is EAV

chanhvinh.com

Table 30.13 SAV/EAV Code Bit Information (2)

F	V	H	P3	P2	P1	P0
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	1	1
1	0	1	1	0	1	0
1	1	0	1	1	0	0
1	1	1	0	0	0	1

Figure 30.9 and Figure 30.10 show the SAV/EAV code tables.

		One Horizontal Period																	
		EAV				H blank	SAV				Valid area								
		1	2	3	4		285	286	287	288	289	290	291	292	...	1725	1726	1727	1728
Field1 (top)	1	FF	00	00	B6		FF	00	00	AB	Digital Blanking Data								
	:	FF	00	00	B6		FF	00	00	AB									
	22	FF	00	00	B6		FF	00	00	AB									
	23	FF	00	00	9D		FF	00	00	80	Cb0	Y0	Cr0	Y1	...	Cb718	Y718	Cr718	Y719
	:	FF	00	00	9D		FF	00	00	80	:	Valid pixel data area							:
	:	FF	00	00	9D		FF	00	00	80	:								:
	:	FF	00	00	9D		FF	00	00	80	:								:
	:	FF	00	00	9D		FF	00	00	80	:								:
	:	FF	00	00	9D		FF	00	00	80	:								:
	310	FF	00	00	9D		FF	00	00	80	Cb0	Y0	Cr0	Y1	...	Cb718	Y718	Cr718	Y719
	311	FF	00	00	B6		FF	00	00	AB	Digital Blanking Data								
	312	FF	00	00	B6		FF	00	00	AB									
Field2 (bottom)	313	FF	00	00	F1		FF	00	00	EC	Digital Blanking Data								
	:	FF	00	00	F1		FF	00	00	EC									
	335	FF	00	00	F1		FF	00	00	EC									
	336	FF	00	00	DA		FF	00	00	C7	Cb0	Y0	Cr0	Y1	...	Cb718	Y718	Cr718	Y719
	:	FF	00	00	DA		FF	00	00	C7	:	Valid pixel data area							:
	:	FF	00	00	DA		FF	00	00	C7	:								:
	:	FF	00	00	DA		FF	00	00	C7	:								:
	:	FF	00	00	DA		FF	00	00	C7	:								:
	:	FF	00	00	DA		FF	00	00	C7	:								:
	623	FF	00	00	DA		FF	00	00	C7	Cb0	Y0	Cr0	Y1	...	Cb718	Y718	Cr718	Y719
	624	FF	00	00	F1		FF	00	00	EC	Digital Blanking Data								
	625	FF	00	00	F1		FF	00	00	EC									

Figure 30.9 SAV/EAV Code in BT656 Format (625 Lines/50.00 Hz)

chanhvinh.com

		One Horizontal Period																	
		EAV				H blank	SAV				Valid area								
		1	2	3	4		273	274	275	276	277	278	279	280	...	1713	1714	1715	1716
Field2	1	FF	00	00	F1		FF	00	00	EC	Digital Blanking Data								
	2	FF	00	00	F1		FF	00	00	EC									
	3	FF	00	00	F1		FF	00	00	EC									
Field1 (top)	4	FF	00	00	B6		FF	00	00	AB	Digital Blanking Data								
	:	FF	00	00	B6		FF	00	00	AB									
	19	FF	00	00	B6		FF	00	00	AB									
	20	FF	00	00	9D		FF	00	00	80	Cb0	Y0	Cr0	Y1	...	Cb718	Y718	Cr718	Y719
	:	FF	00	00	9D		FF	00	00	80	:								:
	:	FF	00	00	9D		FF	00	00	80	:								:
	:	FF	00	00	9D		FF	00	00	80	:	Valid pixel data area							:
	:	FF	00	00	9D		FF	00	00	80	:								:
	:	FF	00	00	9D		FF	00	00	80	:								:
	263	FF	00	00	9D		FF	00	00	80	Cb0	Y0	Cr0	Y1	...	Cb718	Y718	Cr718	Y719
	264	FF	00	00	B6		FF	00	00	AB	Digital Blanking Data								
	265	FF	00	00	B6		FF	00	00	AB									
Field2 (bottom)	266	FF	00	00	F1		FF	00	00	EC	Digital Blanking Data								
	:	FF	00	00	F1		FF	00	00	EC									
	282	FF	00	00	F1		FF	00	00	EC									
	283	FF	00	00	DA		FF	00	00	C7	Cb0	Y0	Cr0	Y1	...	Cb718	Y718	Cr718	Y719
	:	FF	00	00	DA		FF	00	00	C7	:								:
	:	FF	00	00	DA		FF	00	00	C7	:								:
	:	FF	00	00	DA		FF	00	00	C7	:	Valid pixel data area							:
	:	FF	00	00	DA		FF	00	00	C7	:								:
	:	FF	00	00	DA		FF	00	00	C7	:								:
	525	FF	00	00	DA		FF	00	00	C7	Cb0	Y0	Cr0	Y1	...	Cb718	Y718	Cr718	Y719

Figure 30.10 SAV/EAV Code in BT656 Format (525 Lines/59.94 Hz)

chanhvinh.com

30.1.11 BT656 Progressive Format

This product can be connected with devices which output data in the BT656 progressive format. Because the standard for the BT656 format does not include description of output in the progressive format, there is no guarantee that this product is connected with devices which output data in the progressive format. The following description shows how to generate a vertical/horizontal synchronization signal by decoding the SAV/EAV code input via the BT656 interface of this module. Confirm the connection with devises which output data in the BT656 progressive format in accordance with this section.

(1) SAV/EAV Code

The SAV/EAV code consists of four words. When the first word is set to FF and the second and third words are set to 00, timing signals are generated by decoding the value of the fourth word (XY).

For bit information, see Table 30.12 in section 30.1.10, SAV/EAV Code in BT656 Format. This product does not refer to the parity bits (P3 to P0).

(2) Vertical/Horizontal Synchronization Signal

Based on the SAV/EAV code, the vertical/horizontal synchronization signal is generated.

(a) Vertical Synchronization Signal

The vertical synchronization signal is output when the value of the V bit is changed from 0 to 1 in the BT656 format. The timing of the output varies with the setting of INP_EXT_SYNC_CNT.INP_F525_625 setting and the value of the F bit in the BT656 format.

Table 30.14 lists the timing.

Table 30.14 Timing of Delay for Output of Vertical Synchronization Signal

INP_EXT_SYNC_CNT. INP_F525_625	F Bit in BT656 Format	Output Timing	Remark
0: 525 lines	0 (Field 1)	2.5 lines after setting of V bit to 1 is detected	525 lines, vertical synchronization signal for field 2
	1 (Field 2)	3 lines after setting of V bit to 1 is detected	525 lines, vertical synchronization signal for field 1
1: 625 lines	0 (Field 1)	2.5 lines after setting of V bit	625 lines, vertical synchronization signal for field 2
	1 (Field 2)	2 lines after setting of V bit to 1 is detected	625 lines, vertical synchronization signal for field 1

(b) Horizontal Synchronization Signal

Based on the setting of the INP_EXT_SYNC_CNT.INP_H_EDGE_SEL bit, the horizontal synchronization signal is output.

chanhvinh.com

(c) Timing Example of 525-Line Interface Input in BT656 Format

Figure 30.11 and Figure 30.12 show examples of the timing of vertical/horizontal synchronization signal extracted from 525-line interlaced input in the BT656 format.

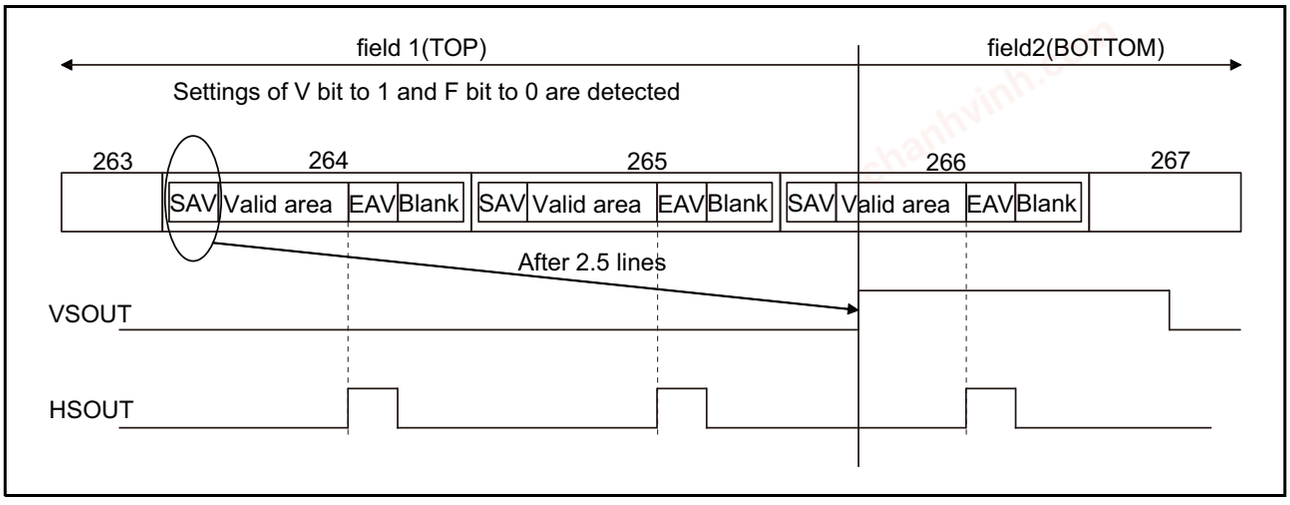


Figure 30.11 Timing of Vertical/Horizontal Synchronization Signal from 525-Line Interlaced Input in BT656 Format (Top to Bottom)

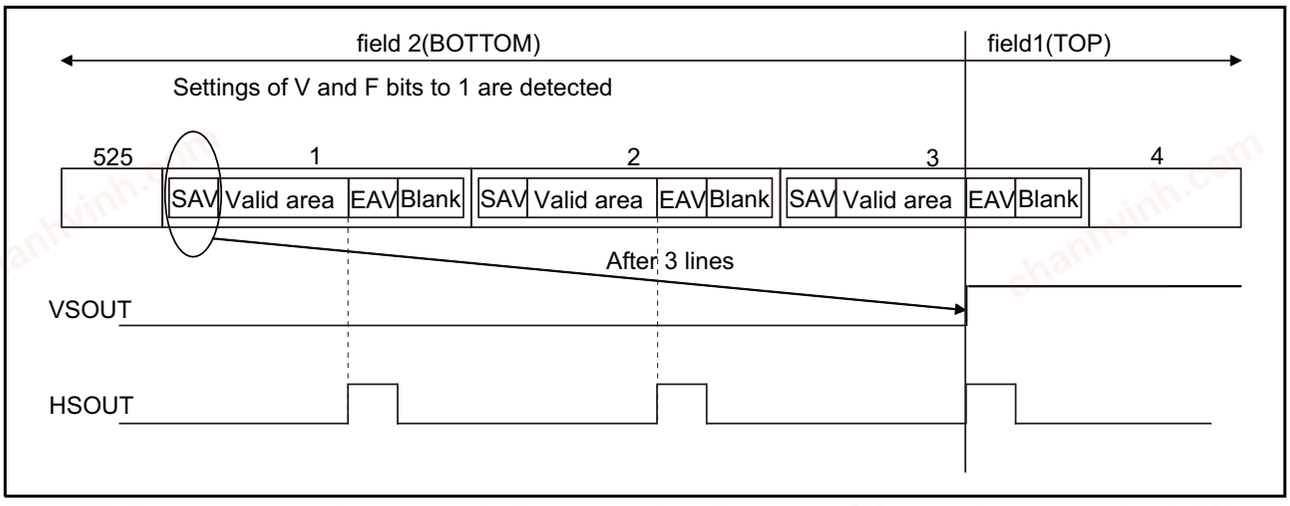


Figure 30.12 Timing of Vertical/Horizontal Synchronization Signal from 525-Line Interlaced Input in BT656 Format (Bottom to Top)

chanhvinh.com

(3) Example of Timing for Progressive Input in BT656 Format

Figure 30.13 shows an example of the SAV/EAV code in 525-line progressive input in the BT656 format. Figure 30.14 shows the vertical/horizontal synchronization signal extracted from 525-line progressive input in the BT656 format. The field is detected as field 1 in this example, because the value of the F bit is set to 0 when that of the V bit is changed from 0 to 1. The filed is regarded as the bottom field. The vertical synchronization signal is output 2.5 lines after the detection of the SAV code.

		One Horizontal Period																
		EAV				H blank	SAV				Valid area							
		1	2	3	4		273	274	275	276	277	278	279	280	...	1713	1714	1715	1716
Field1 (top)	1	FF	00	00	BX		FF	00	00	AX	Digital Blanking Data								
	:	FF	00	00	BX		FF	00	00	AX									
	19	FF	00	00	BX		FF	00	00	AX									
	20	FF	00	00	9X		FF	00	00	8X	Cb0	Y0	Cr0	Y1	...	Cb718	Y718	Cr718	Y719
	:	FF	00	00	9X		FF	00	00	8X	:	Valid pixel data area							:
	:	FF	00	00	9X		FF	00	00	8X	:								:
	:	FF	00	00	9X		FF	00	00	8X	:								:
	:	FF	00	00	9X		FF	00	00	8X	:								:
	:	FF	00	00	9X		FF	00	00	8X	:								:
	504	FF	00	00	9X		FF	00	00	8X	Cb0	Y0	Cr0	Y1	...	Cb718	Y718	Cr718	Y719
	505	FF	00	00	BX		FF	00	00	AX	Digital Blanking Data								
	:	FF	00	00	BX		FF	00	00	AX									
	525	FF	00	00	BX		FF	00	00	AX									

Figure 30.13 SAV/EAV Code in BT656 Progressive Format (525 Lines, 59.94 Hz)

chanhvinh.com

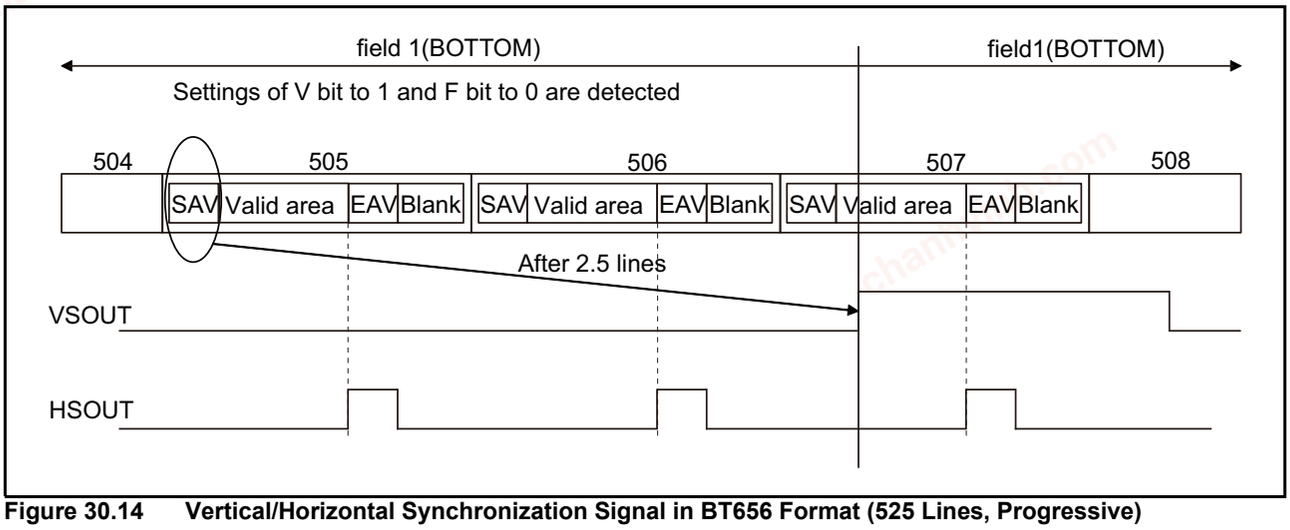


Figure 30.14 Vertical/Horizontal Synchronization Signal in BT656 Format (525 Lines, Progressive)

chanhvinh.com

30.1.12 BT656/BT601/YCbCr422 Format Setting

The BT656 format can be used for the 525-line and 59.94-Hz or the 625-line and 50.00-Hz interlace signal format and progressive signal format (extended).

The BT601 format can be used for the 525-line and 59.94-Hz or the 625-line and 50.00-Hz interlace signal format and progressive signal format (extended).

The YCbCr422 format can be used for the 525-line and 59.94-Hz or the 625-line and 50.00-Hz interlace signal format in the 16-bit data-bus format of the BT601 standard.

The Vsync signal timing for the 525-line BT656 format and 625-line BT656 format are different.

The operating mode is set by the INP_F525_625 bit.

Table 30.15 Operating Mode Setting for BT656 Format

Register Name	Bit Name	Initial Value	Description
INP_EXT_SYNC_CNT	INP_F525_625	0	Number of Lines for BT656 Input of External Input System 0: 525 lines 1: 625 lines

When the interlace signals are to be input in BT656/BT601/YCbCr422 format, half of 2fH phase timings of the Vsync signal and the Hsync signal are set with the INP_FH50[9:0] bits.

The INP_FH50[9:0] bits are also used for the vertical synchronous phase adjustment block. Therefore, for bit description, see Table 30.20.

When the external input is of BT656 format, the reference point of the Hsync signal is set with the INP_H_EDGE_SEL bit.

Table 30.16 Hsync Signal Reference Selection for BT656 Format

Register Name	Bit Name	Initial Value	Description
INP_EXT_SYNC_CNT	INP_H_EDGE_SEL	0	Hsync Signal Reference Select for BT656 Format of External Input System 0: EAV 1: SAV

When the external input is of BT656/BT601 format, the internal signal BTOUT[7:0], which is input from the DV_DATA pins and allocated, is expanded to the 24-bit YCbCr signal.

Expansion timing with respect to the Hsync signal reference is set with the INP_H_POS[1:0] bits.

chanhvinh.com

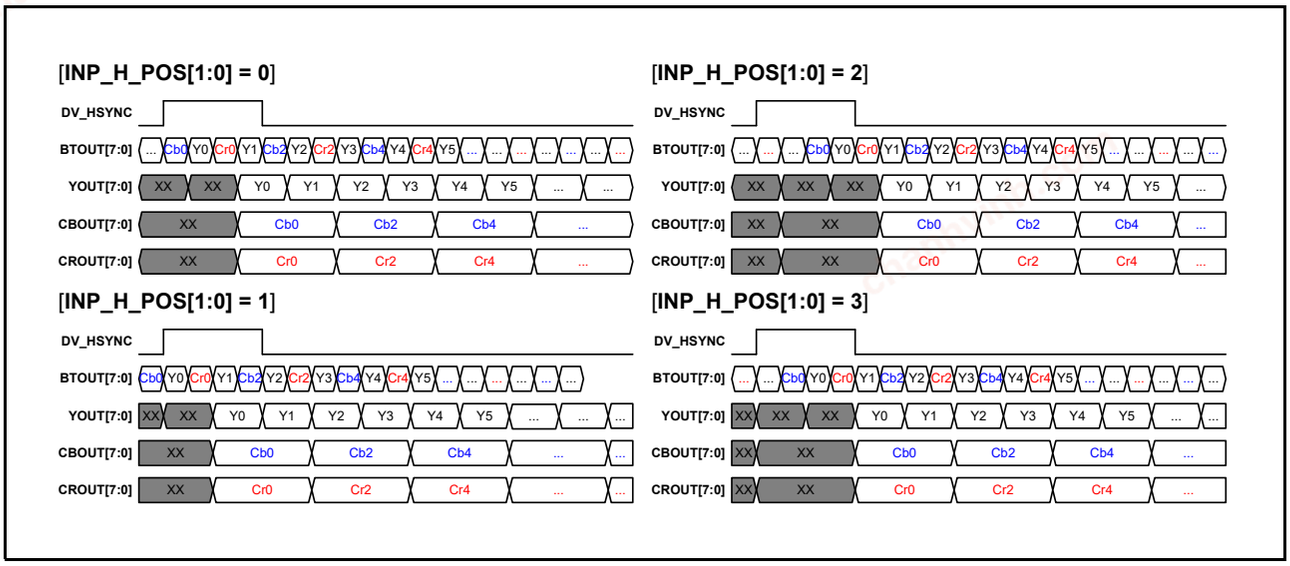


Figure 30.15 YCbCr Data Expansion for BT656/BT601 Input

Table 30.17 Data String Start Timing Selection for BT656/BT601 Input

Register Name	Bit Name	Initial Value	Description
INP_EXT_SYNC_CNT	INP_H_POS[1:0]	0	Y/Cb/Y/Cr Data String Start Timing with respect to Hsync Reference 0: Cb/Y/Cr/Y 1: Y/Cr/Y/Cb 2: Cr/Y/Cb/Y 3: Y/Cb/Y/Cr

When the external input is in YCbCr422 format, the input from the DV_DATA pins is allocated to the internal Y[7:0] and CbCr[7:0] signals, and the CbCr[7:0] are expanded to a 16-bit signal.

Expansion timing with respect to the Hsync signal reference is set with the INP_H_POS[1:0] bits.

chanhvinh.com

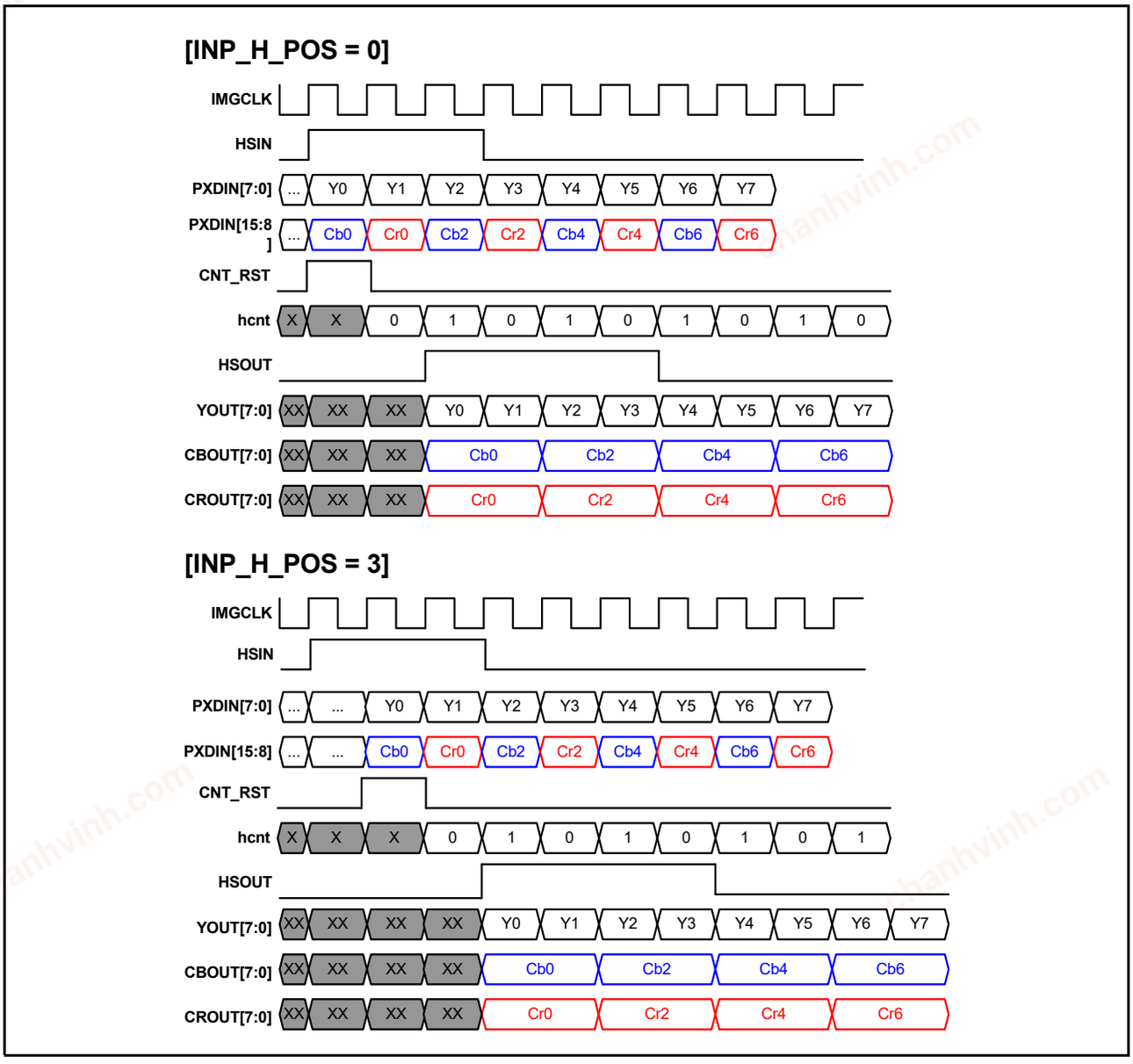


Figure 30.16 YCbCr Data Expansion for YCbCr422 Input

Table 30.18 Data String Start Timing Selection for YCbCr422 Input

Register Name	Bit Name	Initial Value	Description
INP_EXT_SYNC_CNT	INP_H_POS[1:0]	0	Cb/Cr Data String Start Timing with respect to Hsync Reference 0: Cb/Cr 3: Cr/Cb 1, 2: Setting prohibited

chanhvinh.com

30.1.13 YCbCr444/RBG888/666/565 Input Timing

The YCbCr444/RGB888/666/565 format can be used for the progressive YCbCr/RGB signal.

The sync signal width (H_SYNC, V_SYNC), sync signal polarity (H_POL, V_POL), valid period start position (H_BP, V_BP), valid period end position (H_FP, V_FP), and valid period video width (H_ACTIVE, V_ACTIVE) are shown in Table 30.19.

Table 30.19 YCbCr/RGB Signal Reception Timing

Item	Description
External input clock	Maximum external input clock frequency: 87.00 MHz
Vsync signal width (V_SYNC)	Minimum Vsync signal width: 1 CLK
Vsync signal polarity (V_POL)	Positive or negative polarity is selected by the relevant registers.
Vertical valid period start position (V_BP)	From Vsync reference to the head of the video image: 5 lines or more
Vertical valid period video width (V_ACTIVE)	Maximum vertical valid period: 1024 lines
Vertical valid period end position (V_FP)	From the end of the video image to the Vsync reference: 4 lines or more Note 1.
Hsync signal width (H_SYNC)	Minimum Hsync signal width: 1 CLK
Hsync signal polarity (H_POL)	Positive or negative polarity is selected by the relevant registers.
Horizontal valid period start position (H_BP)	From Hsync reference to the head of the video image: 16 CLK or more
Horizontal valid period video width (H_ACTIVE)	Maximum horizontal valid period: 1440 pixels
Horizontal valid period end position (H_FP)	From the end of the video image to the Hsync reference: 16 CLK or more Note 2.
Number of vertical lines (V_BP+V_ACTIVE+V_FP)	Between vertical synchronization signals: 2047 lines or less
Number of horizontal pixels (H_BP+H_ACTIVE+H_FP)	Between horizontal synchronization signals: 2047 CLK or less

Note 1. When V_FP is below four lines, the setting of INP_DLY_ADJ.INP_VS_DLY_L[2:0] should be adjusted so that V_FP is at least four lines.

Note 2. When H_FP is below 16 cycles, the settings of INP_DLY_ADJ.INP_VS_DLY[7:0], INP_HS_DLY[7:0], and INP_FLD_DLY[7:0] should be adjusted so that H_FP is at least 16 cycles of the pixel clock.

chanhvinh.com

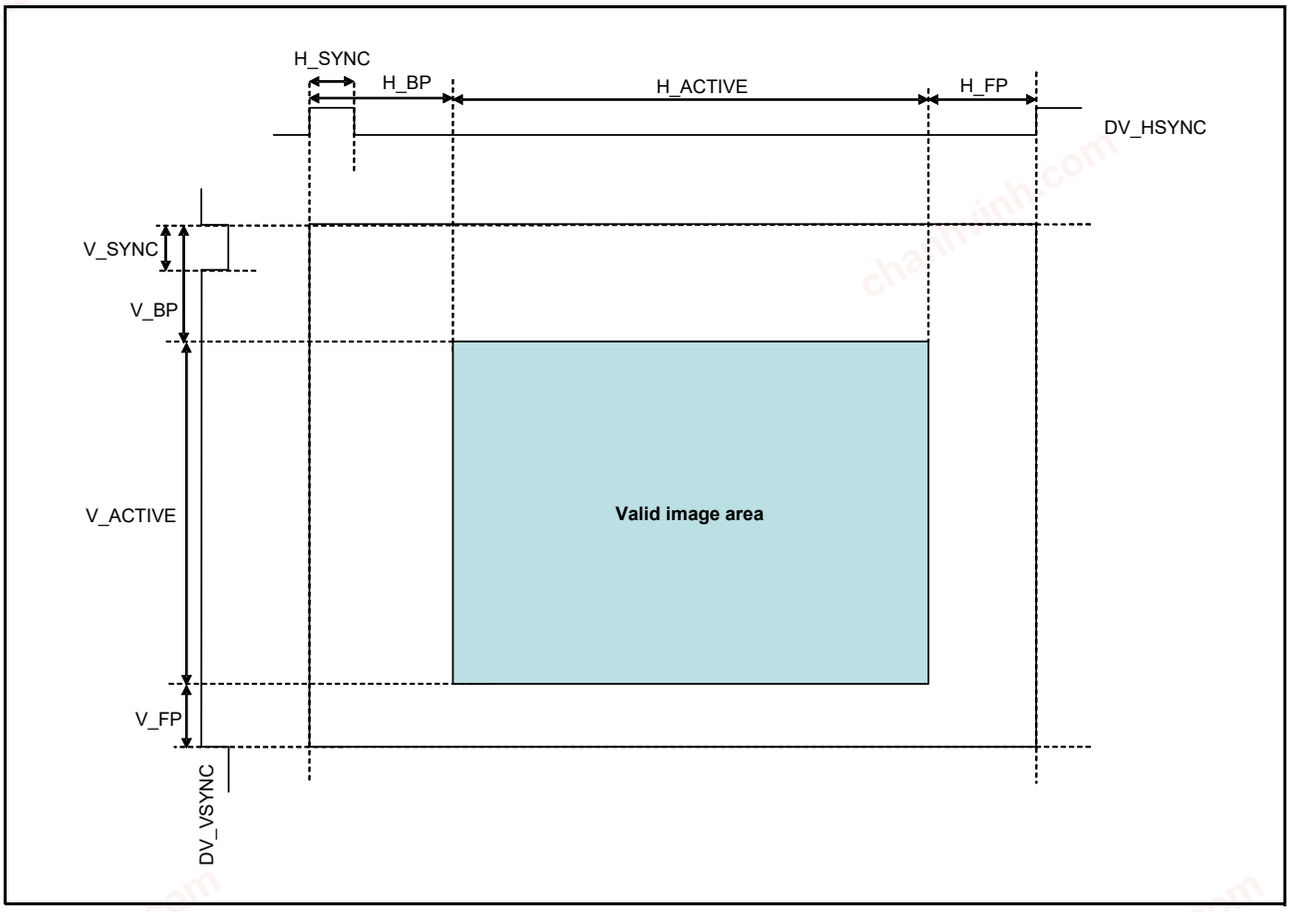


Figure 30.17 YCbCr/RGB Signal Reception Timing

chanhvinh.com

30.1.14 Field Differentiation and Vsync Signal Phase Adjustment

The phase of the input Vsync signal and Hsync signal is detected and the field of the interlace signal is determined. When the reference point of the Vsync signal is detected within ±0.5 horizontal period with respect to the Hsync signal, it is determined as the interlace top field. When the reference point of the Vsync signal is detected outside ±0.5 horizontal period with respect to the Hsync signal, it is determined as the interlace bottom field.

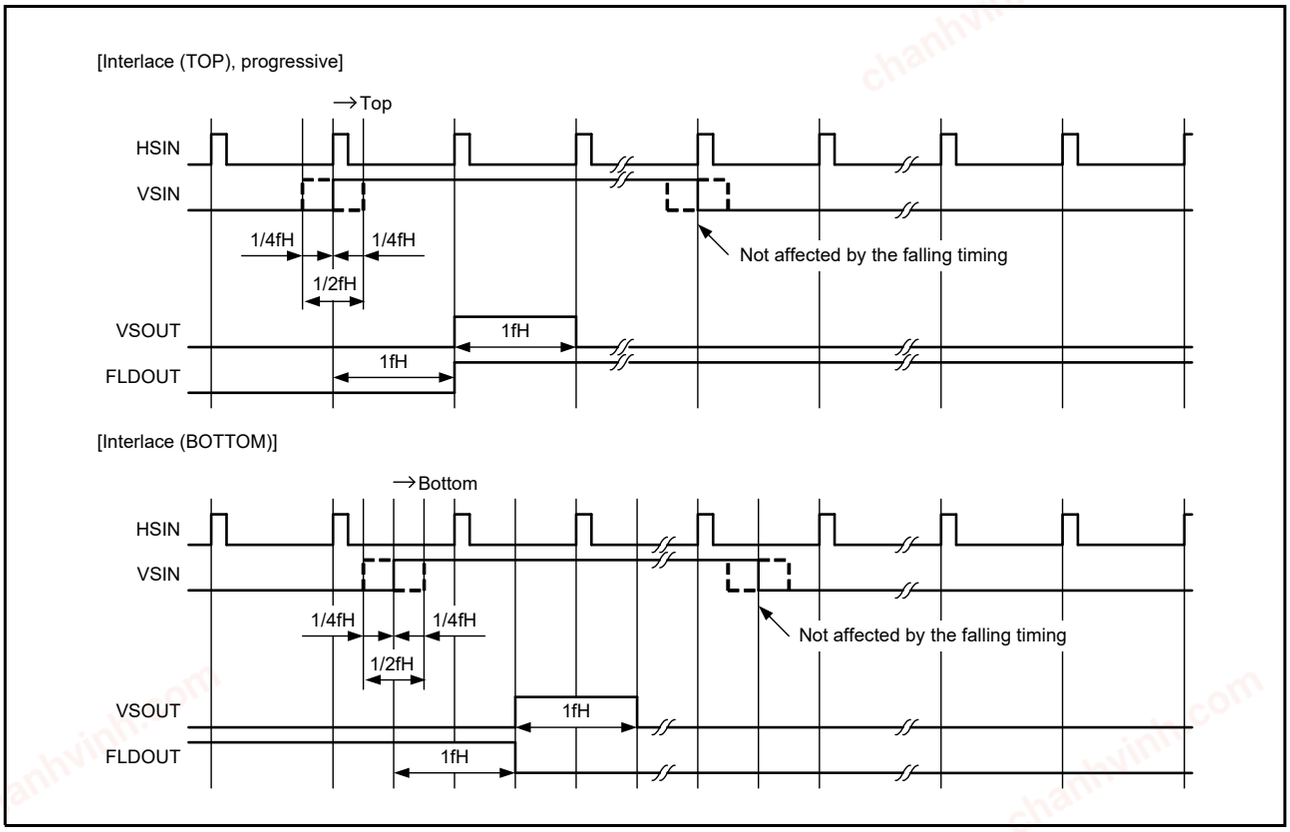


Figure 30.18 Vsync Signal Phase Adjustment

The timings of 1/2fH Vsync signal phase and 1/4fH Vsync signal phase are set with INP_FH50[9:0] and INP_FH25[9:0], respectively.

Table 30.20 Vsync Signal Phase Timing Setting

Register Name	Bit Name	Initial Value	Description
INP_VSYNC_PH_ADJ	INP_FH50[9:0]	858	Vsync Signal 1/2fH Phase Timing Should be 1/2 the horizontal cycle.
INP_VSYNC_PH_ADJ	INP_FH25[9:0]	429	Vsync Signal 1/4fH Phase Timing Should be 1/4 the horizontal cycle.

chanhvinh.com

30.1.15 Vsync Signal Delay Adjustment in Line Units

The Vsync signal line delay adjust block can delay the Vsync signal and the field differentiation signal in line units.

When a video signal with a short vertical front porch is input, the vertical front porch is adjusted.

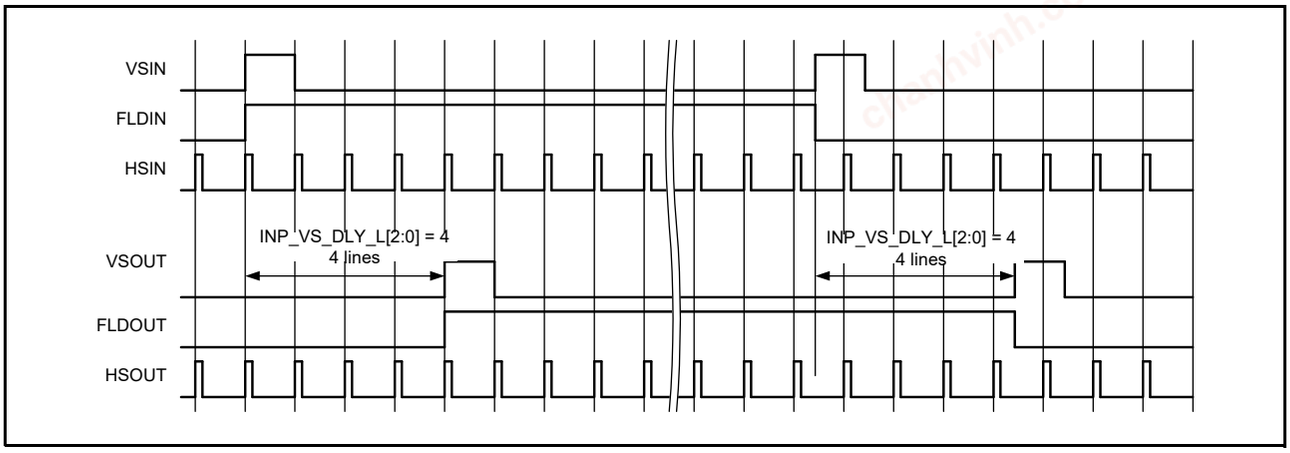


Figure 30.19 Timing of Vsync Signal Delay in Line Units

Table 30.21 Adjustment of Vsync Signal Delay in Line Units

Register Name	Bit Name	Initial Value	Description
INP_DLY_ADJ	INP_VS_DLY_L[2:0]	0	Number of Lines for Delaying Vsync Signal and Field Differentiation Signal Delay amount: 0 to 7 (lines)

30.1.16 Sync Signal Delay Adjustment

Delay can be adjusted independently for the Vsync signal, Hsync signal, and field differentiation signal in the units of clock.

Lacking margin of the horizontal front porch is adjusted according to the input synchronization disturbance.

Table 30.22 Sync Signal Delay Adjustment

Register Name	Bit Name	Initial Value	Description
INP_DLY_ADJ	INP_VS_DLY[7:0]	0	Vsync Signal Delay Amount Delay amount: 0 to 254 (clock cycles)
INP_DLY_ADJ	INP_HS_DLY[7:0]	0	Hsync Signal Delay Amount Delay amount: 0 to 254 (clock cycles)
INP_DLY_ADJ	INP_FLD_DLY[7:0]	0	Field Differentiation Signal Delay Amount Delay amount: 0 to 254 (clock cycles)

chanhvinh.com

30.1.17 Horizontal Noise Reduction

Noise can be reduced according to horizontal pixel reference.

Noise reduction is controlled through noise component frequency band (TAP), noise level (threshold), and noise reduction intensity (gain).

(1) Frequency Band (TAP) Setting for Noise Component

The noise frequency band can be selected independently from the following four types by using the NR1D_Y_TAP[1:0], NR1D_CB_TAP[1:0], and NR1D_CR_TAP[1:0] bits.

When the number of adjacent pixels is one (noise reduction NR1D_Y/CB/CR_TAP is 0):

$$BPF_{(1)} = \frac{1}{4}\left(-1\times Z_{(-1)}, 2\times Z_{(0)}, -1\times Z_{(+1)}\right)$$

When the number of adjacent pixels is two (noise reduction NR1D_Y/CB/CR_TAP is 1):

$$BPF_{(2)} = \frac{1}{4}\left(-1\times Z_{(-2)}, 2\times Z_{(0)}, -1\times Z_{(+2)}\right)$$

When the number of adjacent pixels is three (noise reduction NR1D_Y/CB/CR_TAP is 2):

$$BPF_{(3)} = \frac{1}{4}\left(-1\times Z_{(-3)}, 2\times Z_{(0)}, -1\times Z_{(+3)}\right)$$

When the number of adjacent pixels is four (noise reduction NR1D_Y/CB/CR_TAP is 3):

$$BPF_{(4)} = \frac{1}{4}\left(-1\times Z_{(-4)}, 2\times Z_{(0)}, -1\times Z_{(+4)}\right)$$

Note: $Z_{(0)}$ indicates the target pixel for noise reduction and $Z_{(N)}$ indicates the pixel that is n pixels off from $Z_{(0)}$ In the horizontal direction.

(2) Setting Noise Level (Threshold)

The absolute value of the detected noise amount (BPF output value) is compared with the values of the NR1D_Y_TH[6:0], NR1D_CB_TH[6:0], and NR1D_CR_TH[6:0] bits. When the detected noise amount is greater than NR1D_Y/CB/CR_TH, the absolute value of the detected noise amount is considered as NR1D_Y/CB/CR_TH (fixed value).

ABS($BPF_{(n)}$) ≤ absolute value of detected noise amount when ABS($BPF_{(n)}$) ≤ NR1D_Y/CB/CR_TH: NOISE_ABS = ABS($BPF_{(n)}$)

ABS($BPF_{(n)}$) > absolute value of detected noise amount when ABS($BPF_{(n)}$) > NR1D_Y/CB/CR_TH: NOISE_ABS = NR1D_Y/CB/CR_TH

(3) Setting Noise Reduction Intensity (Gain)

The absolute value of the detected noise amount is multiplied by the value of gain specified by the NR1D_Y_GAIN[1:0], NR1D_CB_GAIN[1:0], and NR1D_CR_GAIN[1:0] bits, and the feedback is calculated for the original signal.

Computation when the amount of detected noise ($BPF_{(n)}$) is negative (−): DOUT = DIN + NOISE_ABS ÷ $2^{(NR1D_Y/CB/CR_GAIN+1)}$

chanhvinh.com

Computation when the amount of detected noise ($BPF_{(n)}$) is positive (+): DOUT = DIN – NOISE_ABS ÷ $2^{(NR1D_Y/CB/CR_GAIN+1)}$

Table 30.23 Horizontal Noise Reduction

Register Name	Bit Name	Initial Value	Description
IMGCNT_NR_CNT0	NR1D_MD	1	Horizontal Noise Reduction Operating Mode 0: R/G/B mode 1: Y/Cb/Cr mode
IMGCNT_NR_CNT0	NR1D_ON	0	Noise Reduction On/Off Control 0: Noise Reduction Off 1: Noise Reduction On
IMGCNT_NR_CNT0	NR1D_Y_TAP[1:0]	0	Y/G Signal TAP Select 0: Adjacent pixel 1: 2 adjacent pixels 2: 3 adjacent pixels 3: 4 adjacent pixels
IMGCNT_NR_CNT0	NR1D_Y_TH[6:0]	8	Maximum Value (Absolute Value) of Y/G Signal Coring Coring is implemented when detected noise amount value ≤ NR1D_Y_TH. Unsigned: 0 to 127 [LSB]
IMGCNT_NR_CNT0	NR1D_Y_GAIN[1:0]	3	Noise Reduction Gain Adjustment of Y/G Signal 0: 1/2 1: 1/4 2: 1/8 3: 1/16
IMGCNT_NR_CNT1	NR1D_CB_TAP[1:0]	0	Cb/B Signal TAP Select 0: Adjacent pixel 1: 2 adjacent pixels 2: 3 adjacent pixels 3: 4 adjacent pixels
IMGCNT_NR_CNT1	NR1D_CB_TH[6:0]	8	Maximum Value (Absolute Value) of Cb/B Signal Coring Coring is implemented when detected noise amount value ≤ NR1D_C_TH. Unsigned: 0 to 127 [LSB]
IMGCNT_NR_CNT1	NR1D_CB_GAIN[1:0]	3	Noise Reduction Gain Adjustment of Cb/B Signal 0: 1/2 1: 1/4 2: 1/8 3: 1/16
IMGCNT_NR_CNT1	NR1D_CR_TAP[1:0]	0	Cr/R Signal TAP Select 0: Adjacent pixel 1: 2 adjacent pixels 2: 3 adjacent pixels 3: 4 adjacent pixels
IMGCNT_NR_CNT1	NR1D_CR_TH[6:0]	8	Maximum Value (Absolute Value) of Cr/R Signal Coring Coring is implemented when detected noise amount value ≤ NR1D_C_TH. Unsigned: 0 to 127 [LSB]
IMGCNT_NR_CNT1	NR1D_CR_GAIN[1:0]	3	Noise Reduction Gain Adjustment of Cr/R Signal 0: 1/2 1: 1/4 2: 1/8 3: 1/16

chanhvinh.com

30.1.18 Color Matrix

By using a color matrix, input signal offsets and nine-axis gain can be adjusted. This enables brightness adjustment, gain adjustment, and YCbCr and GBR mutual conversion.

(1) GBR to GBR Conversion

YGIN_A = YGIN + IMGCNT_MTX_YG – 128

CBBIN_A = CBBIN + IMGCNT_MTX_B – 128

CRRIN_A = CRRIN + IMGCNT_MTX_R – 128

YGOUT = (IMGCNT_MTX_GG×YGIN_A + IMGCNT_MTX_GB×CBBIN_A + IMGCNT_MTX_GR×CRRIN_A) ÷ 256

CBBOUT = (IMGCNT_MTX_BG×YGIN_A + IMGCNT_MTX_BB×CBBIN_A + IMGCNT_MTX_BR×CRRIN_A) ÷ 256

CRROUT = (IMGCNT_MTX_RG×YGIN_A + IMGCNT_MTX_RB×CBBIN_A + IMGCNT_MTX_RR×CRRIN_A) ÷ 256

(2) GBR to YCbCr Conversion

YGIN_A = YGIN + IMGCNT_MTX_YG – 128

CBBIN_A = CBBIN + IMGCNT_MTX_B – 128

CRRIN_A = CRRIN + IMGCNT_MTX_R – 128

YGOUT = (IMGCNT_MTX_GG×YGIN_A + IMGCNT_MTX_GB×CBBIN_A + IMGCNT_MTX_GR×CRRIN_A) ÷ 256

CBBOUT = (IMGCNT_MTX_BG×YGIN_A + IMGCNT_MTX_BB×CBBIN_A + IMGCNT_MTX_BR×CRRIN_A) ÷ 256 + 128

CRROUT = (IMGCNT_MTX_RG×YGIN_A + IMGCNT_MTX_RB×CBBIN_A + IMGCNT_MTX_RR×CRRIN_A) ÷ 256 + 128

Table 30.24 Matrix Coefficient (Typical Value) for SMPTE 293M

	YGIN		CBBIN		CRRIN	
	Coefficient	**Set Value**	**Coefficient**	**Set Value**	**Coefficient**	**Set Value**
YGOUT	0.587	IMGCNT_ MTX_GG = 150	0.114	IMGCNT_MTX_GB = 29	0.299	IMGCNT_MTX_GR = 77
CBBOUT	–0.331	IMGCNT_ MTX_BG = 1963	0.500	IMGCNT_MTX_BB = 128	–0.169	IMGCNT_MTX_BR = 2005
CRROUT	–0.419	IMGCNT_ MTX_RG = 1941	–0.081	IMGCNT_MTX_RB = 2027	0.500	IMGCNT_MTX_RR = 128

chanhvinh.com

(3) YCbCr to GBR Conversion

YGIN_A = YGIN + IMGCNT_MTX_YG – 128

CBBIN_A = CBBIN – 128

CRRIN_A = CRRIN – 128

YGOUT = (IMGCNT_MTX_GG×YGIN_A + IMGCNT_MTX_GB×CBBIN_A + IMGCNT_MTX_GR×CRRIN_A) ÷ 256

CBBOUT = (IMGCNT_MTX_BG×YGIN_A + IMGCNT_MTX_BB×CBBIN_A + IMGCNT_MTX_BR×CRRIN_A) ÷ 256

CRROUT = (IMGCNT_MTX_RG×YGIN_A + IMGCNT_MTX_RB×CBBIN_A + IMGCNT_MTX_RR×CRRIN_A) ÷ 256

Table 30.25 Matrix Coefficient (Typical Value) for SMPTE 293M

	YGIN		CBBIN		CRRIN	
	Coefficient	**Set Value**	**Coefficient**	**Set Value**	**Coefficient**	**Set Value**
YGOUT	1.000	IMGCNT_ MTX_GG = 256	–0.344	IMGCNT_MTX_GB = 1960	–0.714	IMGCNT_MTX_GR = 1865
CBBOUT	1.000	IMGCNT_ MTX_BG = 256	1.772	IMGCNT_MTX_BB = 454	0.000	IMGCNT_MTX_BR = 0
CRROUT	1.000	IMGCNT_ MTX_RG = 256	0.000	IMGCNT_MTX_RB = 0	1.402	IMGCNT_MTX_RR = 359

(4) YCbCr to YCbCr Conversion

YGIN_A = YGIN + IMGCNT_MTX_YG – 128

CBBIN_A = CBBIN – 128

CRRIN_A = CRRIN – 128

YGOUT = (IMGCNT_MTX_GG×YGIN_A + IMGCNT_MTX_GB×CBBIN_A + IMGCNT_MTX_GR×CRRIN_A) ÷ 256

CBBOUT = (IMGCNT_MTX_BG×YGIN_A + IMGCNT_MTX_BB×CBBIN_A + IMGCNT_MTX_BR×CRRIN_A) ÷ 256 + 128

CRROUT = (IMGCNT_MTX_RG×YGIN_A + IMGCNT_MTX_RB×CBBIN_A + IMGCNT_MTX_RR×CRRIN_A) ÷ 256 + 128

chanhvinh.com

Table 30.26 YCbCr to GBR Conversion

Register Name	Bit Name	Initial Value	Description
IMGCNT_MTX_MODE	IMGCNT_MTX_MD [1:0]	3	Operating Mode 0: GBR → GBR 1: GBR → YCbCr 2: YCbCr → GBR 3: YCbCr → YCbCr
IMGCNT_MTX_YG_ ADJ0	IMGCNT_MTX_YG [7:0]	128	Offset (DC) Adjustment of Y/G Signal Unsigned (0 (-128) to 128 (0) to 255 (+127) [LSB], 512 [LSB])
IMGCNT_MTX_CBB_ ADJ0	IMGCNT_MTX_B [7:0]	128	Offset (DC) Adjustment of B Signal Unsigned (0 (-128) to 128 (0) to 255 (+127) [LSB])
IMGCNT_MTX_CRR_ ADJ0	IMGCNT_MTX_R [7:0]	128	Offset (DC) Adjustment of R Signal Unsigned (0 (-128) to 128 (0) to 255 (+127) [LSB])
IMGCNT_MTX_YG_ ADJ0	IMGCNT_MTX_GG [10:0]	256	Y/G Signal Gain Adjustment for Y/G Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
IMGCNT_MTX_YG_ ADJ1	IMGCNT_MTX_GB [10:0]	0	Cb/B Signal Gain Adjustment for Y/G Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
IMGCNT_MTX_YG_ ADJ1	IMGCNT_MTX_GR [10:0]	0	Cr/R Signal Gain Adjustment for Y/G Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
IMGCNT_MTX_CBB_ ADJ0	IMGCNT_MTX_BG [10:0]	0	Y/G Signal Gain Adjustment for Cb/B Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
IMGCNT_MTX_CBB_ ADJ1	IMGCNT_MTX_BB [10:0]	256	Cb/B Signal Gain Adjustment for Cb/B Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
IMGCNT_MTX_CBB_ ADJ1	IMGCNT_MTX_BR [10:0]	0	Cr/R Signal Gain Adjustment for Cb/B Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
IMGCNT_MTX_CRR_ ADJ0	IMGCNT_MTX_RG [10:0]	0	Y/G Signal Gain Adjustment for Cr/R Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
IMGCNT_MTX_CRR_ ADJ1	IMGCNT_MTX_RB [10:0]	0	Cb/B Signal Gain Adjustment for Cr/R Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
IMGCNT_MTX_CRR_ ADJ1	IMGCNT_MTX_RR [10:0]	256	Cr/R Signal Gain Adjustment for Cr/R Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

chanhvinh.com

30.2 Register Descriptions

Table 30.27 and Table 30.28 show register Configuration.

- Symbols used in Register Description:

Initial value: Register value after a power-on reset
—: Undefined value
R/W: Readable/writable. The written value can be read.
R/WC0: Readable/writable. Writing 0 initializes the bit. Writing 1 is ignored.
R/WC1: Readable/writable. Writing 1 initializes the bit. Writing 0 is ignored.
R: Read-only. The write value should always be 0.
—/W: Write-only. The read value is undefined.

Table 30.27 Register Configuration of Input Controller

Name	Abbreviation	R/W	Initial Value	Address	Access Size
External input block register update control register	INP_UPDATE	R/WC1	H'0000 0000	H'FCFF 7400	32
Input select control register	INP_SEL_CNT	R/W	H'0000 0000	H'FCFF 7404	32
External input sync signal control register	INP_EXT_SYNC_CNT	R/W	H'0000 0000	H'FCFF 7408	32
Vsync signal phase adjustment register	INP_VSYNC_PH_ADJ	R/W	H'035A 01AD	H'FCFF 740C	32
Sync signal delay adjustment register	INP_DLY_ADJ	R/W	H'0000 0000	H'FCFF 7410	32

Table 30.28 Register Configuration of Image Quality Adjustment Block

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Image quality adjustment block register update control register	IMGCNT_UPDATE	R/WC1	H'0000 0000	H'FCFF 7480	32
NR control register 0	IMGCNT_NR_CNT0	R/W	H'0010 0803	H'FCFF 7484	32
NR control register 1	IMGCNT_NR_CNT1	R/W	H'0803 0803	H'FCFF 7488	32
Image quality adjustment block matrix mode register	IMGCNT_MTX_MODE	R/W	H'0000 0003	H'FCFF 74A0	32
Image quality adjustment block matrix YG adjustment register 0	IMGCNT_MTX_YG_ ADJ0	R/W	H'0080 0100	H'FCFF 74A4	32
Image quality adjustment block matrix YG adjustment register 1	IMGCNT_MTX_YG_ ADJ1	R/W	H'0000 0000	H'FCFF 74A8	32
Image quality adjustment block matrix CBB adjustment register 0	IMGCNT_MTX_CBB_ ADJ0	R/W	H'0080 0000	H'FCFF 74AC	32
Image quality adjustment block matrix CBB adjustment register 1	IMGCNT_MTX_CBB_ ADJ1	R/W	H'0100 0000	H'FCFF 74B0	32
Image quality adjustment block matrix CRR adjustment register 0	IMGCNT_MTX_CRR_ ADJ0	R/W	H'0080 0000	H'FCFF 74B4	32
Image quality adjustment block matrix CRR adjustment register 1	IMGCNT_MTX_CRR_ ADJ1	R/W	H'0000 0100	H'FCFF 74B8	32

chanhvinh.com

30.2.1 External Input Block Register Update Control Register (INP_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	INP_EXT_UPDATE	—	—	—	INP_IMG_UPDATE
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/WC1	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	INP_EXT_UPDATE	0	R/WC1	External Input Block Register Update 0: Registers are not updated. 1: Registers are updated.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	INP_IMG_UPDATE	0	R/WC1	Sync Signal Adjustment Block Register Update 0: Registers are not updated. 1: Registers are updated.

chanhvinh.com

30.2.2 Input Select Control Register (INP_SEL_CNT)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	INP_SEL	—	—	—	—
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	INP_FORMAT[2:0]			—	—	—	INP_PXD_EDGE	—	—	—	INP_VS_EDGE	—	—	—	INP_HS_EDGE
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	INP_SEL	0	R/W	Input Select 0: Input supplied via the external input pins is off. 1: Input supplied via the external input pins is on.
19 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
14 to 12	INP_FORMAT[2:0]	0	R/W	External Input Format Select 0: YcbCr444, RGB888 1: RGB666 2: RGB565 3: BT656 4: BT601 5: YCbCr422 6, 7: Setting prohibited
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	INP_PXD_EDGE	0	R/W	Clock Edge Select for Capturing External Input Video Image Signals DV_DATA23 to DV_DATA0 0: Rising edge 1: Falling edge
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	INP_VS_EDGE	0	R/W	Clock Edge Select for Capturing External Input Vsync Signals DV_VSYNC 0: Rising edge 1: Falling edge
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	INP_HS_EDGE	0	R/W	Clock Edge Select for Capturing External Input Hsync Signals DV_HSYNC 0: Rising edge 1: Falling edge

Note: INP_FORMAT, INP_PXD_EDGE, INP_VS_EDGE, and INP_HS_EDGE are updated when the INP_EXT_UPDATE bit in INP_UPDATE is 1. INP_SEL is updated when set.

chanhvinh.com

30.2.3 External Input Sync Signal Control Register (INP_EXT_SYNC_CNT)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	INP_ENDIAN_ON	—	—	—	INP_SWAP_ON	—	—	—	INP_VS_INV	—	—	—	INP_HS_INV
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	INP_H_EDGE_SEL	—	—	—	INP_F525_625	—	—	INP_H_POS[1:0]	
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28	INP_ENDIAN_ON	0	R/W	External Input Bit Endian Change On/Off Control 0: Off 1: On
27 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	INP_SWAP_ON	0	R/W	External Input B/R Signal Swap On/Off Control 0: Off 1: On
23 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	INP_VS_INV	0	R/W	External Input Vsync Signal DV_VSYNC Inversion Control 0: Not inverted (positive polarity) 1: Inverted (negative polarity)
19 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	INP_HS_INV	0	R/W	External Input Hsync Signal DV_HSYNC Inversion Control 0: Not inverted (positive polarity) 1: Inverted (negative polarity)
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	INP_H_EDGE_SEL	0	R/W	Reference Select for External Input BT656 Hsync Signal 0: EAV 1: SAV
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	INP_F525_625	0	R/W	Number of Lines for BT656 External Input 0: 525 lines 1: 625 lines
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	INP_H_POS[1:0]	0	R/W	Y/Cb/Y/Cr Data String Start Timing to Hsync Reference for BT656/601 or YCbCr422 External Input 0: Cb/Y/Cr/Y(BT656/601), Cb/Cr (YCbCr422) 1: Y/Cr/Y/Cb(BT656/601), setting prohibited (YCbCr422) 2: Cr/Y/Cb/Y(BT656/601), setting prohibited (YCbCr422) 3: Y/Cb/Y/Cr(BT656/601), Cr/Cb (YCbCr422)

Note: This register is updated when the INP_EXT_UPDATE bit in INP_UPDATE is 1.

chanhvinh.com

30.2.4 Vsync Signal Phase Adjustment Register (INP_VSYNC_PH_ADJ)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	INP_FH50[9:0]									
Initial Value:	0	0	0	0	0	0	1	1	0	1	0	1	1	0	1	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	INP_FH25[9:0]									
Initial Value:	0	0	0	0	0	0	0	1	1	0	1	0	1	1	0	1
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 16	INP_FH50 [9:0]	858	R/W	Vsync Signal 1/2fH Phase Timing 1/2 clock cycle of the horizontal cycle should be set.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	INP_FH25 [9:0]	429	R/W	Vsync Signal 1/4fH Phase Timing 1/4 clock cycle of the horizontal cycle should be set.

Note: The INP_FH50[9:0] bits are updated when the INP_EXT_UPDATE and INP_IMG_UPDATE bits in INP_UPDATE are 1. The IMP_FH25[9:0] bits are updated when the INP_IMG_UPDATE bit is 1.

30.2.5 Sync Signal Delay Adjustment Register (INP_DLY_ADJ)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	INP_VS_DLY_L[2:0]			INP_FLD_DLY[7:0]							
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INP_VS_DLY[7:0]								INP_HS_DLY[7:0]							
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 24	INP_VS_ DLY_L[2:0]	0	R/W	Number of lines for Delaying Vsync signal and Field Differentiation Signal Delay amount: 0 to 7 (lines)
23 to 16	INP_FLD_ DLY[7:0]	0	R/W	Field Differentiation Signal Delay Amount Delay amount: 0 to 254 (clock cycles)
15 to 8	INP_VS_ DLY[7:0]	0	R/W	Vsync Signal Delay Amount Delay amount: 0 to 254 (clock cycles)
7 to 0	INP_HS_ DLY[7:0]	0	R/W	Hsync Signal Delay Amount Delay amount: 0 to 254 (clock cycles)

Note: This register is updated when the INP_IMG_UPDATE bit in INP_UPDATE is 1.

chanhvinh.com

30.2.6 Image Quality Adjustment Block Register Update Control Register (IMGCNT_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IMGCNT_VEN
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	IMGCNT_VEN	0	R/WC1	Image Quality Adjustment Block Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync signal.

chanhvinh.com

30.2.7 NR Control Register 0 (IMGCNT_NR_CNT0)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	NR1D_MD	—	—	—	NR1D_ON
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	NR1D_Y_TH[6:0]							—	—	NR1D_Y_TAP[1:0]		—	—	NR1D_Y_GAIN[1:0]	
Initial Value:	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	NR1D_MD	1	R/W	Horizontal Noise Reduction Operating Mode 0: G/B/R mode 1: Y/Cb/Cr mode
19 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	NR1D_ON	0	R/W	Noise Reduction On/Off Control 0: Noise reduction Off 1: Noise reduction On
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 8	NR1D_Y_TH[6:0]	8	R/W	Maximum Value (Absolute Value) of Y/G Signal Coring Coring is implemented when detected noise amount value ≤ NR1D_Y_TH. Unsigned: 0 to 127 [LSB]
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	NR1D_Y_TAP[1:0]	0	R/W	Y/G Signal TAP Select 0: Adjacent pixel 1: 2 adjacent pixels 2: 3 adjacent pixels 3: 4 adjacent pixels
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	NR1D_Y_GAIN[1:0]	3	R/W	Noise Reduction Gain Adjustment of Y/G Signal 0: 1/2 1: 1/4 2: 1/8 3: 1/16

Note: This register is updated when the IMGCNT_VEN bit in IMGCNT_UPDATE is 1.

chanhvinh.com

30.2.8 NR Control Register 1 (IMGCNT_NR_CNT1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	NR1D_CB_TH[6:0]							—	—	NR1D_CB_TAP[1:0]		—	—	NR1D_CB_GAIN[1:0]	
Initial Value:	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	NR1D_CR_TH[6:0]							—	—	NR1D_CR_TAP[1:0]		—	—	NR1D_CR_GAIN[1:0]	
Initial Value:	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 24	NR1D_CB_TH[6:0]	8	R/W	Maximum Value (Absolute Value) of Cb/B Signal Coring Coring is implemented when detected noise amount value ≤ NR1D_CB_TH. Unsigned: 0 to 127 [LSB]
23, 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21, 20	NR1D_CB_TAP[1:0]	0	R/W	Cb/B Signal TAP Select 0: Adjacent pixel 1: 2 adjacent pixels 2: 3 adjacent pixels 3: 4 adjacent pixels
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	NR1D_CB_GAIN[1:0]	3	R/W	Noise Reduction Gain Adjustment of Cb/B Signal 0: 1/2 1: 1/4 2: 1/8 3: 1/16
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 8	NR1D_CR_TH[6:0]	8	R/W	Maximum Value (Absolute Value) of Cr/R Signal Coring Coring is implemented when detected noise amount value ≤ NR1D_CR_TH. Unsigned: 0 to 127 [LSB]
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	NR1D_CR_TAP[1:0]	0	R/W	Cr/R Signal TAP Select 0: Adjacent pixel 1: 2 adjacent pixels 2: 3 adjacent pixels 3: 4 adjacent pixels
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	NR1D_CR_GAIN[1:0]	3	R/W	Noise Reduction Gain Adjustment of Cr/R Signal 0: 1/2 1: 1/4 2: 1/8 3: 1/16

Note: This register is updated when the IMGCNT_VEN bit in IMGCNT_UPDATE is 1.

chanhvinh.com

30.2.9 Image Quality Adjustment Block Matrix Mode Register (IMGCNT_MTX_MODE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IMGCNT_MTX_MD[1:0]	
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	IMGCNT_MTX_MD[1:0]	3	R/W	Operating Mode 0: GBR → GBR 1: GBR → YCbCr 2: YCbCr → GBR 3: YCbCr → YCbCr

Note: This register is updated when the IMGCNT_VEN bit in IMGCNT_UPDATE is 1.

30.2.10 Image Quality Adjustment Block Matrix YG Adjustment Register 0 (IMGCNT_MTX_YG_ADJ0)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	IMGCNT_MTX_YG[7:0]							
Initial Value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	IMGCNT_MTX_GG[10:0]										
Initial Value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	IMGCNT_MTX_YG[7:0]	128	R/W	Offset (DC) Adjustment of Y/G Signal Unsigned (0 (-128) to 128 (0) to 255 (+127) [LSB])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	IMGCNT_MTX_GG[10:0]	256	R/W	Y/G Signal Gain Adjustment for Y/G Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when the IMGCNT_VEN bit in IMGCNT_UPDATE is 1.

chanhvinh.com

30.2.11 Image Quality Adjustment Block Matrix YG Adjustment Register 1 (IMGCNT_MTX_YG_ADJ1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	IMGCNT_MTX_GB[10:0]										
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	IMGCNT_MTX_GR[10:0]										
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	IMGCNT_MTX_GB[10:0]	0	R/W	Cb/B Signal Gain Adjustment for Y/G Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	IMGCNT_MTX_GR[10:0]	0	R/W	Cr/R Signal Gain Adjustment for Y/G Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when the IMGCNT_VEN bit in IMGCNT_UPDATE is 1.

30.2.12 Image Quality Adjustment Block Matrix CBB Adjustment Register 0 (IMGCNT_MTX_CBB_ADJ0)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	IMGCNT_MTX_B[7:0]							
Initial Value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	IMGCNT_MTX_BG[10:0]										
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	IMGCNT_MTX_B[7:0]	128	R/W	Offset (DC) Adjustment of Cb/B Signal Unsigned (0 (-128) to 128 (0) to 255 (+127) [LSB])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	IMGCNT_MTX_BG[10:0]	0	R/W	Y/G Signal Gain Adjustment for Cb/B Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when the IMGCNT_VEN bit in IMGCNT_UPDATE is 1.

chanhvinh.com

30.2.13 Image Quality Adjustment Block Matrix CBB Adjustment Register 1 (IMGCNT_MTX_CBB_ADJ1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	IMGCNT_MTX_BB[10:0]										
Initial Value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	IMGCNT_MTX_BR[10:0]										
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	IMGCNT_ MTX_BB [10:0]	256	R/W	Cb/B Signal Gain Adjustment for Cb/B Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	IMGCNT_ MTX_BR [10:0]	0	R/W	Cr/R Signal Gain Adjustment for Cb/B Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when the IMGCNT_VEN bit in IMGCNT_UPDATE is 1.

chanhvinh.com

30.2.14 Image Quality Adjustment Block Matrix CRR Adjustment Register 0 (IMGCNT_MTX_CRR_ADJ0)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	IMGCNT_MTX_R[7:0]							
Initial Value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	IMGCNT_MTX_RG[10:0]										
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	IMGCNT_ MTX_R[7:0]	128	R/W	Offset (DC) Adjustment of Cr/R Signal Unsigned (0 (-128) to 128 (0) to 255 (+127) [LSB])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	IMGCNT_ MTX_RG[10:0]	0	R/W	Y/G Signal Gain Adjustment for Cr/R Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when the IMGCNT_VEN bit in IMGCNT_UPDATE is 1.

chanhvinh.com

30.2.15 Image Quality Adjustment Block Matrix CRR Adjustment Register 1 (IMGCNT_MTX_CRR_ADJ1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	IMGCNT_MTX_RB[10:0]										
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	IMGCNT_MTX_RR[10:0]										
Initial Value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	IMGCNT_ MTX_RB[10:0]	0	R/W	Cb/B Signal Gain Adjustment for Cr/R Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	IMGCNT_ MTX_RR[10:0]	256	R/W	Cr/R Signal Gain Adjustment for Cr/R Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when the IMGCNT_VEN bit in IMGCNT_UPDATE is 1.

chanhvinh.com

30.3 Usage Methods

30.3.1 Input Format Adjustment Method

Setting examples of each input format are shown below.

Table 30.29 External Input (BT656, 525i) Setting Example

Register Name	Bit Name	Description	Setting Value
INP_SEL_CNT	INP_SEL	Performs the on/off control of input supplied via the external input pins.	1
INP_SEL_CNT	INP_FORMAT[2:0]	Selects the externally input format.	3
INP_SEL_CNT	INP_PXD_EDGE	Selects the clock edge for capturing the externally input video signals.	0
INP_SEL_CNT	INP_VS_EDGE	Selects the clock edge for capturing the externally input Vsync signals.	0
INP_SEL_CNT	INP_HS_EDGE	Selects the clock edge for capturing the externally input Hsync signals.	0
INP_EXT_SYNC_CNT	INP_ENDIAN_ON	Changes the bit endian of the external input.	0
INP_EXT_SYNC_CNT	INP_SWAP_ON	Enables or disables the B/R signal swap of the external input.	0
INP_EXT_SYNC_CNT	INP_HS_INV	Enables or disables the Hsync signal inversion of the external input.	1
INP_EXT_SYNC_CNT	INP_VS_INV	Enables or disables the Hsync signal inversion of the external input.	1
INP_EXT_SYNC_CNT	INP_H_EDGE_SEL	Selects the Hsync reference for BT656 input.	0
INP_EXT_SYNC_CNT	INP_F525_625	Sets the number of lines for BT656 input.	0
INP_EXT_SYNC_CNT	INP_H_POS[1:0]	Sets the data start timing with respect to the Hsync in the BT656/601 format.	0
INP_VSYNC_PH_ADJ	INP_FH50[9:0]	Sets the 1/2fH phase in clock cycle units.	858
INP_VSYNC_PH_ADJ	INP_FH25[9:0]	Sets the 1/4fH phase in clock cycle units.	429
INP_DLY_ADJ	INP_VS_DLY_L[2:0]	Sets the number of lines for delaying the Vsync signal and field differentiation signal.	0
INP_DLY_ADJ	INP_VS_DLY[7:0]	Sets the amount of delay of the Vsync signal in clock cycle units.	0
INP_DLY_ADJ	INP_HS_DLY[7:0]	Sets the amount of delay of the Hsync signal in clock cycle units.	0
INP_DLY_ADJ	INP_FLD_DLY[7:0]	Sets the amount of delay of the field differentiation signal in clock units.	0

Note: Some registers require, after they are set, that the INP_EXT_UPDATE and INP_IMG_UPDATE bits in INP_UPDATE should be set to 1.

chanhvinh.com

Table 30.30 External Input (BT601, 525i) Setting Example

Register Name	Bit Name	Description	Setting Value
INP_SEL_CNT	INP_SEL	Performs the on/off control of input supplied via the external input pins.	1
INP_SEL_CNT	INP_FORMAT[2:0]	Selects the externally input format.	4
INP_SEL_CNT	INP_PXD_EDGE	Selects the clock edge for capturing the externally input video signals.	0
INP_SEL_CNT	INP_VS_EDGE	Selects the clock edge for capturing the externally input Vsync signals.	0
INP_SEL_CNT	INP_HS_EDGE	Selects the clock edge for capturing the externally input Hsync signals.	0
INP_EXT_SYNC_CNT	INP_ENDIAN_ON	Changes the bit endian of the external input.	0
INP_EXT_SYNC_CNT	INP_SWAP_ON	Enables or disables the B/R signal swap of the external input.	0
INP_EXT_SYNC_CNT	INP_HS_INV	Enables or disables the Hsync signal inversion of the external input.	1
INP_EXT_SYNC_CNT	INP_VS_INV	Enables or disables the Hsync signal inversion of the external input.	1
INP_EXT_SYNC_CNT	INP_H_EDGE_SEL	Selects the Hsync reference for BT656 input.	0
INP_EXT_SYNC_CNT	INP_F525_625	Sets the number of lines for BT656 input.	0
INP_EXT_SYNC_CNT	INP_H_POS[1:0]	Sets the data start timing with respect to the Hsync in the BT656/601 format.	0
INP_VSYNC_PH_ADJ	INP_FH50[9:0]	Sets the 1/2fH phase in clock cycle units.	858
INP_VSYNC_PH_ADJ	INP_FH25[9:0]	Sets the 1/4fH phase in clock cycle units.	429
INP_DLY_ADJ	INP_VS_DLY_L[2:0]	Sets the number of lines for delaying the Vsync signal and field differentiation signal.	0
INP_DLY_ADJ	INP_VS_DLY[7:0]	Sets the amount of delay of the Vsync signal in clock cycle units.	0
INP_DLY_ADJ	INP_HS_DLY[7:0]	Sets the amount of delay of the Hsync signal in clock cycle units.	0
INP_DLY_ADJ	INP_FLD_DLY[7:0]	Sets the amount of delay of the field differentiation signal in clock units.	0

Note: Some registers require, after they are set, that the INP_EXT_UPDATE and INP_IMG_UPDATE bits in INP_UPDATE should be set to 1.

chanhvinh.com

30.3.2 Usage Method of Conversion Color Matrix

Typical data conversion setting examples are shown below.

Table 30.31 Conversion Color Matrix

Register Name	Bit Name	GBR to GBR	GBR to YCbCr	YCbCr to GBR	YCbCr to YCbCr
IMGCNT_MTX_MODE	IMGCNT_MTX_MD[1:0]	0	1	2	3
IMGCNT_MTX_YG_ADJ0	IMGCNT_MTX_YG[7:0]	128	128	128	128
IMGCNT_MTX_YG_ADJ0	IMGCNT_MTX_GG[10:0]	256	150	256	256
IMGCNT_MTX_YG_ADJ1	IMGCNT_MTX_GB[10:0]	0	29	1960	0
IMGCNT_MTX_YG_ADJ1	IMGCNT_MTX_GR[10:0]	0	77	1865	0
IMGCNT_MTX_CBB_ADJ0	IMGCNT_MTX_B[7:0]	128	128	128	128
IMGCNT_MTX_CBB_ADJ0	IMGCNT_MTX_BG[10:0]	0	1963	256	0
IMGCNT_MTX_CBB_ADJ1	IMGCNT_MTX_BB[10:0]	256	128	454	256
IMGCNT_MTX_CBB_ADJ1	IMGCNT_MTX_BR[10:0]	0	2005	0	0
IMGCNT_MTX_CRR_ADJ0	IMGCNT_MTX_R[7:0]	128	128	128	128
IMGCNT_MTX_CRR_ADJ0	IMGCNT_MTX_RG[10:0]	0	1941	256	0
IMGCNT_MTX_CRR_ADJ1	IMGCNT_MTX_RB[10:0]	0	2027	0	0
IMGCNT_MTX_CRR_ADJ1	IMGCNT_MTX_RR[10:0]	256	128	359	256

Note: The registers require, after they are set, that the IMGCNT_VEN bit in IMGCNT_UPDATE should be set to 1.

chanhvinh.com

31. Video Display Controller 5 (3): Scaler

31.1 Scaler

31.1.1 Overview of Functions

The scaler subjects the YCbCr and RGB signals output from the input controller, to sync signal generation; and reduction, enlargement, and rotation of the images.

The scaler also records video image in the frame buffer.

In scaler 0, either enlargement process or graphics 0 process can be used.

The functional block diagram of scaler 0 is shown below.

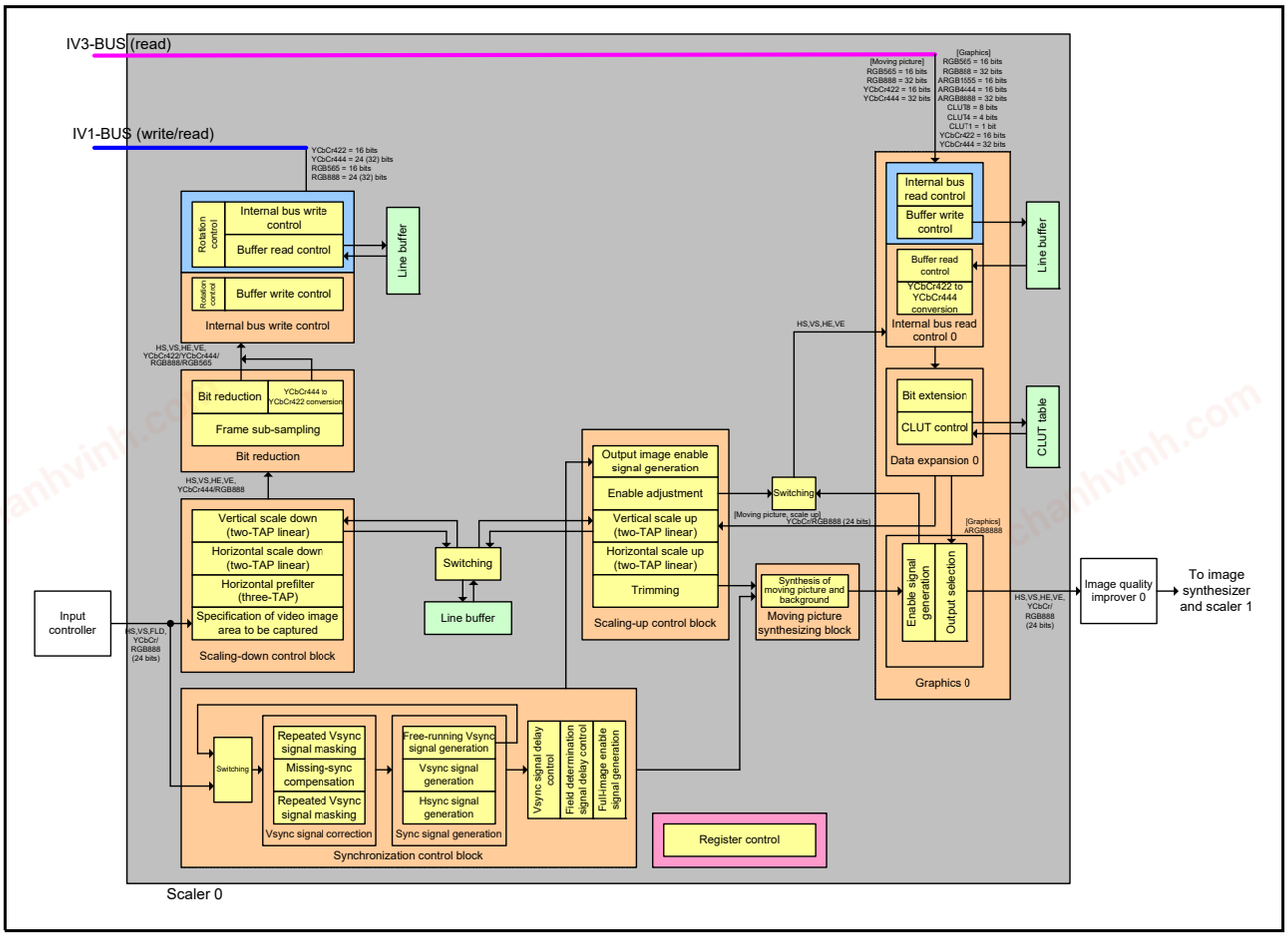


Figure 31.1 Functional Block Diagram of Scaler 0

The registers and bits in the scaler are named SC0_xxxx and those in the graphics blocks are named GR0_xxxx, but in this section, they are collectively called SC_xxxx or GR_xxxx.

chanhvinh.com

31.1.2 Register Control

(1) Updating Registers

The Vsync signal is used to control the update timing of all the registers of the scaling and graphics blocks except some registers of the sync control block and some of the other blocks.

After 1 is set to the bits in the update control register, the contents of the relevant registers are modified at the rising edge of the Vsync signal. The update control register is automatically cleared to 0 after the modification.

Table 31.1 Register Update Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_UPDATE	SC_SCL0_UPDATE	0	SYNC Control Register Update 0: Registers are not updated. 1: Registers are updated.
SC_SCL0_UPDATE	SC_SCL0_VEN_D	0	Scaling-Up Control and Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
SC_SCL0_UPDATE	SC_SCL0_VEN_C	0	Scaling-Down Control and Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
SC_SCL0_UPDATE	SC_SCL0_VEN_B	0	Synchronization Control and Scaling-up Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
SC_SCL0_UPDATE	SC_SCL0_VEN_A	0	Scaling-Down Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
SC_SCL1_UPDATE	SC_SCL1_UPDATE_B	0	Frame Buffer Write Control Register Update 0: Registers are not updated. 1: Registers are updated.
SC_SCL1_UPDATE	SC_SCL1_UPDATE_A	0	Frame Buffer Write Control Register Update 0: Registers are not updated. 1: Registers are updated.
SC_SCL1_UPDATE	SC_SCL1_VEN_B	0	Frame Buffer Write Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
SC_SCL1_UPDATE	SC_SCL1_VEN_A	0	Frame Buffer Write Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
GR_UPDATE	GR_UPDATE	0	Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated.
GR_UPDATE	GR_P_VEN	0	Graphics Display Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
GR_UPDATE	GR_IBUS_VEN	0	Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

The registers controlled by SC_SCL0_VEN_A, SC_SCL0_VEN_C, SC_SCL1_VEN_A, and SC_SCL1_VEN_B are modified at the rising edge of the input Vsync signal.

The registers controlled by SC_SCL0_VEN_B, SC_SCL0_VEN_D, GR_P_VEN, and GR_IBUS_VEN are modified at the rising edge of the output Vsync signal.

chanhvinh.com

31.1.3 Synchronization Control

(1) Selecting Vsync Signal

The Vsync signal to be output from the scaler can be selected.

When an external input signal is to be displayed, an external input Vsync signal should be selected to be output.

When an external input signal is not provided, a free-running Vsync signal should be selected to be output.

Table 31.2 Vsync Signal Selection Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_FRC3	SC_RES_VS_SEL	1	Vsync Signal Output Select 0: External input Vsync signal 1: Internally generated free-running Vsync signal

(2) Masking Repeated Vsync Signals

It is possible to prevent receiving the Vsync signal with a period shorter than the standard period. This is achieved by setting the start timing to receive the next Vsync signal after receiving an input Vsync signal.

The Vsync signal reception masking period is set with the SC_RES_VMASK[15:0] bits.

Masking period [usec] = SC_RES_VMASK × 128 ÷ pixel clock frequency [MHz]

This function is enabled or disabled by the SC_RES_VMASK_ON bit.

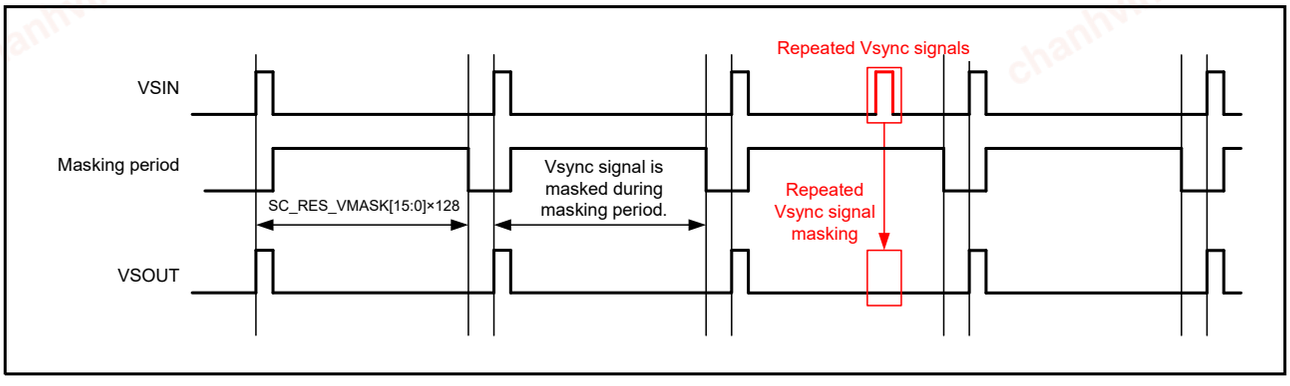


Figure 31.2 Timing for Masking Repeated Vsync Signals

Table 31.3 Repeated Vsync Signal Mask Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_FRC1	SC_RES_VMASK_ON	1	Repeated Vsync Signal Masking Control 0: Repeated Vsync signal masking control is disabled. 1: Repeated Vsync signal masking control is enabled.
SC_SCL0_FRC1	SC_RES_VMASK[15:0]	2800	Repeated Vsync Signal Masking Period Sets the repeated Vsync signal masking period beginning at a Vsync signal in terms of 128 pixel-clock periods. Masking period [usec] = SC_RES_VMASK × 128 ÷ pixel clock frequency [MHz]

chanhvinh.com

(3) Compensating for Missing Vsync Signals

It is possible to prevent output of the Vsync signal with a period longer than the standard period. This is achieved by setting the wait time after reception of an input Vsync signal until reception of the next Vsync signal.

If no Vsync signals are received during the wait time, an internally generated sync signal is inserted.

The wait time can be set using the SC_RES_VLACK[15:0] bits.

Wait time [usec] = SC_RES_VLACK × 128 ÷ pixel clock frequency [MHz]

This function is enabled or disabled by the SC_RES_VLACK_ON bit.

If no Vsync signals are input during the Vsync signal reception time, the SC_RES_QVLACK bit is set to the high level.

If Vsync signals are continuously detected four or more times during the Vsync signal reception time, the SC_RES_QVLOCK bit is set to the high level.

The SC_RES_QVLOCK bit is valid even when both the SC_RES_VMASK_ON and SC_RES_VLACK_ON bits are set to turn off the corresponding functions.

Note that, however, the SC_RES_VMASK and SC_RES_VLACK bits must be set correctly.

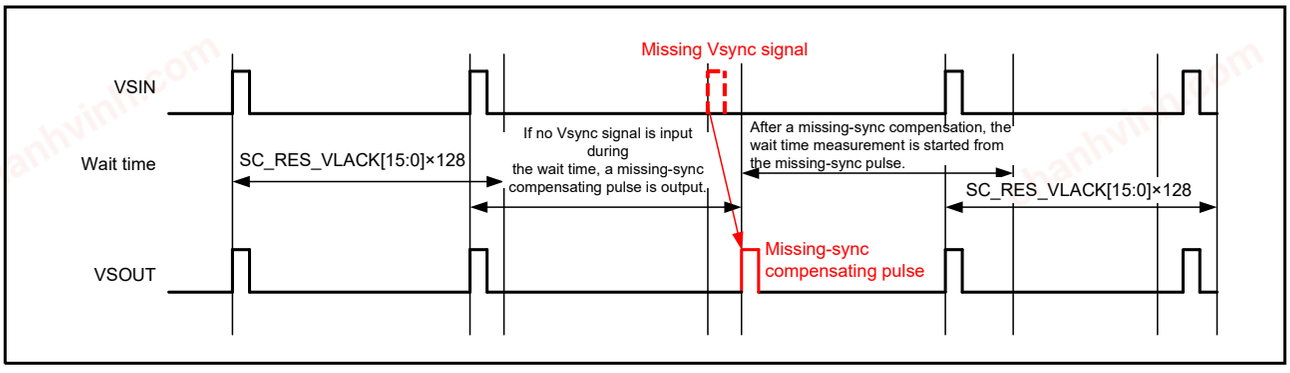


Figure 31.3 Compensation of Missing Vsync Signals

chanhvinh.com

Table 31.4 Missing Vsync Compensation Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_FRC2	SC_RES_VLACK_ON	1	Missing Vsync Signal Compensation 0: Compensation of missing Vsync signals is disabled. 1: Compensation of missing Vsync signals is enabled.
SC_SCL0_FRC2	SC_RES_VLACK[15:0]	3600	Missing-Sync Compensating Pulse Output Wait Time Sets the wait time before outputting a missing-sync compensating pulse after a Vsync signal. Wait time [usec] = SC_RES_VLACK × 128 ÷ pixel clock frequency [MHz]
SC_SCL0_FRC9	SC_RES_QVLACK	—	Missing Vsync Signal Detection Flag 1: Missing Vsync signal input has been detected. 0: No missing Vsync signal input has been detected.
SC_SCL0_FRC9	SC_RES_QVLOCK	—	Locked Vsync Signal Detection Flag 1: No repeated or missing Vsync signal input has been detected for four or more vertical periods. 0: Repeated or missing Vsync signal input has been detected.

For the Vsync signal, repeated-signal masking is first carried out and then missing-signal compensation is carried out, followed by another repeated-signal masking.

Repetition masking is inserted after missing-Vsync compensation to prevent output of the Vsync signal even in cases such as the input of a Vsync signal immediately after the input of a pulse to compensate for a missing Vsync signal.

On/off control of the missing-Vsync compensation also applies to the second repeated-Vsync masking; and masking period setting of the first repeated-Vsync masking also applies to the second repeated-Vsync masking.

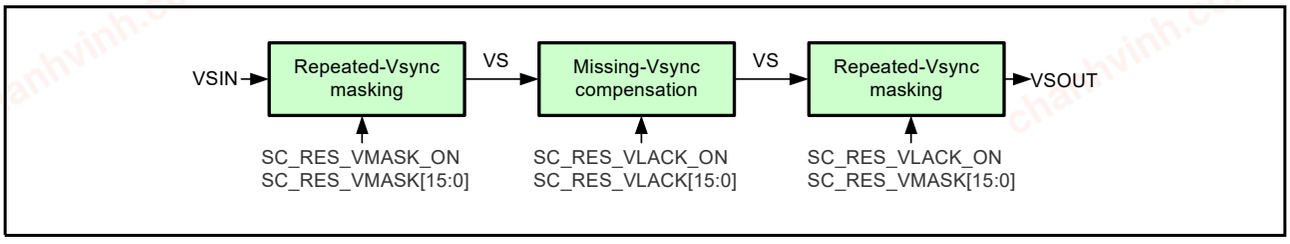


Figure 31.4 Repeated-Vsync Masking and Missing-Vsync Compensation

chanhvinh.com

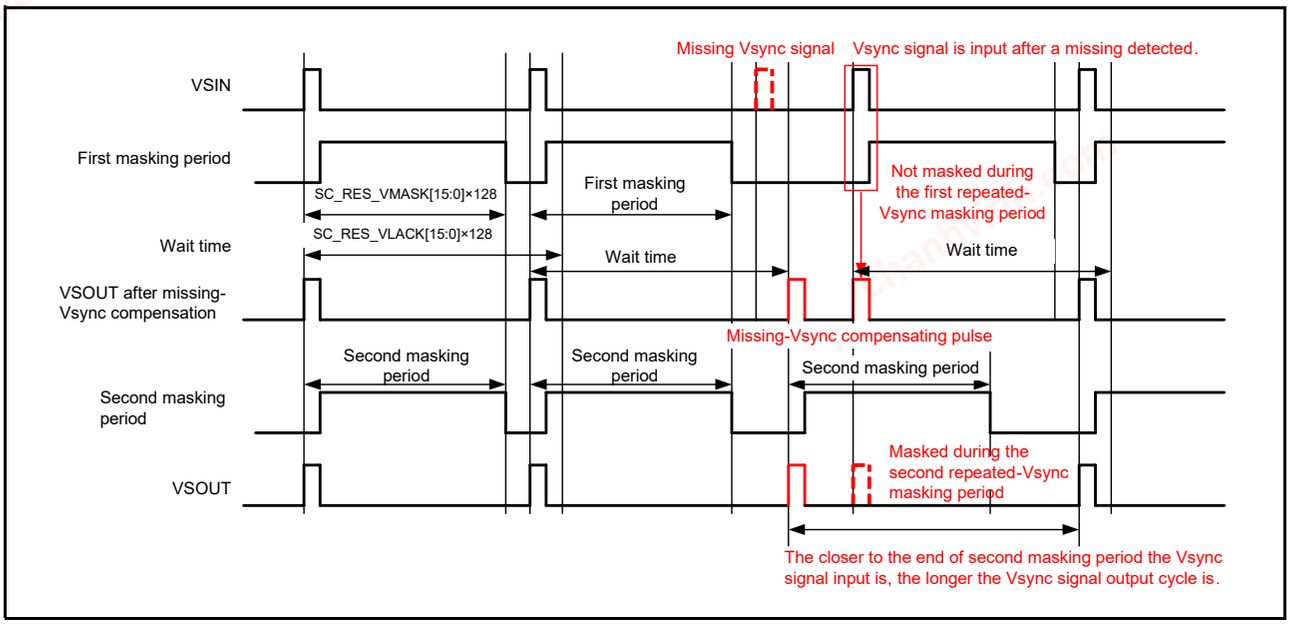


Figure 31.5 Timing for Masking Repeated Vsync Signals and Missing Vsync Signal Compensation

(4) Free-Running Period

Free-running Vsync and Hsync periods can be set.

Hsync period [usec] = (SC_RES_FH + 1) ÷ pixel clock frequency [MHz]

Vsync period [usec] = horizontal period [usec] × (SC_RES_FV + 1)

Table 31.5 Free-Running Period Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_FRC4	SC_RES_FV[10:0]	524	Free-Running Vsync Period Setting Free-running Vsync period = (SC_RES_FV + 1) × horizontal period [usec]
SC_SCL0_FRC4	SC_RES_FH[10:0]	799	Hsync Period Setting Hsync period [usec] = (SC_RES_FH +1) ÷ pixel clock frequency [MHz]

When selecting an external input Vsync signal, set the SC_RES_VS_SEL bit to 0. At this time, the internally generated free-running Vsync signal is not output.

In the meantime, the Hsync signal is always generated according to the free-running signal setting and output from the scaler.

chanhvinh.com

(5) Vsync Signal Delay Control

Delay of Vsync signal output from the scaler can be controlled.

The delay is used to adjust the frame buffer read timing.

Table 31.6 Vsync Output Delay Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_FRC5	SC_RES_VSDLY[7:0]	1	Vsync Signal Delay Control Adjusts the Vsync signal delay in the output Hsync period units. Vsync signal delay [usec]: SC_RES_VSDLY × output Hsync period [usec]

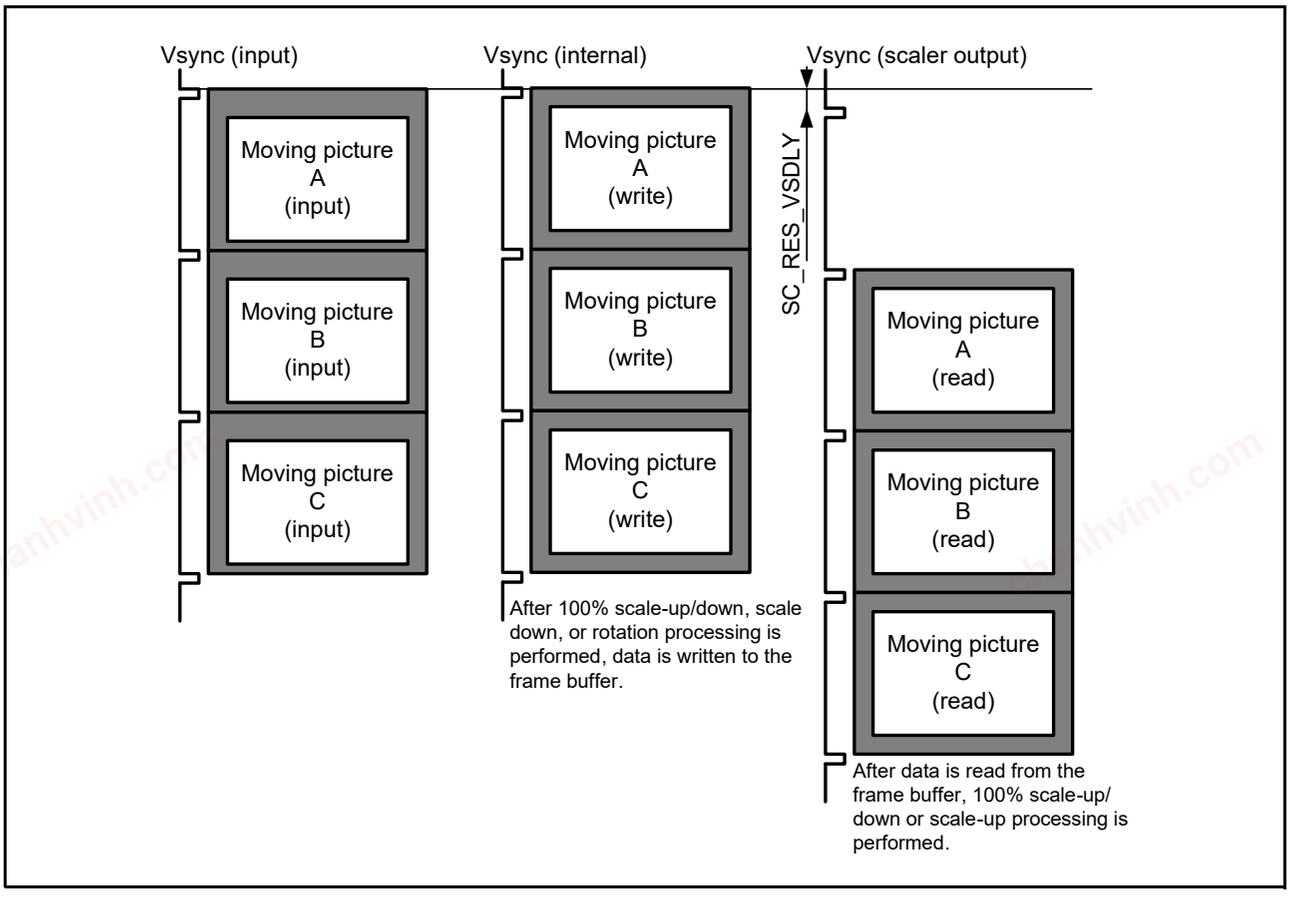


Figure 31.6 Vsync Signal Phases (Two Frame-Buffer Planes Used)

chanhvinh.com

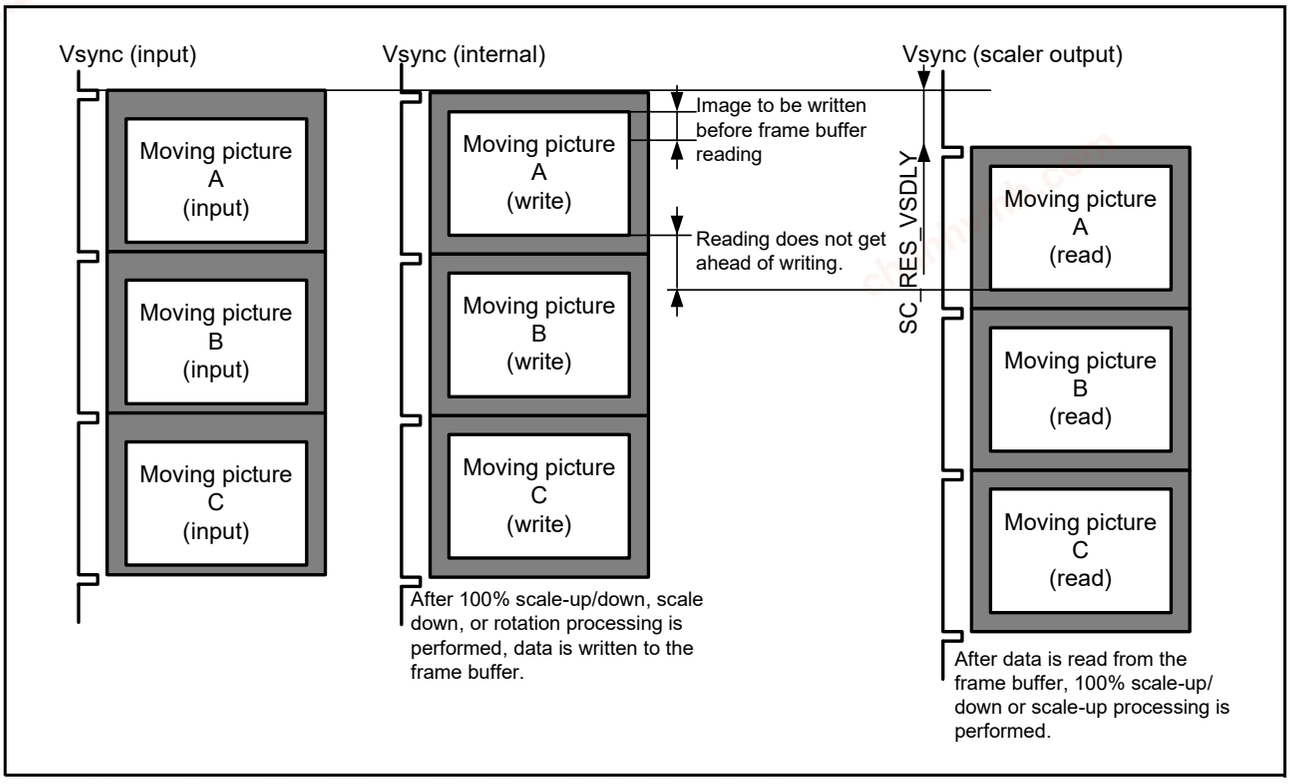


Figure 31.7 Vsync Signal Phases (One Frame-Buffer Plane Used)

31.1.4 Setting Angle of View

(1) Setting Image Area to be Captured

The image area to be captured can be set for reduction or enlargement.

The area is defined by specifying its start position and width based on the input Hsync and Vsync signals.

Table 31.7 Control of Image Area to be Captured

Register Name	Bit Name	Initial Value	Description
SC_SCL0_DS2	SC_RES_VS[10:0]	18	Vertical Position Setting for Video Signal Capturing (VSYNC + (V backporch - 1) lines) Note: The set value should be four or more (lines). SC_RES_VS + SC_RES_VW should be equal to or less than 2039 (lines).
SC_SCL0_DS2	SC_RES_VW[10:0]	240	Vertical Width of Video Signal to be Captured (lines) Note: SC_RES_VS + SC_RES_VW should be equal to or less than 2039 (lines).
SC_SCL0_DS3	SC_RES_HS[10:0]	244	Horizontal Position Setting for Video Signal Capturing (HSYNC + H backporch video-image clock cycles) Note: The set value should be 16 or more (clock cycles). SC_RES_HS + SC_RES_HW should be equal to or less than 2015 (clock cycles).
SC_SCL0_DS3	SC_RES_HW[10:0]	1440	Horizontal Width of Video Signal to be Captured (video-image clock cycles) Note: SC_RES_HS + SC_RES_HW should be equal to or less than 2015 (clock cycles).

chanhvinh.com

(2) Generating a Full-Screen Enable Signal

The valid period of the full screen to be output from the scaler can be set.

The valid period is defined by specifying its start position and width based on the Hsync and Vsync signals output from the scaler.

The vertical front porch should be set to four or more lines, and the horizontal front porch should be 16 or more clock cycles.

Table 31.8 Full-Screen Enable Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_FRC6	SC_RES_F_VS[10:0]	35	Vertical Enable Signal Start Position for Full Screen. (VSYNC + V backporch lines) Note: The set value should be four or more (lines). SC_RES_F_VS + SC_RES_F_VW should be equal to or less than 2039 (lines).
SC_SCL0_FRC6	SC_RES_F_VW[10:0]	480	Vertical Enable Signal Width for Full Screen (lines) Note: SC_RES_F_VS + SC_RES_F_VW should be equal to or less than 2039 (lines).
SC_SCL0_FRC7	SC_RES_F_HS[10:0]	144	Horizontal Enable Signal Start Position for Full Screen. (HSYNC + H backporch pixel-clock cycles) Note: The set value should be 16 or more (clock cycles). SC_RES_F_HS + SC_RES_F_HW should be equal to or less than 2015 (clock cycles).
SC_SCL0_FRC7	SC_RES_F_HW[10:0]	640	Horizontal Enable Signal Width for Full Screen (pixel-clock cycles) Note 1. SC_RES_F_HS + SC_RES_F_HW should be equal to or less than 2015 (clock cycles). Note 2. The set value should be equal to (horizontal signal width for full screen + 2) when serial RGB output is selected as an LCD output signal.

chanhvinh.com

(3) Generating an Image Output Enable Signal

The valid period of the image to be output can be set.

The valid period is defined by specifying its start position and width based on the Hsync and Vsync signals output from the scaler.

Table 31.9 Image Output Enable Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_US2	SC_RES_P_VS[10:0]	35	Vertical Enable Signal Start Position for Output Image. (VSYNC + V backporch lines) Note: The set value should be four or more (lines). SC_RES_P_VS + SC_RES_P_VW should be equal to or less than 2039 (lines).
SC_SCL0_US2	SC_RES_P_VW[10:0]	480	Vertical Enable Signal Width for Output Image (lines) Note: SC_RES_P_VS + SC_RES_P_VW should be equal to or less than 2039 (lines).
SC_SCL0_US3	SC_RES_P_HS[10:0]	144	Horizontal Enable Signal Start Position for Output Image. (HSYNC + H backporch pixel-clock cycles) Note: The set value should be 16 or more (clock cycles). SC_RES_P_HS + SC_RES_P_HW should be equal to or less than 2015 (clock cycles).
SC_SCL0_US3	SC_RES_P_HW[10:0]	640	Horizontal Enable Signal Width for Output Image (pixel-clock cycles) Note: SC_RES_P_HS + SC_RES_P_HW should be equal to or less than 2015 (clock cycles).

chanhvinh.com

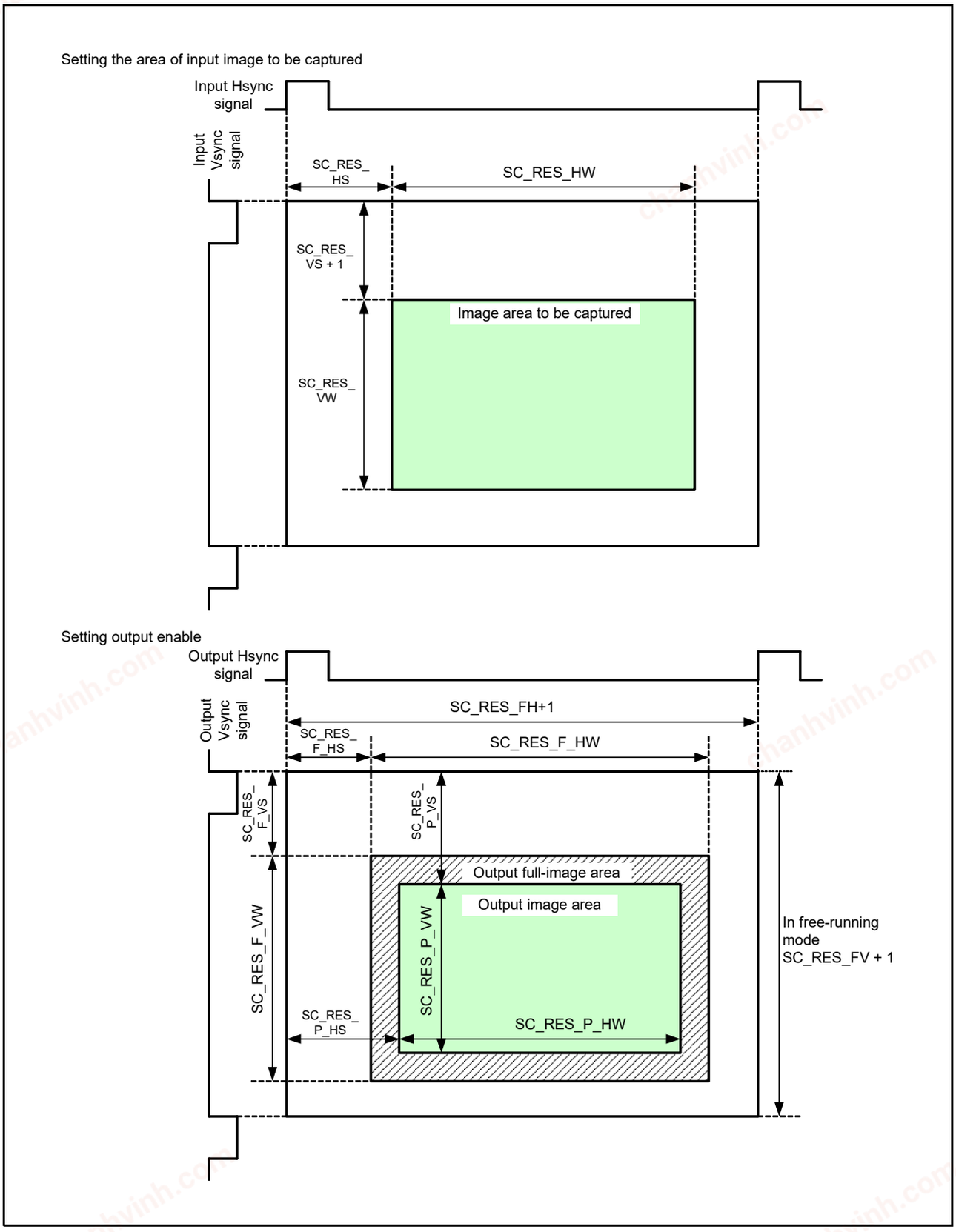


Figure 31.8 Enable Settings

chanhvinh.com

31.1.5 Scaling Settings

(1) Scaling Processing Block

The scaling-down control block scales down the input image from the input controller.

When rotation is required, the scaling-down control block first scales down the image and then rotates it before writing it into the frame buffer.

The scaling-up control block reads the rotated image from the frame buffer and scales it up.

Table 31.10 Rotation and Scaling Process

Rotation	Horizontal Scaling	Vertical Scaling	Scaling-Down Control Block	Scaling-Up Control Block
Normal	Horizontal scale down	Vertical scale down	Horizontal scale down/ vertical scale down	Horizontal 100% scale up/ vertical 100% scale up
	Horizontal scale down	Vertical scale up	Horizontal scale down/ vertical 100% scale up	Horizontal 100% scale up/ vertical scale up
	Horizontal scale up	Vertical scale down	Horizontal 100% scale up/ vertical scale down	Horizontal scale up/ vertical 100% scale up
	Horizontal scale up	Vertical scale up	Horizontal 100% scale up/ vertical 100% scale up	Horizontal scale up/ vertical scale up
Horizontal mirroring	Horizontal scale down	Vertical scale down	Horizontal scale down/ vertical scale down	Horizontal 100% scale up/ vertical 100% scale up
	Horizontal scale down	Vertical scale up	Horizontal scale down/ vertical 100% scale up	Horizontal 100% scale up/ vertical scale-up
	Horizontal scale up	Vertical scale down	Horizontal 100% scale up/ vertical scale down	Horizontal scale up/ vertical 100% scale up
	Horizontal scale up	Vertical scale up	Horizontal 100% scale up/ vertical 100% scale up	Horizontal scale up/ vertical scale up
90° rotation	(Horizontal input → vertical output) scale down	(Vertical input → horizontal output) scale down	Horizontal scale down/ vertical scale down	Horizontal 100% scale up/ vertical 100% scale up
	(Horizontal input → vertical output) scale down	(Vertical input → horizontal output) scale up	Horizontal scale down/ vertical 100% scale up	Horizontal scale up/ vertical 100% scale up
	(Horizontal input → vertical output) scale up	(Vertical input → horizontal output) scale up	Horizontal 100% scale up/ vertical 100% scale up	Horizontal scale up/ vertical scale up
180° rotation	Horizontal scale down	Vertical scale down	Horizontal scale down/ vertical scale down	Horizontal 100% scale up/ vertical 100% scale up
	Horizontal scale down	Vertical scale up	Horizontal scale down/ vertical 100% scale up	Horizontal 100% scale up/ vertical scale up
	Horizontal scale up	Vertical scale down	Horizontal 100% scale up/ vertical scale down	Horizontal scale up/ vertical 100% scale up
	Horizontal scale up	Vertical scale up	Horizontal 100% scale up/ vertical 100% scale up	Horizontal scale up/ vertical scale up
270° rotation	(Horizontal input→vertical output) scale down	(Vertical input→horizontal output) scale down	Horizontal scale down/ vertical scale down	Horizontal 100% scale up/ vertical 100% scale up
	(Horizontal input → vertical output) scale down	(Vertical input → horizontal output) scale up	Horizontal scale down/ vertical 100% scale up	Horizontal scale up/ vertical 100% scale up
	(Horizontal input → vertical output) scale up	(Vertical input → horizontal output) scale up	Horizontal 100% scale up/ vertical 100% scale up	Horizontal scale up/ vertical scale up

chanhvinh.com

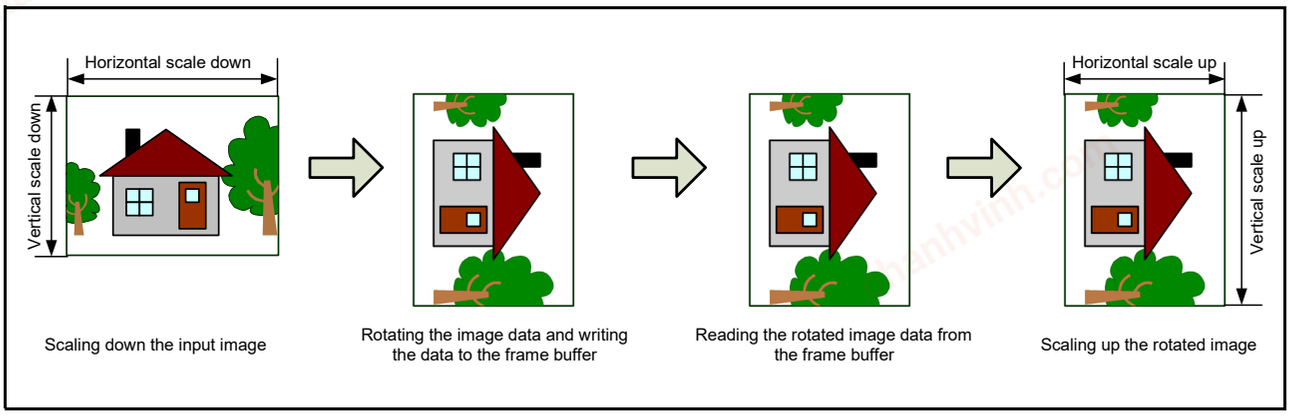


Figure 31.9 Rotation and Scaling Process

It is impossible to use vertical reduction by the scaling-down control block and vertical enlargement by the scaling-up control block simultaneously because they are mutually exclusive. Thus, the following scaling processes cannot be performed with 90° rotation or 270° rotation.

Table 31.11 Impossible Scaling Process

Rotation	Horizontal Scaling	Vertical Scaling	Scaling-Down Control Block	Scaling-Up Control Block
90° rotation 270° rotation	(Horizontal input → vertical output) scale up	(Vertical input → horizontal output) scale down	Horizontal 100% scale up/ vertical scale down	Horizontal 100% scale up/ vertical scale up

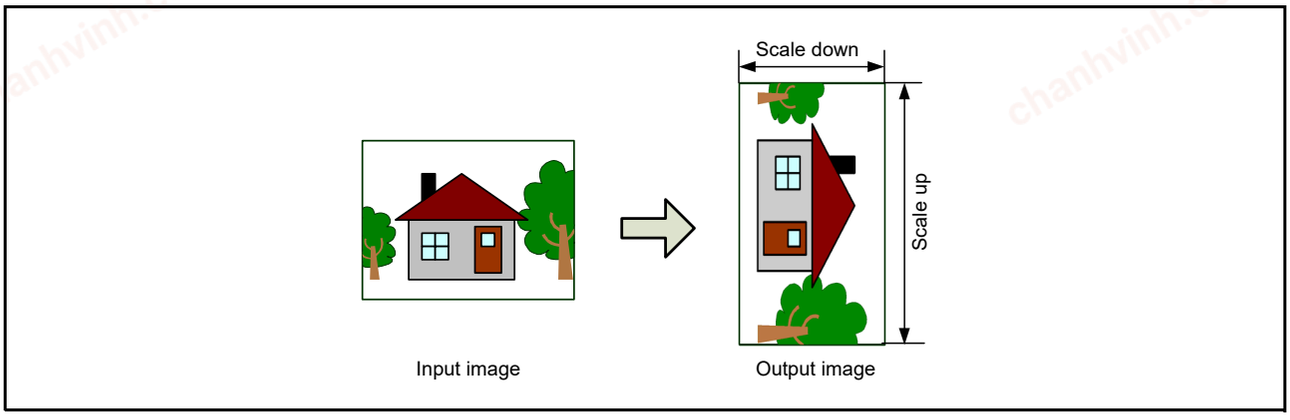


Figure 31.10 Impossible Scaling Process

chanhvinh.com

31.1.6 Horizontal Prefilter

The horizontal prefilter can be turned on or off for brightness (Y) and RGB signals to suppress the frequency band of the signals during horizontal size reduction. The input format depends on the SC_RES_MD[1:0] bit setting in the writing mode register (SC_SCL1_WR1).

When the horizontal reduction ratio is high and there is too much folding frequency component to ignore, the horizontal prefilter should be turned on.

Table 31.12 Horizontal Prefilter Settings

Input Format	SC_RES_PFIL_SEL	Operation
YCbCr input	1	Turns on the horizontal prefilter for Y signal and turns off the horizontal prefilter for Cb/Cr signal.
	0	Turns off the horizontal prefilter.
RGB input	1	Turns on the horizontal prefilter for RGB signal.
	0	Turns off the horizontal prefilter.

Table 31.13 Horizontal Prefilter Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_DS4	SC_RES_PFIL_SEL	0	Prefilter Mode Select for Brightness Signals 0: The prefilter is turned off. 1: The prefilter is turned on. (1/4 + 1/2 + 1/4)

31.1.7 Horizontal Scale-Down

The number of horizontally arranged pixels can be decreased at a desired ratio in the range of 1/1 to 1/8 using pixel conversion.

For the scaling filter, either hold or linear interpolation mode can be selected.

(1) One-TAP Hold Interpolation

When the interpolation position is between input pixels X_n and X_{n+1}, the $X_{interpo}$ interpolation value is defined as follows.

$$X_{interpo} = X_n$$

(2) Two-TAP Linear Interpolation

When the interpolation position is between input pixels X_n and X_{n+1}, the $X_{interpo}$ interpolation value is defined as follows based on the interpolation position "phase".

$$X_{interpo} = (X_n \times (4096 - phase) + X_{n+1} \times phase) / 4096$$

(3) Calculation of Horizontal Scale Down Ratio

The value to be set to the horizontal scale-down ratio SC_RES_DS_H_RATIO can be obtained using the following equation based on the number of input pixels SC_RES_HW and number of output pixels SC_RES_OUT_HW, where the decimals are rounded off.

SC_RES_DS_H_RATIO = round (SC_RES_HW ÷ SC_RES_OUT_HW × 4096)

Note that, for 100% horizontal scale-up, the SC_RES_HW and SC_RES_OUT_HW values should be identical and the SC_RES_DS_H_RATIO bits should be set to 4096.

chanhvinh.com

(4) Handling for Lack of Last-Input Pixel

Interpolation is carried out between the second-last-input and last-input pixels to produce the last-output pixel at the right end of a screen. The interpolation position of the last-output pixel may be close to the second-last-input pixel depending on the horizontal scale-down ratio; in this case, it may appear that the last-input pixel is lacking.

The undesirable influence by lack of last-input pixel can be decreased by appropriately adjusting the horizontal scale-down ratio using the following equations.

Pre-adjustment horizontal scale-down ratio RATIO_org should be calculated first to find adjustment value σ, and then scale-down ratio SC_RES_DS_H_RATIO should be determined.

RATIO_org = round (SC_RES_HW ÷ SC_RES_OUT_HW × 4096)

σ = (RATIO_org × (SC_RES_OUT_HW − 1) − (SC_RES_HW − 1) × 4096) ÷ (SC_RES_OUT_HW − 1)

SC_RES_DS_H_RATIO = roundup (RATIO_org − σ)

Table 31.14 Horizontal Scale Down Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_DS1	SC_RES_DS_H_ON	1	Horizontal Scale Down On/Off 0: Off 1: On
SC_SCL0_DS7	SC_RES_OUT_HW[10:0]	640	Number of Valid Horizontal Pixels Output by Scaling-down Control Block (Video-image clock cycles)
SC_SCL0_DS4	SC_RES_DS_H_INTERPOTYP	1	Horizontal Interpolation Mode Select 0: Hold interpolation 1: Linear interpolation
SC_SCL0_DS4	SC_RES_DS_H_RATIO[15:0]	9224	Horizontal Scale Down Ratio ([15:12]: Integer part, [11:0]: Decimal part) round(SC_RES_HW ÷ SC_RES_OUT_HW × 4096) SC_RES_DS_H_RATIO < 4096: Setting prohibited SC_RES_DS_H_RATIO = 4096: 100% scale up SC_RES_DS_H_RATIO > 4096: Scale down

Note: The SC_RES_OUT_HW value should be aligned in 4-pixel units and equal to or smaller than the SC_RES_HW value.

chanhvinh.com

31.1.8 Vertical Scale-Down

The number of lines can be decreased in the vertical direction at a desired ratio in the range of 1/1 to 1/8 using pixel conversion.

For the scaling filter, either hold or linear interpolation mode can be selected.

(1) One-TAP Hold Interpolation

When the interpolation position is between input lines X_n and X_{n+1}, the $X_{interpo}$ interpolation value is defined as follows.

$$X_{interpo} = X_n$$

(2) Two-TAP Linear Interpolation

When the interpolation position is between input lines X_n and X_{n+1}, the $X_{interpo}$ interpolation value is defined as follows based on the interpolation position "phase".

$$X_{interpo} = (X_n \times (4096 - phase) + X_{n+1} \times phase) / 4096$$

(3) Calculation of Vertical Scale Down Ratio

The value to be set to the vertical scale-down ratio SC_RES_V_RATIO can be obtained using the following equation based on the number of input lines SC_RES_VW and number of output lines SC_RES_OUT_VW, where the decimals are rounded off.

$$SC_RES_V_RATIO = round\ (SC_RES_VW \div SC_RES_OUT_VW \times 4096)$$

Note that the SC_RES_VW and SC_RES_OUT_VW values should be identical for vertical enlargement or 100% vertical enlargement.

For 100% vertical enlargement, reduction is carried out assuming SC_RES_V_RATIO as 4096.

(4) Handling for Lack of Last-Input Line

Interpolation is carried out between the second-last-input and last-input lines to produce the last-output line at the lower end of a screen. The interpolation position of the last-output line may be close to the second-last-input line depending on the vertical scale-down ratio; in this case, it may appear that the last-input line is lacking.

The undesirable influence by the lack of last-input line can be decreased by appropriately adjusting the vertical scale-down ratio using the following equations.

Pre-adjustment vertical scale-down ratio RATIO_org should be calculated first to find adjustment value σ, and then scale-down ratio SC_RES_V_RATIO should be determined.

$$RATIO_org = round\ (SC_RES_VW \div SC_RES_OUT_VW \times 4096)$$

$$\sigma = (RATIO_org \times (SC_RES_OUT_VW - 1) - (SC_RES_VW - 1) \times 4096) \div (SC_RES_OUT_VW - 1)$$

$$SC_RES_V_RATIO = round\ (RATIO_org - \sigma)$$

chanhvinh.com

Table 31.15 Vertical Scale Down Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_DS1	SC_RES_DS_V_ON	1	Vertical Scale Down On/Off 0: Off 1: On
SC_SCL0_DS7	SC_RES_OUT_VW [10:0]	240	Number of Valid Lines in Vertical Direction Output by Scaling-Down Control Block (lines) This bit setting is used for the number of lines to be written to the frame buffer. When SC_SCL1_WR1.SC_RES_LOOP is 0 (frame write mode), these bits specify the number of lines for one frame. When SC_SCL1_WR1.SC_RES_LOOP is 1 (line write mode), these bits specify the number of lines for writing in a ring configuration.
SC_SCL0_DS5	SC_RES_V_ INTERPOTYP	1	Vertical Interpolation Mode Select 0: Hold interpolation 1: Linear interpolation
SC_SCL0_DS6	SC_RES_V_RATIO [15:0]	2044	Vertical Scale UP/Down Ratio ([15:12]: Integer part, [11:0]: Decimal part) For scale down: round(SC_RES_VW ÷ SC_RES_OUT_VW × 4096) For scale up: round(SC_RES_IN_VW ÷ SC_RES_P_VW × 4096) SC_RES_V_RATIO < 4096: Scale up SC_RES_V_RATIO = 4096: 100% scale up SC_RES_V_RATIO > 4096: Scale down

Note: SC_RES_V_RATIO and SC_RES_V_INTERPOTYP are both shared by vertical reduction and vertical enlargement.
It is impossible to use vertical reduction and vertical enlargement simultaneously because they are mutually exclusive.
The SC_RES_OUT_VW value should be aligned in 4-line units and equal to or smaller than the SC_RES_VW value.

chanhvinh.com

31.1.9 Horizontal Scale Up

The number of horizontally arranged pixels can be increased at a desired ratio in the range of 1/1 to 8/1 using pixel conversion.

For the scaling filter, either hold or linear interpolation mode can be selected.

(1) One-TAP Hold Interpolation

When the interpolation position is between input pixels X_n and X_{n+1}, the $X_{interpo}$ interpolation value is defined as follows.

$$X_{interpo} = X_n$$

(2) Two-TAP Linear Interpolation

When the interpolation position is between input pixels X_n and X_{n+1}, the $X_{interpo}$ interpolation value is defined as follows based on the interpolation position "phase".

$$X_{interpo} = (X_n \times (4096 - phase) + X_{n+1} \times phase) / 4096$$

(3) Calculation of Horizontal Scale Up Ratio

The value to be set to the horizontal scale-up ratio SC_RES_US_H_RATIO can be obtained using the following equation based on the number of input pixels SC_RES_IN_HW and number of output pixels SC_RES_P_HW, where the decimals are rounded off.

$$SC_RES_US_H_RATIO = round (SC_RES_IN_HW \div SC_RES_P_HW \times 4096)$$

Note that, for 100% horizontal scale-up, the SC_RES_IN_HW and SC_RES_P_HW values should be identical and the SC_RES_US_H_RATIO bits should be set to 4096.

(4) Folding Handling

Since interpolation is carried out between the last-input pixel and second-last-input folding pixel to produce the last-output pixel at the right end of a screen, folding may undesirably stand out depending on the horizontal scale up ratio.

The undesirable influence by folding pixels can be decreased by appropriately adjusting the horizontal scale-up ratio using the following equations.

Pre-adjustment horizontal scale-up ratio RATIO_org should be calculated first to find adjustment value σ, and then scale-up ratio SC_RES_US_H_RATIO should be determined.

$$RATIO_org = round (SC_RES_IN_HW \div SC_RES_P_HW \times 4096)$$

$$\sigma = (RATIO_org \times (SC_RES_P_HW - 1) - (SC_RES_IN_HW - 1) \times 4096 \div (SC_RES_P_HW - 1)$$

$$SC_RES_US_H_RATIO = round (RATIO_org - \sigma)$$

chanhvinh.com

Table 31.16 Horizontal Scale Up Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_US1	SC_RES_US_H_ON	1	Horizontal Scale Up On/Off 0: Off 1: On
SC_SCL0_US4	SC_RES_IN_HW[10:0]	640	Number of Valid Horizontal Pixels Input to Scaling-up Control Block (Pixel-clock cycles)
SC_SCL0_US6	SC_RES_US_H_ INTERPOTYP	1	Horizontal Interpolation Mode Select 0: Hold interpolation 1: Linear interpolation
SC_SCL0_US5	SC_RES_US_H_RATIO [15:0]	9224	Horizontal Scale Up Ratio ([15:12]: Integer part, [11:0]: Decimal part) round(SC_RES_IN_HW ÷ SC_RES_P_HW × 4096) SC_RES_US_H_RATIO < 4096: Scale up SC_RES_US_H_RATIO = 4096: 100% scale-up SC_RES_US_H_RATIO > 4096: Setting prohibited

31.1.10 Vertical Scale-Up

The number of lines can be increased in the vertical direction at a desired ratio in the range of 1/1 to 8/1 using pixel conversion.

For the scaling filter, either hold or linear interpolation mode can be selected.

(1) One-TAP Hold Interpolation

When the interpolation position is between input lines X_n and X_{n+1}, the $X_{interpo}$ interpolation value is defined as follows.

$$X_{interpo} = X_n$$

(2) Two-TAP Linear Interpolation

When the interpolation position is between input lines X_n and X_{n+1}, the $X_{interpo}$ interpolation value is defined as follows based on the interpolation position "phase".

$$X_{interpo} = (X_n \times (4096 - phase) + X_{n+1} \times phase) / 4096$$

(3) Calculation of Vertical Scale Up Ratio

The value to be set to the vertical scale-up ratio SC_RES_V_RATIO can be obtained using the following equation based on the number of input lines SC_RES_IN_VW and number of output lines SC_RES_P_VW, where the decimals are rounded off.

SC_RES_V_RATIO = round (SC_RES_IN_VW ÷ SC_RES_P_VW × 4096)

Note that, for 100% vertical enlargement or vertical reduction, the SC_RES_IN_VW and SC_RES_P_VW values should be identical.

chanhvinh.com

(4) Folding Handling

The last line to be output at the bottom of the screen is produced by interpolation between the last line and line for folding (second-last line to be input). According to the vertical scale-up rate, this may cause folding to stand out.

The undesirable influence by folding lines can be decreased by appropriately adjusting the vertical scale-up ratio using the following equations.

Pre-adjustment vertical scale-up ratio RATIO_org should be calculated first to find adjustment value σ, and then scale-up ratio SC_RES_V_RATIO should be determined.

RATIO_org = round (SC_RES_IN_VW ÷ SC_RES_P_VW × 4096)

σ = (RATIO_org × (SC_RES_P_VW − 1) − (SC_RES_IN_VW − 1) × 4096) ÷ (SC_RES_P_VW − 1)

SC_RES_V_RATIO = round (RATIO_org − σ)

Table 31.17 Vertical Scale Up Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_US1	SC_RES_US_V_ON	1	Vertical Scale Up On/Off 0: Off 1: On
SC_SCL0_US4	SC_RES_IN_VW[10:0]	240	Number of Valid Lines in Vertical Direction Input to Scaling-up Control Block (Lines)
SC_SCL0_DS5	SC_RES_V_INTERPOTYP	1	Vertical Interpolation Mode Select 0: Hold interpolation 1: Linear interpolation
SC_SCL0_DS6	SC_RES_V_RATIO[15:0]	2044	Vertical Scale Up Ratio ([15:12]: Integer part, [11:0]: Decimal part) For scale down: round(SC_RES_VW ÷ SC_RES_OUT_VW × 4096) For scale up: round(SC_RES_IN_VW ÷ SC_RES_P_VW × 4096) SC_RES_V_RATIO < 4096: Scale up SC_RES_V_RATIO = 4096: 100% scale up SC_RES_V_RATIO > 4096: Scale down

Note: SC_RES_V_RATIO and SC_RES_V_INTERPOTYP are both shared by vertical reduction and vertical enlargement.
It is impossible to use vertical reduction and vertical enlargement simultaneously because they are mutually exclusive.

chanhvinh.com

31.1.11 IP Conversion

(1) Initial Phase Control

When interlaced signals are input, line flickering caused by the line offset between the top and bottom fields can be decreased before being displayed by independently adjusting the initial scaling phases of the fields.

For various operations, appropriate settings should be made referring to the relevant registers as listed in Table 31.18.

Table 31.18 Initial Scaling Phase Settings (Standard Values) for IP Conversion

Rotation	Horizontal Scaling	Vertical Scaling	Reference Bit (Setting)
Normal	Horizontal scale down	Vertical scale down	SC_RES_TOP_INIPHASE = 2048
	Horizontal scale down	Vertical scale up	SC_RES_TOP_INIPHASE = 2048
	Horizontal scale up	Vertical scale down	SC_RES_TOP_INIPHASE = 2048
	Horizontal scale up	Vertical scale up	SC_RES_TOP_INIPHASE = 2048
Horizontal mirroring	Horizontal scale down	Vertical scale down	SC_RES_TOP_INIPHASE = 2048
	Horizontal scale down	Vertical scale up	SC_RES_TOP_INIPHASE = 2048
	Horizontal scale up	Vertical scale down	SC_RES_TOP_INIPHASE = 2048
	Horizontal scale up	Vertical scale up	SC_RES_TOP_INIPHASE = 2048
90° rotation	(Horizontal input → vertical output) scale down	(Vertical input → horizontal output) scale down	SC_RES_TOP_INIPHASE = 2048
	(Horizontal input → vertical output) scale down	(Vertical input → horizontal output) scale up	SC_RES_TOP_INIPHASE = 2048
	(Horizontal input → vertical output) scale up	(Vertical input → horizontal output) scale up	SC_RES_US_HB_INIPHASE = 2048
180° rotation	Horizontal scale down	Vertical scale down	SC_RES_TOP_INIPHASE = 2048
	Horizontal scale down	Vertical scale up	SC_RES_BTM_INIPHASE = 2048
	Horizontal scale up	Vertical scale down	SC_RES_TOP_INIPHASE = 2048
	Horizontal scale up	Vertical scale up	SC_RES_BTM_INIPHASE = 2048
270° rotation	(Horizontal input → vertical output) scale down	(Vertical input → horizontal output) scale down	SC_RES_TOP_INIPHASE = 2048
	(Horizontal input → vertical output) scale down	(Vertical input → horizontal output) scale up	SC_RES_TOP_INIPHASE = 2048
	(Horizontal input → vertical output) scale up	(Vertical input → horizontal output) scale up	SC_RES_US_HT_INIPHASE = 2048

Note: Set 0 to the initial phase control registers where the specific value is not shown in the table.
Set 0 to all the initial phase control registers when progressive signals are input.

Table 31.19 Initial Scaling Phase Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_DS5	SC_RES_BTM_INIPHASE [11:0]	0	Vertical Interpolation Start Phase for Bottom Field 0 to 4095 (0 to approx. 1.0)
SC_SCL0_DS5	SC_RES_TOP_INIPHASE [11:0]	2048	Vertical Interpolation Start Phase for Top Field 0 to 4095 (0 to approx. 1.0)
SC_SCL0_US6	SC_RES_US_HB_INIPHASE [11:0]	0	Horizontal Interpolation Start Phase for Bottom Field 0 to 4095 (0 to approx. 1.0)
SC_SCL0_US6	SC_RES_US_HT_INIPHASE [11:0]	0	Horizontal Interpolation Start Phase for Top Field 0 to 4095 (0 to approx. 1.0)

chanhvinh.com

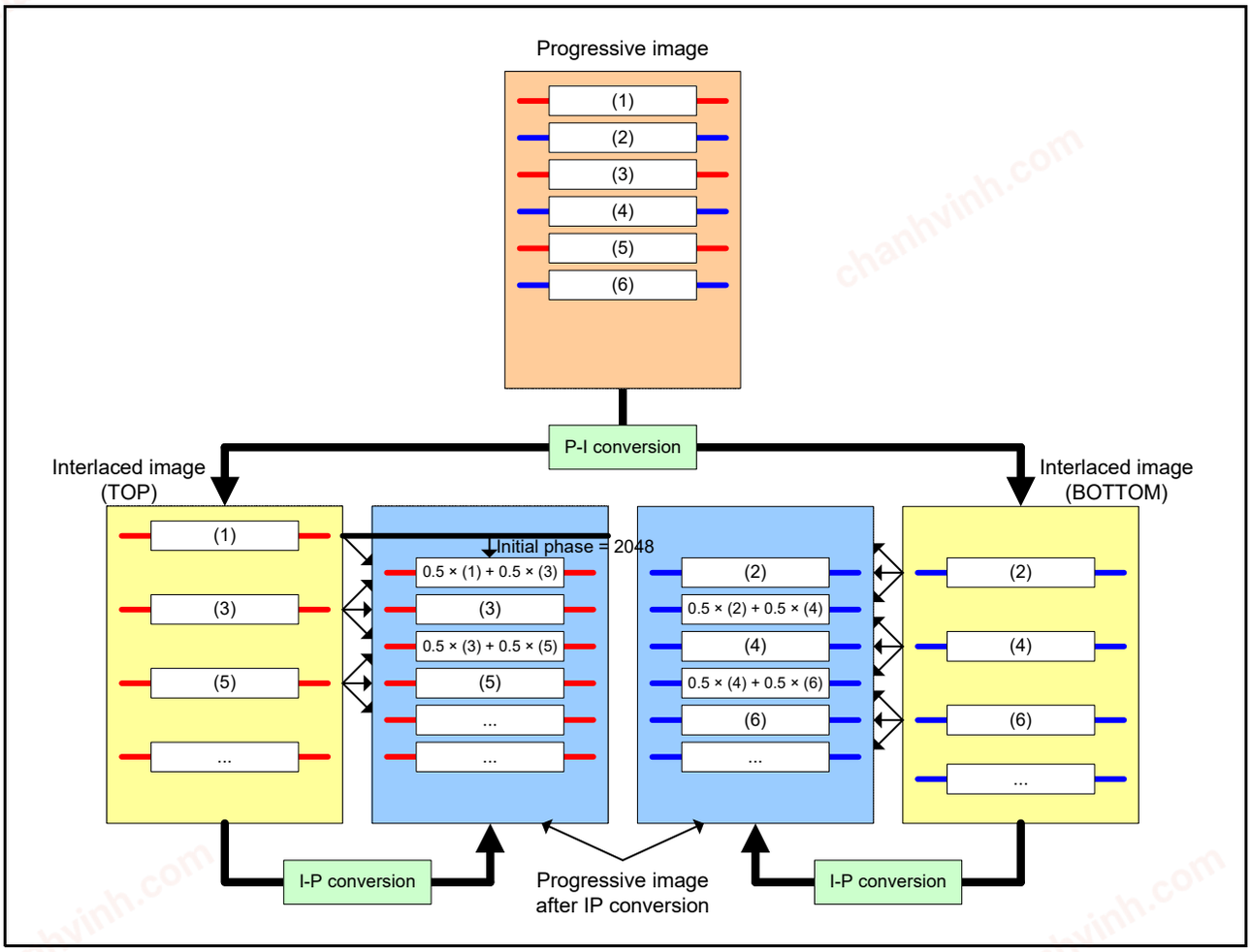


Figure 31.11 IP Conversion Processing Schematic Diagram

(2) Field Determination Signal Control

When interlaced signals are input, the field determination signal can be controlled, which is output to the scaling-up control block during vertical scaling.

When progressive signals are input or vertical scaling is carried out by the scaling-down control block, the field determination signal output to the scaling-up control block is fixed to the specific level, and thus either 0 or 1 can be set to the SC_RES_FLD_DLY_SEL bit.

chanhvinh.com

Table 31.20 Settings for Field Determination Signal Control

Input Signal	Rotation	Vertical Processing	Frame Buffer	SC_RES_FLD_DLY_SEL
Progressive	—	—	—	—
Interlace	Normal Horizontal mirroring 180° rotation	Vertical scale down	—	—
		Vertical scale up	One plane or less	0
			Two planes or more	1
	90° rotation 270° rotation	(Horizontal input → vertical output) scale down	—	—
		(Horizontal input → vertical output) scale up	Two planes or more	1

Table 31.21 Field Determination Signal Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_FRC5	SC_RES_FLD_DLY_SEL	1	Field Determination Signal Delay Control 0: No delay 1: Delay of one vertical cycle

31.1.12 Control of Interrupt on Specified Image Line before Scaling-down, and Reading of Current Image Line before Scaling-down

When the location of the image line input to the scaling-down control block matches the SC_SCL1_LINE setting, an interrupt processing is done. In addition, the current location of the line input to the scaling-down control block can be read from a register.

Table 31.22 Control of Interrupt on Specified Image Line before Scaling-down, and Reading of Current Image line before Scaling-down

Register Name	Bit Name	Initial Value	Description
SC_SCL0_INT	SC_RES_LINE[10:0]	All 0	Setting of Interrupt on Image Line Input to Scaling-down Control Block When the location of the image line input to the scaling-down control block matches the SC_SCL0_LINE setting, an interrupt signal is output. (Setting prohibited in this product)
SC_SCL0_MON0	SC_RES_LIN_STAT[10:0]	All 0	Current Location of Image Line Input to Scaling-down Control Block

chanhvinh.com

31.1.13 Trimming

The upper, lower, right, and left parts of a post-scaling image can be trimmed off as specified by the SC_RES_VCUT and SC_RES_HCUT bits before being output.

The frame lines of the post-scaling image can also be displayed by setting the SC_RES_DISP_ON bit to 1.

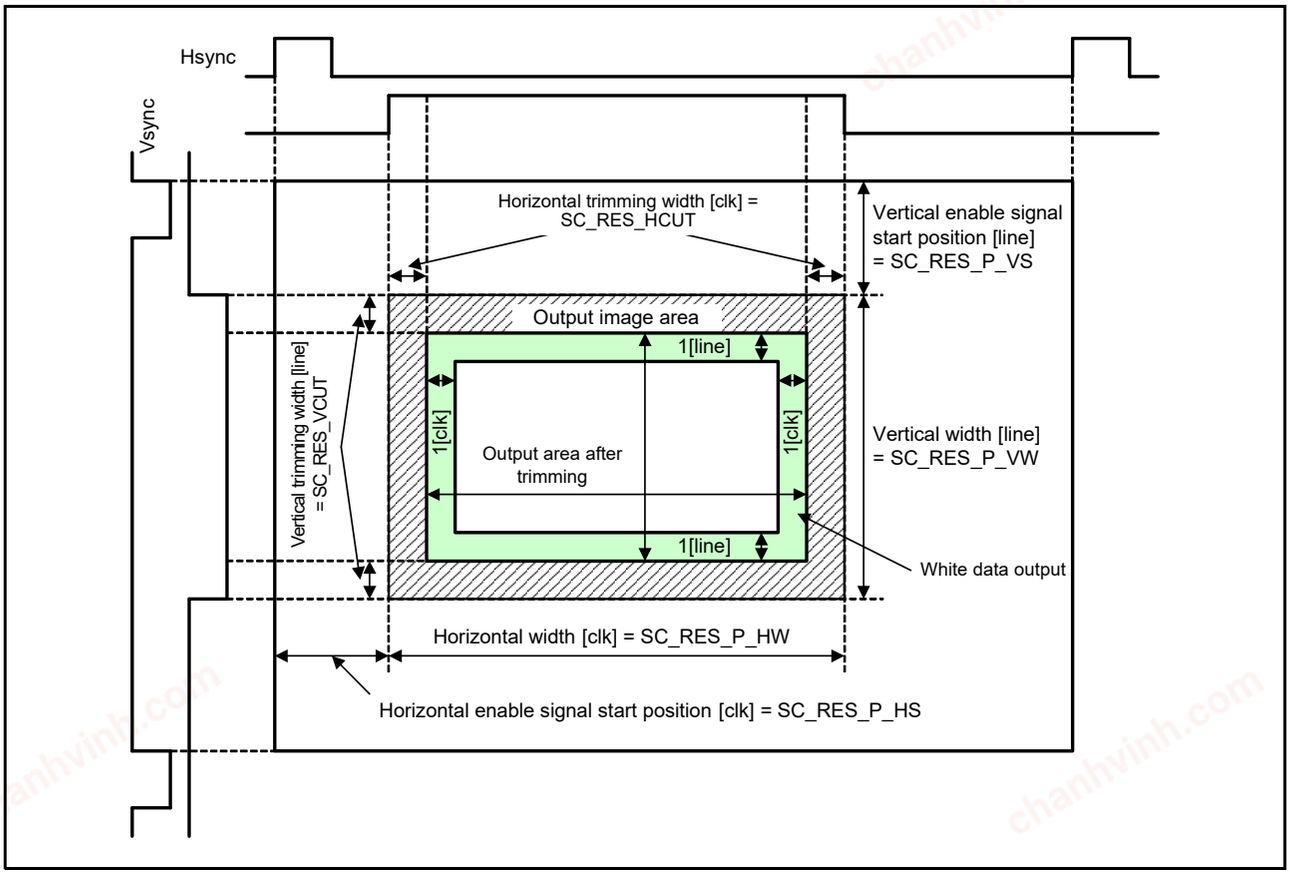


Figure 31.12 Area Relationship for Trimming (Frame Lines Displayed)

Table 31.23 Trimming Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_US7	SC_RES_HCUT[7:0]	0	Horizontal Amount of Cut-off Post-Scaling Image (Right and Left Parts) Sets the number of pixel-clock cycles.
SC_SCL0_US7	SC_RES_VCUT[7:0]	0	Vertical Amount of Cut-off Post-Scaling Image (Upper and Lower Parts) Sets the number of lines.
SC_SCL0_US8	SC_RES_DISP_ON	0	Post-Scaling Image Frame Display On/Off 0: Frame display on 1: Frame display off

chanhvinh.com

31.1.14 Screen Synthesis

During the valid full-screen period, the image area can be overlaid before being output. If the image area to be output is smaller than a full-screen, the background color specified by the SC_RES_BK_COL_R, SC_RES_BK_COL_G, and SC_RES_BK_COL_B bits are displayed to fill the background.

Table 31.24 Screen Synthesis Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_OVR1	SC_RES_BK_COL_R [7:0]	128	Background Color Setting R/Cr Signal R: 8 bits; unsigned (0 to 255 [LSB]) Cr: 8 bits; 128 offset binary; unsigned (0 to 255 [LSB])
SC_SCL0_OVR1	SC_RES_BK_COL_B [7:0]	128	Background Color Setting B/Cb Signal B: 8 bits; unsigned (0 to 255 [LSB]) Cb: 8 bits; 128 offset binary; unsigned (0 to 255 [LSB])
SC_SCL0_OVR1	SC_RES_BK_COL_G [7:0]	0	Background Color Setting G/Y Signal G/Y: 8 bits; unsigned (0 to 255 [LSB])

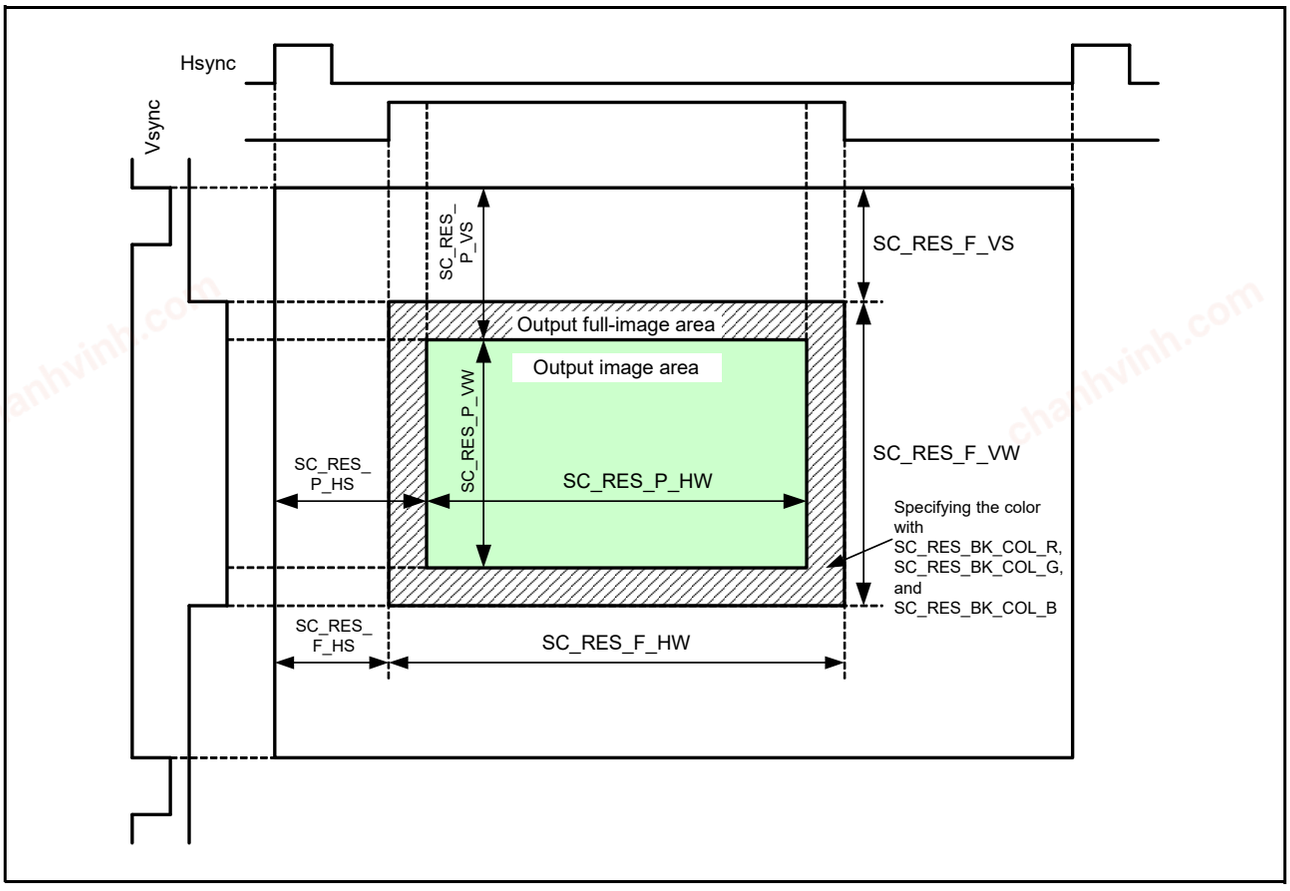


Figure 31.13 Area Relationship with Output Image Size Smaller than a Full Screen

chanhvinh.com

31.1.15 Selecting Format for Writing Video Image Signals to Frame Buffer

A format can be selected for writing video image signals to the frame buffer.

Although 24-bit YCbCr signals or 24-bit RGB signals are input to the scaling control block, they are converted into 16-bit YCbCr422 signals, 16-bit RGB565 signals, 32-bit YCbCr444 signals, or 32-bit RGB888 signals before being written to the frame buffer.

As bit reduction processing of RGB565, rounding off or 2 × 2 pattern dither can be selected with the SC_RES_DTH_ON bit. For details on pattern dither, see section 34.1.7, Dither Process in section 34., Video Display Controller 5 (7): Output Controller.

Table 31.25 Frame Buffer Writing Mode Setting

RES_BITDEC_ON	SC_RES_MD[1:0]	Writing Mode
0	3	YCbCr444 (normal, horizontal mirroring)
0	2	RGB888 (normal, horizontal mirroring)
1	1	RGB565 (normal, horizontal mirroring, rotation)
*	0	YCbCr422 (normal, horizontal mirroring, rotation)

Table 31.26 Video Signal Writing Format Selection Control

Register Name	Bit Name	Initial Value	Description
SC_SCL1_WR1	SC_RES_MD[1:0]	0	Frame Buffer Video-Signal Writing Format 0: YCbCr422 (16 bits) 1: RGB565 (16 bits) 2: RGB888 (24 (32) bits) 3: YCbCr444 (24 (32) bits)
SC_SCL1_WR6	SC_RES_BITDEC_ON	0	Bit Reduction On/Off 0: Off 1: On
SC_SCL1_WR6	SC_RES_DTH_ON	0	Dither Correction On/Off 0: Off (rounded off) 1: On (2 × 2 pattern dither)

chanhvinh.com

31.1.16 Horizontal Mirroring and Rotation

Horizontal mirroring and rotation can be carried out for scaled-down images before being written to the frame buffer.

Table 31.27 and Table 31.28 show the relationship between various writing modes for image processing and video signals.

Table 31.27 Relationship between Writing Modes and Video Signals

RES_DS_WR_MD[2:0]	Writing Modes	YCbCr444	YCbCr422	RGB565	RGB888
0	Normal writing	Enabled	Enabled	Enabled	Enabled
1	Horizontal mirroring	Enabled	Enabled	Enabled	Enabled
2	90° rotation	Disabled	Enabled	Enabled	Disabled
3	180° rotation	Disabled	Enabled	Enabled	Disabled
4	270° rotation	Disabled	Enabled	Enabled	Disabled
5 to 7	Setting prohibited	—	—	—	—

Table 31.28 Horizontal Mirroring and Rotation Control

Register Name	Bit Name	Initial Value	Description
SC_SCL1_WR1	SC_RES_DS_WR_MD [2:0]	0	Frame Buffer Writing Mode for Image Processing 0: Normal 1: Horizontal mirroring 2: 90° rotation 3: 180° rotation 4: 270° rotation 5 to 7: Setting prohibited

31.1.17 Writing to Frame Buffer

(1) Frame Buffer Transfer Mode

Either 32-byte or 128-byte transfer mode can be selected for accessing the frame buffer in which video image data and graphics data are stored.

Table 31.29 Frame Buffer Transfer Mode

Register Name	Bit Name	Initial Value	Description
SC_SCL1_WR1	SC_RES_BST_MD	0	Transfer Burst Length for Frame Buffer Writing 0: 32-byte 1: 128-byte

(2) Frame Buffer Write Control

Frame buffer writing is enabled or disabled.

Table 31.30 Frame Buffer Writing Control

Register Name	Bit Name	Initial Value	Description
SC_SCL1_WR5	SC_RES_WENB	0	Frame Buffer Write Enable After making the setting to enable writing, writing starts from the second frame. 0: Frame buffer writing is disabled. 1: Frame buffer writing is enabled.

chanhvinh.com

(3) Frame Buffer Writing Rate Selection

A frame buffer writing rate can be selected from among 1/1, 1/2, 1/4, and 1/8 the vertical frequency of the input signal.

When 1/2, 1/4, or 1/8 is selected, either the top or bottom field can be selected for writing.

Table 31.31 Frame Buffer Write Control

Register Name	Bit Name	Initial Value	Description
SC_SCL1_WR5	SC_RES_FS_RATE [1:0]	0	Writing Rate Sets the frame buffer writing rate to the vertical frequency of the input signal. 0: 1/1 an input signal (The SC_RES_FLD_SEL setting is invalid.) 1: 1/2 an input signal 2: 1/4 an input signal 3: 1/8 an input signal
SC_SCL1_WR5	SC_RES_FLD_SEL	0	Write Field Select 0: Top field 1: Bottom field
SC_SCL1_WR5	SC_RES_INTER	1	Field Operating Mode Select 0: Progressive 1: Interlace

(4) Frame Buffer Write Addresses

Frame buffer addresses are specified using the base address, line offset address, frame offset address, data size of a line, and the number of lines in a frame. When an interlaced video image is input, the top and bottom field data can be separately stored in the frame buffer.

The SC_RES_BASE[31:0], SC_RES_LN_OFF[14:0], and SC_RES_FLM_OFF[22:0] bits should be set in 32-byte units (the lower five bits should be fixed to 0).

For 128-byte transfer, bits [6:5] in the address control registers should be fixed to 0 since addresses should be specified in 128-byte units.

For the data size of a line and the number of lines in a frame, the relevant register values set for the scaling-down control block are used.

chanhvinh.com

Table 31.32 Frame Buffer Write Address Control

Register Name	Bit Name	Initial Value	Description
SC_SCL1_WR1	SC_RES_TB_ADD_MOD	0	Top and Bottom Data Write Address Specification Method 0: A write address is specified in common for top and bottom data. 1: Separate write addresses are specified for top and bottom data.
SC_SCL1_WR2	SC_RES_BASE [31:0]	0	Frame Buffer Base Address Sets the start address of the frame buffer to store the frame data for the top field when SC_RES_TB_ADD_MOD = 1 or that for the top and bottom fields when SC_RES_TB_ADD_MOD = 0. For 32-byte transfer: The lower five bits should be fixed to 0_0000. For 128-byte transfer: The lower seven bits should be fixed to 000_0000.
SC_SCL1_WR8	SC_RES_BASE_B [31:0]	0	Frame Buffer Base Address for Bottom Sets the start address of the frame buffer to store the frame data for the bottom field when SC_RES_TB_ADD_MOD = 1. For 32-byte transfer: The lower five bits should be fixed to 0_0000. For 128-byte transfer: The lower seven bits should be fixed to 000_0000.
SC_SCL1_WR3	SC_RES_LN_OFF [14:0]	2048	Frame Buffer Line Offset Address Sets the line offset address for calculating the line start address for the top field when SC_RES_TB_ADD_MOD = 1 or that for the top and bottom fields when SC_RES_TB_ADD_MOD = 0. Line 0: SC_RES_BASE Line 1: SC_RES_BASE + SC_RES_LN_OFF × 1 : Line n: SC_RES_BASE + SC_RES_LN_OFF × n For 32-byte transfer: The lower five bits should be fixed to 0_0000. For 128-byte transfer: The lower seven bits should be fixed to 000_0000.
SC_SCL1_WR9	SC_RES_LN_OFF_B [14:0]	2048	Frame Buffer Line Offset Address for Bottom Sets the line offset address for calculating the line start address for the bottom field when SC_RES_TB_ADD_MOD = 1. Line 0: SC_RES_BASE_B Line 1: SC_RES_BASE_B + SC_RES_LN_OFF_B × 1 : Line n: SC_RES_BASE_B+ SC_RES_LN_OFF_B × n For 32-byte transfer: The lower five bits should be fixed to 0_0000. For 128-byte transfer: The lower seven bits should be fixed to 000_0000.
SC_SCL1_WR4	SC_RES_FLM_OFF [22:0]	524288	Frame Buffer Frame Offset Address Sets the frame offset address for calculating the start address of each frame for the top field when SC_RES_TB_ADD_MOD = 1 or that for the top and bottom fields when SC_RES_TB_ADD_MOD = 0. Buffer 0: SC_RES_BASE Buffer 1: SC_RES_BASE + SC_RES_FLM_OFF × 1 : Buffer n: SC_RES_BASE + SC_RES_FLM_OFF × n For 32-byte transfer: The lower five bits should be fixed to 0_0000. For 128-byte transfer: The lower seven bits should be fixed to 000_0000.

chanhvinh.com

Table 31.32 Frame Buffer Write Address Control

Register Name	Bit Name	Initial Value	Description
SC_SCL1_WR10	SC_RES_FLM_OFF_B [22:0]	524288	Frame Buffer Frame Offset Address for Bottom Sets the frame offset address for calculating the start address of each frame for the bottom field when SC_RES_TB_ADD_MOD = 1. Buffer 0: SC_RES_BASE_B Buffer 1: SC_RES_BASE_B + SC_RES_FLM_OFF_B × 1 : Buffer n: SC_RES_BASE_B + SC_RES_FLM_OFF_B × n For 32-byte transfer: The lower five bits should be fixed to 0_0000. For 128-byte transfer: The lower seven bits should be fixed to 000_0000.

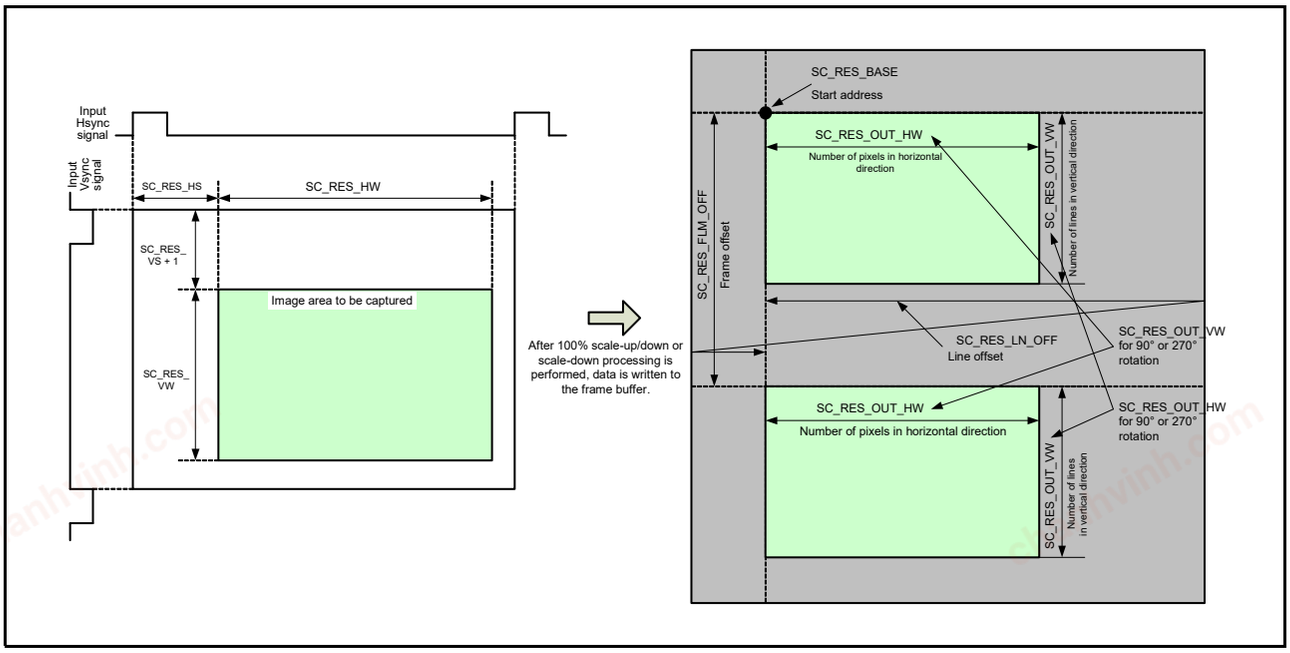


Figure 31.14 Data Arrangement in Frame Buffer

chanhvinh.com

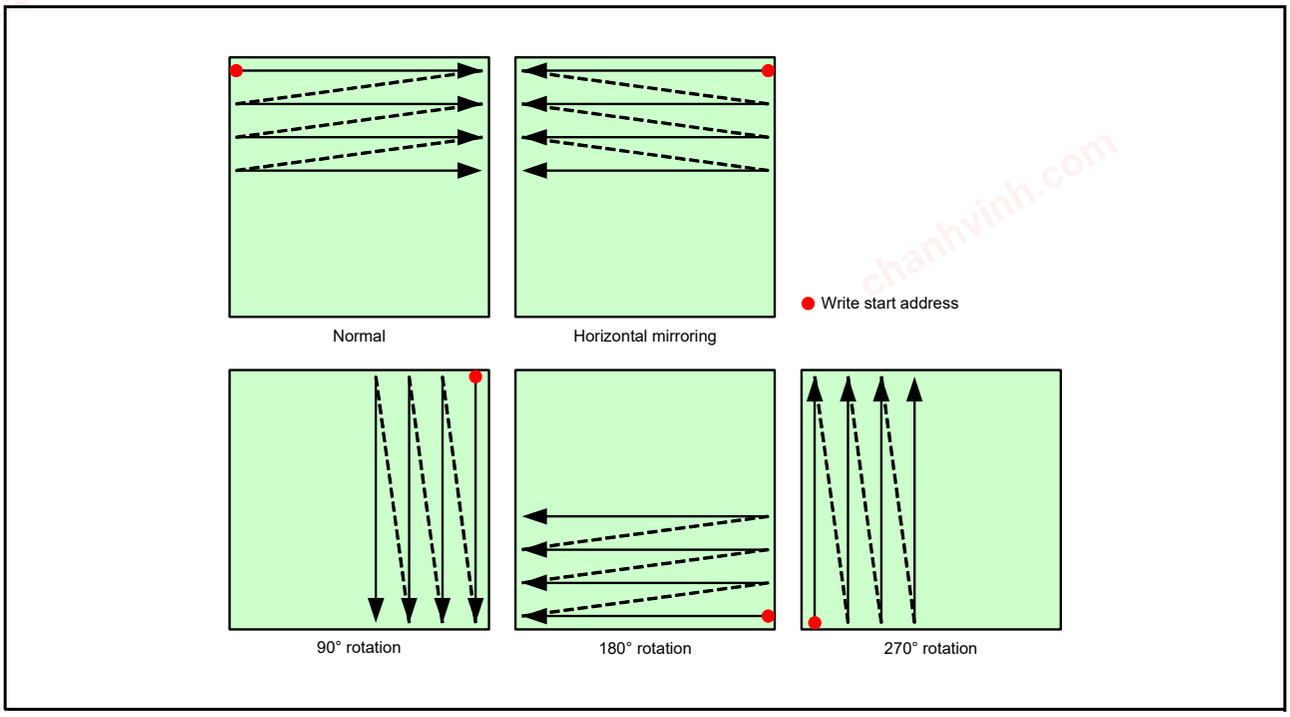


Figure 31.15 Data Arrangement in Frame Buffer in Various Writing Modes

(5) Frame Buffer Management

The scaling control block can handle multiple frames as the frame buffer.

Data is written to the buffer in cyclic mode according to the number of frames specified by the SC_RES_FLM_NUM bits.

For rotation, the SC_RES_FLM_NUM bits should be set to two or more frames.

To use the frame buffer as the ring buffer in line mode, the SC_RES_FLM_NUM bits should be set to 0 (1 frame) and the SC_RES_LOOP bit to 1.

chanhvinh.com

Table 31.33 Frame Buffer Write Control

Register Name	Bit Name	Initial Value	Description
SC_SCL1_WR3	SC_RES_FLM_NUM[9:0]	1	Number of Frames of Buffer to be Written to Sets the number of frames for the top field when SC_RES_TB_ADD_MOD = 1 or that for the top and bottom fields when SC_RES_TB_ADD_MOD = 0 Number of frames defined by SC_RES_FLM_NUM + 1 are used.
SC_SCL1_WR9	SC_RES_FLM_NUM_B[9:0]	1	Number of Frames of Buffer to be Written to for Bottom Field when SC_RES_TB_ADD_MOD = 1 Number of frames defined by SC_RES_FLM_NUM_B + 1 are used.
SC_SCL1_WR1	SC_RES_LOOP	0	Frame Buffer Write Mode Select 0: Frame mode 1: Line mode (read as ring buffer)
SC_SCL1_WR7	SC_RES_FLM_CNT[9:0]	—	Frame Number of Frame Being Accessed Frame number of the frame being accessed in the top field when SC_RES_TB_ADD_MOD = 1 or that in the top or bottom field when SC_RES_TB_ADD_MOD = 0.
SC_SCL1_WR11	SC_RES_FLM_CNT_B[9:0]	—	Frame Number of Frame Being Accessed in Bottom Field Frame number of the frame being accessed in the bottom field when SC_RES_TB_ADD_MOD = 1.

(6) Buffer Overflow Handling

If writing to the frame buffer cannot be completed due to bus-traffic related problems, an overflow interrupt can be output to the interrupt controller.

Table 31.34 Buffer Overflow Detection

Register Name	Bit Name	Initial Value	Description
SC_SCL1_WR7	SC_RES_OVERFLOW	—	Line Buffer Overflow Detect 1: Line buffer has overflowed. 0: Line buffer has not overflowed.

(7) Frame Buffer Write End Flag

When writing one frame of data to the frame buffer is completed, a frame buffer write end interrupt can be output to the interrupt controller.

chanhvinh.com

31.1.18 Selecting a Scaling-up Process or Graphics 0 Process

Scaling-up process and graphics 0 process are mutually exclusive and thus frame buffer cannot be read out simultaneously for the processes.

When displaying input video image signals or displaying enlarged graphics, data is read from the frame buffer via the scaling-up control block.

However, graphics can be enlarged and displayed by the scaling-up control block only when the RGB565, RGB888, YCbCr422, or YCbCr444 format is used.

When displaying graphics without enlargement, the data is read from the frame buffer via the graphics 0 processing block.

With the SC_RES_IBUS_SYNC_SEL bit, sync signals for reading out the frame buffer and read size setting bits are selected.

Table 31.35 Selection of Scaling-Up Process and Graphics 0 Process

Type of Output Scaling Display	SC_RES_ IBUS_SYNC_SEL	Sync Signals for Frame Buffer Read	Frame Buffer Read Size Setting Bits	Display Enabling Bits
Input video signal display Enlarged graphics display	0	Output from scaling-up control block	SC_RES_IN_VW SC_RES_IN_HW	SC_RES_P_VS SC_RES_P_VW SC_RES_P_HS SC_RES_P_HW
Graphics display	1	Output from graphics 0 processing block	GR_FLM_LNUM* GR_HW*	GR_GRC_VS GR_GRC_VW GR_GRC_HS GR_GRC_HW

Note: * The value set to the register + 1 is the actual read size.

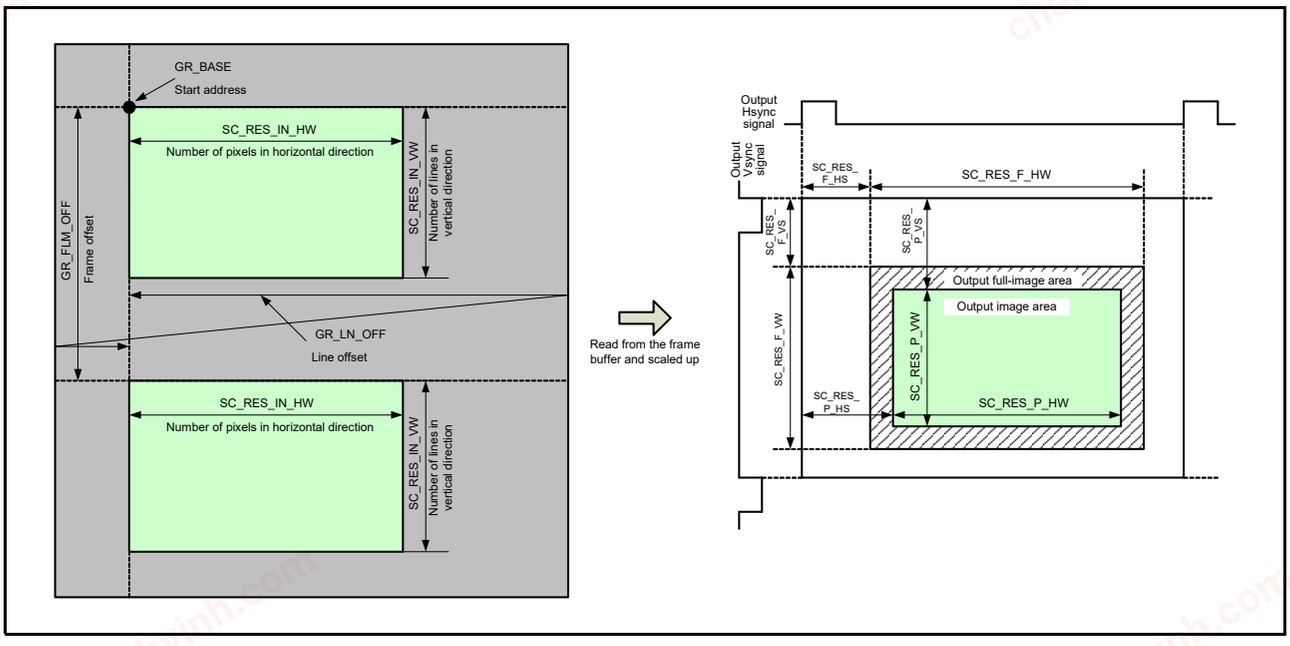


Figure 31.16 Area Setting for Input Video Image Signal Display and Enlarged Graphics Display

chanhvinh.com

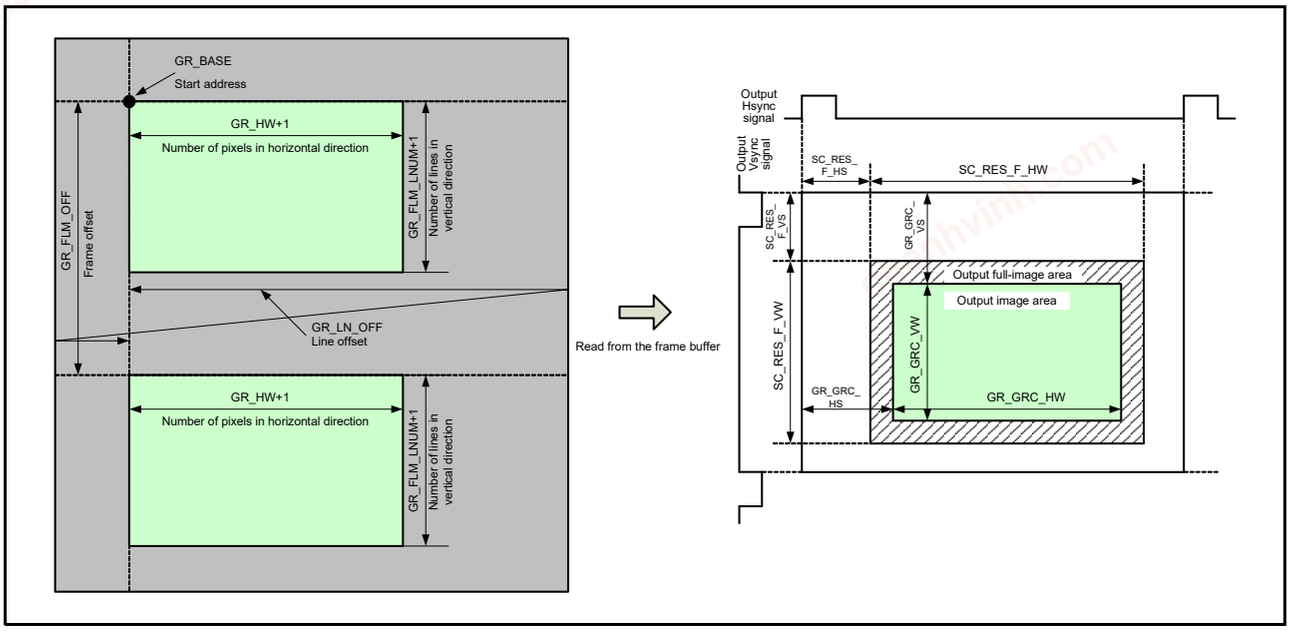


Figure 31.17 Area Setting for Graphics Display

Table 31.36 Scaling-Up Process or Graphics 0 Process Selection

Register Name	Bit Name	Initial Value	Description
SC_SCL0_US8	SC_RES_IBUS_SYNC_SEL	0	Sync Signal Select for Frame Buffer Read Block 0: Sync signals from the scaling-up control block 1: Sync signals from the graphics processing block

The GR_DISP_SEL bits are used to select a display by the scaling-up control block (video image display or enlarged graphics display) or graphics display.

For details on the graphics processing, refer to the section 33., Video Display Controller 5 (5): Image Synthesizer.

31.1.19 Selecting Field for Frame Buffer Reading

For the next frame buffer to be read, the top or bottom field can be selected. This field selection is used in the scaling-up control block.

Table 31.37 Field Specification for Frame Buffer Reading

Register Name	Bit Name	Initial Value	Description
GR_FLM1	GR_FLD_SEL	0	Enables or disables top or bottom field selection for the next frame buffer to be read. 0: Field selection is disabled. 1: Top or bottom field can be selected.
GR_FLM3	GR_FLD_NXT	0	Selects the top or bottom field for the next frame buffer. 0: Bottom 1: Top

chanhvinh.com

31.1.20 Pointer Buffer and Frame Buffer Reading Processing

(1) Pointer Buffers

The pointer buffers can be used to control the frame buffer for the input video image. They are mainly used to prevent flicker in the output video image, which occurs when the input and output vertical sync signals are asynchronous.

Four pointer buffers are provided and each pointer buffer has a start address register that shows the start location of the frame buffer and a field information register that shows the current field is the top or bottom field. The four pointer buffers are arranged in a ring structure and a write pointer is provided to indicate the pointer buffer corresponding to the location currently being written to. The location pointed to by the write pointer is being written to, and the corresponding pointer buffer value is undetermined. The value in the pointer buffer corresponding to the location being written to and the value in the write pointer are automatically updated when frame data writing is completed.

When the frame buffer address setting signal is linked with the pointer buffer (GR_FLM_SEL = 3), frame buffer reading can be controlled by using the read pointer that indicates the pointer buffer corresponding to the location being read; the start address and field information of the next frame buffer to be read should be read from the pointer buffer and they should be set in the frame buffer base address and field information. The read pointer value is automatically updated at the rising edge of the vertical sync signal on the reading side.

(2) Write Pointer Control

The write pointer is incremented by one every time frame data writing is completed.

(3) Read Pointer Control

The read pointer is updated at the rising edge of the vertical sync signal on the reading side according to the difference between the read and write pointer values as follows.

(A) When (write pointer value) – (read pointer value) ≤ 1

The read pointer value is not updated (the same frame is displayed continuously).

(B) When (write pointer value) – (read pointer value) = 2

The read pointer is incremented by one with the next updating timing.

(C) When (write pointer value) – (read pointer value) ≥ 3

The read pointer is incremented by two with the next updating timing.

(One frame is skipped.)

(4) Frame Buffer Read Control

(A) When SC_RES_WENB = 0

Frame data is not written to the frame buffer, and the frame buffer is not read.

(B) When frame data writing is terminated with SC_RES_WENB = 1

As the pointer buffer value is determined, the frame buffer is read.

chanhvinh.com

Table 31.38 Frame Buffer Control

Register Name	Bit Name	Initial Value	Description
SC_SCL1_PBUF0	SC_BUF0_ADD	0	Start address of the write buffer pointed to by pointer buffer 0
SC_SCL1_PBUF1	SC_BUF1_ADD	0	Start address of the write buffer pointed to by pointer buffer 1
SC_SCL1_PBUF2	SC_BUF2_ADD	0	Start address of the write buffer pointed to by pointer buffer 2
SC_SCL1_PBUF3	SC_BUF3_ADD	0	Start address of the write buffer pointed to by pointer buffer 3
SC_SCL1_PBUF_FLD	SC_FLD_INF0	0	Top or bottom field information pointed to by pointer buffer 0 0: Bottom 1: Top
SC_SCL1_PBUF_FLD	SC_FLD_INF1	0	Top or bottom field information pointed to by pointer buffer 1 0: Bottom 1: Top
SC_SCL1_PBUF_FLD	SC_FLD_INF2	0	Top or bottom field information pointed to by pointer buffer 2 0: Bottom 1: Top
SC_SCL1_PBUF_FLD	SC_FLD_INF3	0	Top or bottom field information pointed to by pointer buffer 3 0: Bottom 1: Top
SC_SCL1_PBUF_CNT	SC_PBUF_RST	0	Reset Control for the Pointer Buffer 0: Pointer buffer is not reset. 1: Pointer buffer is reset.
SC_SCL1_MON1	SC_PBUF_NUM	0	Write pointer indicating the pointer buffer number corresponding to the location currently being written to.
SC_SCL1_WR5	SC_RES_WENB	0	Frame Buffer Write Enable After making the setting to enable writing, writing starts from the second frame. 0: Writing is disabled. 1: Writing is enabled.
GR_FLM1	GR_FLM_SEL	0	Frame Buffer Address Setting Signal Selection 0: Links to scaling-down process. (This setting is prohibited when separate write addresses are specified for the top and bottom fields; that is, SC_RES_TB_ADD_MOD = 1 in SC_SCL1_WR1.) 1: Selects GR0_FLM_NUM. 2: Setting prohibited 3: Links to pointer buffer.
GR_FLM2	GR_BASE	0	Frame Buffer Base Address Sets the start address of the frame buffer where frame data is to be stored. GR_BASE[4:3] and GR_BASE[6:3] are referred to during 32-byte burst transfer and 128-byte burst transfer, respectively, to skip the start line data. The lower three bits should be fixed to 000.

For other frame buffer read operation and graphics processing, refer to section 33., Video Display Controller 5 (5): Image Synthesizer.

chanhvinh.com

31.2 Register Descriptions

Table 31.39 shows the register configuration.

- Symbols used in Register Description:

Initial value: Register value after a reset
—: Undefined value
R/W: Readable/writable. The written value can be read.
R/WC0: Readable/writable. Writing 0 initializes the bit. Writing 1 is ignored.
R/WC1: Readable/writable. Writing 1 initializes the bit. Writing 0 is ignored.
R: Read-only. The write value should always be 0.
—/W: Write-only. The read value is undefined.

Table 31.39 Register Configuration of the Scaler

Name	Abbreviation	R/W	Initial Value	Address	Access Size
SCL0 register update control register (SC0)	SC0_SCL0_UPDATE	R/WC1	H'0000 0000	H'FCFF 7500	32
Mask control register (SC0)	SC0_SCL0_FRC1	R/W	H'0AF0 0001	H'FCFF 7504	32
Missing Vsync compensation control register (SC0)	SC0_SCL0_FRC2	R/W	H'0E10 0001	H'FCFF 7508	32
Output sync select register (SC0)	SC0_SCL0_FRC3	R/W	H'0000 0001	H'FCFF 750C	32
Free-running period control register (SC0)	SC0_SCL0_FRC4	R/W	H'020C 031F	H'FCFF 7510	32
Output delay control register (SC0)	SC0_SCL0_FRC5	R/W	H'0000 0101	H'FCFF 7514	32
Full-screen vertical size register (SC0)	SC0_SCL0_FRC6	R/W	H'0023 01E0	H'FCFF 7518	32
Full-screen horizontal size register (SC0)	SC0_SCL0_FRC7	R/W	H'0090 0280	H'FCFF 751C	32
Vsync detection register (SC0)	SC0_SCL0_FRC9	R	H'0000 0000	H'FCFF 7524	32
Status monitor 0 register (SC0)	SC0_SCL0_MON0	R	H'0000	H'FCFF 7528	16
Interrupt control register (SC0)	SC0_SCL0_INT	R/W	H'0000	H'FCFF 752A	16
Scaling-down control register (SC0)	SC0_SCL0_DS1	R/W	H'0000 0011	H'FCFF 752C	32
Vertical capture size register (SC0)	SC0_SCL0_DS2	R/W	H'0012 00F0	H'FCFF 7530	32
Horizontal capture size register (SC0)	SC0_SCL0_DS3	R/W	H'00F4 05A0	H'FCFF 7534	32
Horizontal scale down register (SC0)	SC0_SCL0_DS4	R/W	H'1000 2408	H'FCFF 7538	32
Initial vertical phase register (SC0)	SC0_SCL0_DS5	R/W	H'1800 0000	H'FCFF 753C	32
Vertical scaling register (SC0)	SC0_SCL0_DS6	R/W	H'0000 07FC	H'FCFF 7540	32
Scaling-down control block output size register (SC0)	SC0_SCL0_DS7	R/W	H'00F0 0280	H'FCFF 7544	32
Scaling-up control register (SC0)	SC0_SCL0_US1	R/W	H'0000 0011	H'FCFF 7548	32
Output image vertical size register (SC0)	SC0_SCL0_US2	R/W	H'0023 01E0	H'FCFF 754C	32
Output image horizontal size register (SC0)	SC0_SCL0_US3	R/W	H'0090 0280	H'FCFF 7550	32
Scaling-up control block input size register (SC0)	SC0_SCL0_US4	R/W	H'00F0 0280	H'FCFF 7554	32
Horizontal scale up register (SC0)	SC0_SCL0_US5	R/W	H'0000 2408	H'FCFF 7558	32
Horizontal scale up initial phase register (SC0)	SC0_SCL0_US6	R/W	H'1000 0000	H'FCFF 755C	32
Trimming register (SC0)	SC0_SCL0_US7	R/W	H'0000 0000	H'FCFF 7560	32
Frame buffer read select register (SC0)	SC0_SCL0_US8	R/W	H'0000 0000	H'FCFF 7564	32
Background color register (SC0)	SC0_SCL0_OVR1	R/W	H'0080 0080	H'FCFF 756C	32

chanhvinh.com

Table 31.39 Register Configuration of the Scaler

Name	Abbreviation	R/W	Initial Value	Address	Access Size
SCL1 register update control register (SC0)	SC0_SCL1_UPDATE	R/WC1	H'0000 0000	H'FCFF 7580	32
Writing mode register (SC0)	SC0_SCL1_WR1	R/W	H'0000 0000	H'FCFF 7588	32
Write address register 1T (SC0)	SC0_SCL1_WR2	R/W	H'0000 0000	H'FCFF 758C	32
Write address register 2T (SC0)	SC0_SCL1_WR3	R/W	H'0800 0001	H'FCFF 7590	32
Write address register 3T (SC0)	SC0_SCL1_WR4	R/W	H'0008 0000	H'FCFF 7594	32
Frame sub-sampling register (SC0)	SC0_SCL1_WR5	R/W	H'0000 1000	H'FCFF 759C	32
Bit reduction register (SC0)	SC0_SCL1_WR6	R/W	H'0000 0000	H'FCFF 75A0	32
Write detection register (SC0)	SC0_SCL1_WR7	R	H'0000 0000	H'FCFF 75A4	32
Write address register 1B (SC0)	SC0_SCL1_WR8	R/W	H'0000 0000	H'FCFF 75A8	32
Write address register 2B (SC0)	SC0_SCL1_WR9	R/W	H'0800 0001	H'FCFF 75AC	32
Write address register 3B (SC0)	SC0_SCL1_WR10	R/W	H'0008 0000	H'FCFF 75B0	32
Write detection register B (SC0)	SC0_SCL1_WR11	R	H'0000 0000	H'FCFF 75B4	32
Status monitor 1 register (SC0)	SC0_SCL1_MON1	R	H'0000 0000	H'FCFF 75B8	32
Pointer buffer 0 register (SC0)	SC0_SCL1_PBUF0	R	H'0000 0000	H'FCFF 75BC	32
Pointer buffer 1 register (SC0)	SC0_SCL1_PBUF1	R	H'0000 0000	H'FCFF 75C0	32
Pointer buffer 2 register (SC0)	SC0_SCL1_PBUF2	R	H'0000 0000	H'FCFF 75C4	32
Pointer buffer 3 register (SC0)	SC0_SCL1_PBUF3	R	H'0000 0000	H'FCFF 75C8	32
Pointer buffer and field information register (SC0)	SC0_SCL1_PBUF_FLD	R	H'0000 0000	H'FCFF 75CC	32
Pointer buffer control register (SC0)	SC0_SCL1_PBUF_CNT	R/W	H'0000 0000	H'FCFF 75D0	32
Graphics 0 register update control register	GR0_UPDATE	R/WC1	H'0000 0000	H'FCFF 7600	32
Frame buffer read control register (graphics 0)	GR0_FLM_RD	R/W	H'0000 0000	H'FCFF 7604	32
Frame buffer control register 1 (graphics 0)	GR0_FLM1	R/W	H'0000 0000	H'FCFF 7608	32
Frame buffer control register 2 (graphics 0)	GR0_FLM2	R/W	H'0000 0000	H'FCFF 760C	32
Frame buffer control register 3 (graphics 0)	GR0_FLM3	R/W	H'0800 0001	H'FCFF 7610	32
Frame buffer control register 4 (graphics 0)	GR0_FLM4	R/W	H'0008 0000	H'FCFF 7614	32
Frame buffer control register 5 (graphics 0)	GR0_FLM5	R/W	H'0000 03FF	H'FCFF 7618	32
Frame buffer control register 6 (graphics 0)	GR0_FLM6	R/W	H'8000 0000	H'FCFF 761C	32
Alpha blending control register 1 (graphics 0)	GR0_AB1	R/W	H'0000 0000	H'FCFF 7620	32
Alpha blending control register 2 (graphics 0)	GR0_AB2	R/W	H'0000 0000	H'FCFF 7624	32
Alpha blending control register 3 (graphics 0)	GR0_AB3	R/W	H'0000 0000	H'FCFF 7628	32
Alpha blending control register 7 (graphics 0)	GR0_AB7	R/W	H'00FF 0000	H'FCFF 7638	32
Alpha blending control register 8 (graphics 0)	GR0_AB8	R/W	H'0000 0000	H'FCFF 763C	32
Alpha blending control register 9 (graphics 0)	GR0_AB9	R/W	H'0000 0000	H'FCFF 7640	32

chanhvinh.com

Table 31.39 Register Configuration of the Scaler

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Alpha blending control register 10 (graphics 0)	GR0_AB10	R/W	H'0000 0000	H'FCFF 7644	32
Alpha blending control register 11 (graphics 0)	GR0_AB11	R/W	H'0000 0000	H'FCFF 7648	32
Background color control register (graphics 0)	GR0_BASE	R/W	H'0000 8080	H'FCFF 764C	32
CLUT table control register (graphics 0)	GR0_CLUT	R/W	H'0000 0000	H'FCFF 7650	32

31.2.1 SCL0 Register Update Control Register (SC0_SCL0_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SC0_SCL0_VEN_D	SC0_SCL0_VEN_C	—	—	—	SC0_SCL0_UPDATE	—	—	—	SC0_SCL0_VEN_B	—	—	—	SC0_SCL0_VEN_A
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/WC1	R/WC1	R	R	R	R/WC1	R	R	R	R/WC1	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	SC0_SCL0_VEN_D	0	R/WC1	Scaling-Up Control and Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
12	SC0_SCL0_VEN_C	0	R/WC1	Scaling-Down Control and Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	SC0_SCL0_UPDATE	0	R/WC1	SYNC Control Register Update 0: Registers are not updated. 1: Registers are updated.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SC0_SCL0_VEN_B	0	R/WC1	Synchronization Control and Scaling-up Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC0_SCL0_VEN_A	0	R/WC1	Scaling-Down Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

chanhvinh.com

31.2.2 Mask Control Register (SC0_SCL0_FRC1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SC0_RES_VMASK[15:0]															
Initial value:	0	0	0	0	1	0	1	0	1	1	1	1	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SC0_RES_VMASK_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SC0_RES_VMASK [15:0]	2800	R/W	Repeated Vsync Signal Masking Period Sets the repeated Vsync signal masking period beginning at a Vsync signal in terms of 128 pixel-clock periods. Masking period [usec] = SC0_RES_VMASK × 128 ÷ pixel clock frequency [MHz]
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC0_RES_ VMASK_ON	1	R/W	Repeated Vsync Signal Masking Control 0: Repeated Vsync signal masking control is disabled. 1: Repeated Vsync signal masking control is enabled.

Note: This register is updated when the SC0_SCL0_UPDATE bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

31.2.3 Missing Vsync Compensation Control Register (SC0_SCL0_FRC2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SC0_RES_VLACK[15:0]															
Initial value:	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SC0_RES_VLACK_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SC0_RES_ VLACK [15:0]	3600	R/W	Missing-Sync Compensating Pulse Output Wait Time Sets the wait time before outputting a missing-sync compensating pulse after a Vsync signal. Wait time [usec] = SC0_RES_VLACK × 128 ÷ pixel clock frequency [MHz]
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC0_RES_ VLACK_ON	1	R/W	Missing Vsync Signal Compensation 0: Compensation of missing Vsync signals is disabled. 1: Compensation of missing Vsync signals is enabled.

Note: This register is updated when the SC0_SCL0_UPDATE bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

chanhvinh.com

31.2.4 Output Sync Select Register (SC0_SCL0_FRC3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SC0_RES_VS_SEL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC0_RES_VS_SEL	1	R/W	Vsync Signal Output Select 0: Externally input Vsync signal 1: Internally generated free-running Vsync signal

Note: This register is updated when the SC0_SCL0_UPDATE bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

chanhvinh.com

31.2.5 Free-Running Period Control Register (SC0_SCL0_FRC4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SC0_RES_FV[10:0]										
Initial value:	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC0_RES_FH[10:0]										
Initial value:	0	0	0	0	0	0	1	1	0	0	0	1	1	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	SC0_RES_FV [10:0]	524	R/W	Free-Running Vsync Period Setting Free-running Vsync period = (SC0_RES_FV + 1) × horizontal period [usec]
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC0_RES_FH [10:0]	799	R/W	Hsync Period Setting Hsync period [usec] = (SC0_RES_FH +1) ÷ pixel clock frequency [MHz]

Note: This register is updated when the SC0_SCL0_UPDATE bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

31.2.6 Output Delay Control Register (SC0_SCL0_FRC5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SC0_RES_FLD_DLY_SEL	SC0_RES_VSDLY[7:0]							
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	SC0_RES_ FLD_DLY_SEL	1	R/W	Field Determination Signal Delay Control 0: No delay 1: Delay of one vertical cycle
7 to 0	SC0_RES_ VSDLY[7:0]	1	R/W	Vsync Signal Delay Control Adjusts the Vsync signal delay in the output Hsync period units. Vsync signal delay [usec]: SC0_RES_VSDLY × output Hsync period [usec]

Note: This register is updated when the SC0_SCL0_VEN_B bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

chanhvinh.com

31.2.7 Full-Screen Vertical Size Register (SC0_SCL0_FRC6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SC0_RES_F_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC0_RES_F_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	SC0_RES_F_VS[10:0]	35	R/W	Vertical Enable Signal Start Position for Full Screen. (VSYNC + V backporch lines) Note: The set value should be four or more (lines). SC0_RES_F_VS + SC0_RES_F_VW should be equal to or less than 2039 (lines).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC0_RES_F_VW[10:0]	480	R/W	Vertical Enable Signal Width for Full Screen (lines) Note: SC0_RES_F_VS + SC0_RES_F_VW should be equal to or less than 2039 (lines).

Note: This register is updated when the SC0_SCL0_VEN_B bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

chanhvinh.com

31.2.8 Full-Screen Horizontal Size Register (SC0_SCL0_FRC7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SC0_RES_F_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC0_RES_F_HW[10:0]										
Initial value:	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	SC0_RES_F_HS[10:0]	144	R/W	Horizontal Enable Signal Start Position for Full Screen. (HSYNC+H backporch pixel-clock cycles) Note: The set value should be 16 or more (clock cycles). SC0_RES_F_HS + SC0_RES_F_HW should be equal to or less than 2015 (clock cycles).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC0_RES_F_HW[10:0]	640	R/W	Horizontal Enable Signal Width for Full Screen (pixel-clock cycles) Note 1. SC0_RES_F_HS + SC0_RES_F_HW should be equal to or less than 2015 (clock cycles). Note 2. The set value should be equal to (horizontal signal width for full screen + 2) when serial RGB output is selected as an LCD output signal.

Note: This register is updated when the SC0_SCL0_VEN_B bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

chanhvinh.com

31.2.9 Vsync Detection Register (SC0_SCL0_FRC9)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SC0_RES_QVLOCK	—	—	—	SC0_RES_QVLACK
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SC0_RES_QVLOCK	0	R	Locked Vsync Signal Detection Flag 1: No repeated or missing Vsync signal input has been detected for four or more vertical periods. 0: Repeated or missing Vsync signal input has been detected.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC0_RES_QVLACK	0	R	Missing Vsync Signal Detection Flag 1: Missing Vsync signal input has been detected. 0: No missing Vsync signal input has been detected.

31.2.10 Status Monitor 0 Register (SC0_SCL0_MON0)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC0_RES_LIN_STAT[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC0_RES_LIN_STAT[10:0]	All 0	R	Current location of the image line input to the scaling-down control block.

chanhvinh.com

31.2.11 Interrupt Control Register (SC0_SCL0_INT)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC0_RES_LINE[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC0_RES_LINE[10:0]	All 0	R/W	Setting of Interrupt on Image Line Input to Scaling-down Control Block When the location of the image line input to the scaling-down control block matches the SC0_RES_LINE setting, an interrupt signal is output. (Setting prohibited in this product)

Note: This register is updated when the SC0_SCL0_VEN_A bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

31.2.12 Scaling-Down Control Register (SC0_SCL0_DS1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SC0_RES_DS_V_ON	—	—	—	SC0_RES_DS_H_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SC0_RES_DS_V_ON	1	R/W	Vertical Scale Down On/Off 0: Off 1: On
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC0_RES_DS_H_ON	1	R/W	Horizontal Scale Down On/Off 0: Off 1: On

Note: This register is updated when the SC0_SCL0_VEN_A bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

chanhvinh.com

31.2.13 Vertical Capture Size Register (SC0_SCL0_DS2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SC0_RES_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC0_RES_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	SC0_RES_VS [10:0]	18	R/W	Vertical Position Setting for Video Signal Capturing (VSYNC + (V backporch - 1) lines) Note: The set value should be four or more (lines). SC0_RES_VS + SC0_RES_VW should be equal to or less than 2039 (lines).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC0_RES_VW [10:0]	240	R/W	Vertical Width of Video Signal to be Captured (Lines) Note: SC0_RES_VS + SC0_RES_VW should be equal to or less than 2039 (lines).

Note: This register is updated when the SC0_SCL0_VEN_A bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

chanhvinh.com

31.2.14 Horizontal Capture Size Register (SC0_SCL0_DS3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SC0_RES_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	0	1	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC0_RES_HW[10:0]										
Initial value:	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	SC0_RES_HS [10:0]	244	R/W	Horizontal Position Setting for Video Signal Capturing (HSYNC + H backporch video-image clock cycles) Note: The set value should be 16 or more (clock cycles). SC0_RES_HS + SC0_RES_HW should be equal to or less than 2015 (clock cycles).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC0_RES_HW [10:0]	1440	R/W	Horizontal Width of Video Signal to be Captured (Video-image clock cycles) Note: SC0_RES_HS + SC0_RES_HW should be equal to or less than 2015 (clock cycles).

Note: This register is updated when the SC0_SCL0_VEN_A bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

chanhvinh.com

31.2.15 Horizontal Scale Down Register (SC0_SCL0_DS4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	SC0_RES_PFIL_SEL	SC0_RES_DS_H_INTERPOTYP	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC0_RES_DS_H_RATIO[15:0]															
Initial value:	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
29	SC0_RES_PFIL_SEL	0	R/W	Prefilter Mode Select for Brightness Signals 0: The prefilter is turned off. 1: The prefilter is turned on. (1/4 + 1/2 + 1/4)
28	SC0_RES_DS_H_INTERPOTYP	1	R/W	Horizontal Interpolation Mode Select 0: Hold interpolation 1: Linear interpolation
27 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	SC0_RES_DS_H_RATIO[15:0]	9224	R/W	Horizontal Scale Down Ratio ([15:12]: Integer part, [11:0]: Decimal part) round(SC0_RES_HW ÷ SC0_RES_OUT_HW × 4096) SC0_RES_DS_H_RATIO < 4096: Setting prohibited SC0_RES_DS_H_RATIO = 4096: 100% scale up SC0_RES_DS_H_RATIO > 4096: Scale down

Note: This register is updated when the SC0_SCL0_VEN_A bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

chanhvinh.com

31.2.16 Initial Vertical Phase Register (SC0_SCL0_DS5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	SC0_RES_V_INTERPOTYP	SC0_RES_TOP_INIPHASE[11:0]											
Initial value:	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	SC0_RES_BTM_INIPHASE[11:0]											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28	SC0_RES_V_INTERPOTYP	1	R/W	Vertical Interpolation Mode Select 0: Hold interpolation 1: Linear interpolation
27 to 16	SC0_RES_TOP_INIPHASE[11:0]	2048	R/W	Vertical Interpolation Start Phase for Top Field 0 to 4095 (0 to approx. 1.0)
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	SC0_RES_BTM_INIPHASE[11:0]	0	R/W	Vertical Interpolation Start Phase for Bottom Field 0 to 4095 (0 to approx. 1.0)

Note: This register is updated when the SC0_SCL0_VEN_A and SC0_SCL0_VEN_B bits in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) are 1.

chanhvinh.com

31.2.17 Vertical Scaling Register (SC0_SCL0_DS6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC0_RES_V_RATIO[15:0]															
Initial value:	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	SC0_RES_V_RATIO [15:0]	2044	R/W	Vertical Scale Up/Down Ratio ([15:12]: Integer part, [11:0]: Decimal part) For scale down: round (SC0_RES_VW ÷ SC0_RES_OUT_VW × 4096) For scale up: round (SC0_RES_IN_VW ÷ SC0_RES_P_VW × 4096) SC0_RES_V_RATIO < 4096: Scale up SC0_RES_V_RATIO = 4096: 100% scale up SC0_RES_V_RATIO > 4096: Scale down

Note: These bits updated when the SC0_SCL0_VEN_A and SC0_SCL0_VEN_B bits in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) are 1. Accordingly, even a scaled-up graphics display requires both an input Vsync signal and output Vsync signal.

chanhvinh.com

31.2.18 Scaling-Down Control Block Output Size Register (SC0_SCL0_DS7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SC0_RES_OUT_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC0_RES_OUT_HW[10:0]										
Initial value:	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	SC0_RES_OUT_VW [10:0]	240	R/W	Number of Valid Lines in Vertical Direction Output by Scaling-down Control Block (lines) This bit setting is used for the number of lines to be written to the frame buffer. When SC0_SCL1_WR1.SC0_RES_LOOP is 0 (frame write mode), specify the number of lines for one frame. When SC0_SCL1_WR1.SC0_RES_LOOP is 1 (line write mode), specify the number of lines for repeated write. Note: The SC0_RES_OUT_VW value should be aligned in 4-line units and equal to or smaller than the SC0_RES_VW value.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC0_RES_OUT_HW [10:0]	640	R/W	Number of Valid Horizontal Pixels Output by Scaling-Down Control Block (video-image clock cycles) Note: The SC0_RES_OUT_HW value should be aligned in 4-pixel units and equal to or smaller than the SC0_RES_HW value.

Note: This register is updated when the SC0_SCL0_VEN_A and SC0_SCL0_VEN_C bits in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) are 1.

chanhvinh.com

31.2.19 Scaling-Up Control Register (SC0_SCL0_US1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SC0_RES_US_V_ON	—	—	—	SC0_RES_US_H_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SC0_RES_US_V_ON	1	R/W	Vertical Scale Up On/Off 0: Off 1: On
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC0_RES_US_H_ON	1	R/W	Horizontal Scale Up On/Off 0: Off 1: On

Note: This register is updated when the SC0_SCL0_VEN_B bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

chanhvinh.com

31.2.20 Output Image Vertical Size Register (SC0_SCL0_US2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SC0_RES_P_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC0_RES_P_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	SC0_RES_P_VS[10:0]	35	R/W	Vertical Enable Signal Start Position for Output Image (VSYNC + V backporch lines) Note: The set value should be four or more (lines). SC0_RES_P_VS + SC0_RES_P_VW should be equal to or less than 2039 (lines).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC0_RES_P_VW[10:0]	480	R/W	Vertical Enable Signal Width for Output Image (lines) Note: SC0_RES_P_VS + SC0_RES_P_VW should be equal to or less than 2039 (lines).

Note: This register is updated when the SC0_SCL0_VEN_B bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

chanhvinh.com

31.2.21 Output Image Horizontal Size Register (SC0_SCL0_US3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SC0_RES_P_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC0_RES_P_HW[10:0]										
Initial value:	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	SC0_RES_P_HS[10:0]	144	R/W	Horizontal Enable Signal Start Position for Output Image (HSYNC+H backporch pixel-clock cycles) Note: The set value should be 16 or more (clock cycles). SC0_RES_P_HS + SC0_RES_P_HW should be equal to or less than 2015 (clock cycles).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC0_RES_P_HW[10:0]	640	R/W	Horizontal Enable Signal Width for Output Image (pixel-clock cycles) Note: SC0_RES_P_HS + SC0_RES_P_HW should be equal to or less than 2015 (clock cycles).

Note: This register is updated when the SC0_SCL0_VEN_B bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

chanhvinh.com

31.2.22 Scaling-Up Control Block Input Size Register (SC0_SCL0_US4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SC0_RES_IN_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC0_RES_IN_HW[10:0]										
Initial value:	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	SC0_RES_IN_VW[10:0]	240	R/W	Number of Valid Lines in Vertical Direction Input to Scaling-up Control Block (lines)
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC0_RES_IN_HW[10:0]	640	R/W	Number of Valid Horizontal Pixels Input to Scaling-up Control Block (pixel-clock cycles)

Note: This register is updated when the SC0_SCL0_VEN_B and SC0_SCL0_VEN_D bits in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) are 1.

31.2.23 Horizontal Scale Up Register (SC0_SCL0_US5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC0_RES_US_H_RATIO[15:0]															
Initial value:	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	SC0_RES_US_H_RATIO [15:0]	9224	R/W	Horizontal Scale Up Ratio ([15:12]: Integer part, [11:0]: Decimal part) round (SC0_RES_IN_HW ÷ SC0_RES_P_HW × 4096) SC0_RES_US_H_RATIO < 4096: Scale up SC0_RES_US_H_RATIO = 4096: 100% scale up SC0_RES_US_H_RATIO > 4096: Setting prohibited

Note: This register is updated when the SC0_SCL0_VEN_B bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

chanhvinh.com

31.2.24 Horizontal Scale Up Initial Phase Register (SC0_SCL0_US6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	SC0_RES_US_H_INTERPOTYP	SC0_RES_US_HT_INIPHASE[11:0]											
Initial value:	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	SC0_RES_US_HB_INIPHASE[11:0]											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28	SC0_RES_US_H_INTERPOTYP	1	R/W	Horizontal Interpolation Mode Select 0: Hold interpolation 1: Linear interpolation
27 to 16	SC0_RES_US_HT_INIPHASE[11:0]	0	R/W	Horizontal Interpolation Start Phase for Top Field 0 to 4095 (0 to approx. 1.0)
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	SC0_RES_US_HB_INIPHASE[11:0]	0	R/W	Horizontal Interpolation Start Phase for Bottom Field 0 to 4095 (0 to approx. 1.0)

Note: This register is updated when the SC0_SCL0_VEN_B bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

31.2.25 Trimming Register (SC0_SCL0_US7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC0_RES_HCUT[7:0]								SC0_RES_VCUT[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 8	SC0_RES_HCUT[7:0]	0	R/W	Horizontal Amount of Cut-off Post-Scaling Image (Right and Left Parts) Sets the number of pixel-clock cycles.
7 to 0	SC0_RES_VCUT[7:0]	0	R/W	Vertical Amount of Cut-off Post-Scaling Image (Upper and Lower Parts) Sets the number of lines.

Note: This register is updated when the SC0_SCL0_VEN_B bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

chanhvinh.com

31.2.26 Frame Buffer Read Select Register (SC0_SCL0_US8)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SC0_RES_IBUS_SYNC_SEL	—	—	—	SC0_RES_DISP_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SC0_RES_IBUS_SYNC_SEL	0	R/W	Sync Signal Select for Frame Buffer Read Block 0: Sync signals from the scaling-up control block 1: Sync signals from the graphics processing block
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC0_RES_DISP_ON	0	R/W	Post-Scaling Image Frame Display On/Off 0: Frame display on 1: Frame display off

Note: SC0_RES_IBUS_SYNC_SEL is updated when the SC0_SCL0_VEN_B and SC0_SCL0_VEN_D bits in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) are 1.
SC0_RES_DISP_ON is updated when the SC0_SCL0_VEN_B bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

31.2.27 Background Color Register (SC0_SCL0_OVR1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	SC0_RES_BK_COL_R[7:0]							
Initial value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC0_RES_BK_COL_G[7:0]								SC0_RES_BK_COL_B[7:0]							
Initial value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	SC0_RES_BK_CLO_R[7:0]	128	R/W	Background Color Setting R/Cr Signal R: 8 bits; unsigned (0 to 255 [LSB]) Cr: 8 bits; 128 offset binary; unsigned (0 to 255 [LSB])
15 to 8	SC0_RES_BK_COL_G[7:0]	0	R/W	Background Color Setting G/Y Signal G/Y: 8 bits; unsigned (0 to 255 [LSB])

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SC0_RES_BK_COL_B[7:0]	128	R/W	Background Color Setting B/Cb Signal B: 8 bits; unsigned (0 to 255 [LSB]) Cb: 8 bits; 128 offset binary; unsigned (0 to 255 [LSB])

Note: This register is updated when the SC0_SCL0_VEN_B bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

31.2.28 SCL1 Register Update Control Register (SC0_SCL1_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	SC0_SCL1_UPDATE_B	—	—	—	SC0_SCL1_UPDATE_A
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/WC1	R	R	R	R/WC1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SC0_SCL1_VEN_B	—	—	—	SC0_SCL1_VEN_A
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/WC1	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	SC0_SCL1_UPDATE_B	0	R/WC1	Frame Buffer Write Control Register Update 0: Registers are not updated. 1: Registers are updated.
19 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	SC0_SCL1_UPDATE_A	0	R/WC1	Frame Buffer Write Control Register Update 0: Registers are not updated. 1: Registers are updated.
15 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SC0_SCL1_VEN_B	0	R/WC1	Frame Buffer Write Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC0_SCL1_VEN_A	0	R/WC1	Frame Buffer Write Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

chanhvinh.com

31.2.29 Writing Mode Register (SC0_SCL1_WR1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	SC0_RES_WRSWA [2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SC0_RES_TB_ADD_MOD	SC0_RES_DS_WR_MD[2:0]			SC0_RES_MD[1:0]		SC0_RES_LOOP	SC0_RES_BST_MD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 19	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
18 to 16	SC0_RES_WRSWA[2:0]	All 0	R/W	8-Bit, 16-Bit, or 32-Bit Swap Setting These bits control swapping in frame buffer writing as follows. Bit 0 0: Swapped in 8-bit units. 1: Not swapped in 8-bit units. Bit 1 0: Swapped in 16-bit units. 1: Not swapped in 16-bit units. Bit 2 0: Swapped in 32-bit units. 1: Not swapped in 32-bit units. According to the setting of these bits, data is swapped as follows. Each number in parentheses ((1) to (8)) indicates 8-bit data. 000: (1) (2) (3) (4) (5) (6) (7) (8) [Not swapped] 001: (2) (1) (4) (3) (6) (5) (8) (7) [Swapped in 8-bit units] 010: (3) (4) (1) (2) (7) (8) (5) (6) [Swapped in 16-bit units] 011: (4) (3) (2) (1) (8) (7) (6) (5) [Swapped in 16-bit units + 8-bit units] 100: (5) (6) (7) (8) (1) (2) (3) (4) [Swapped in 32-bit units] 101: (6) (5) (8) (7) (2) (1) (4) (3) [Swapped in 32-bit units + 8-bit units] 110: (7) (8) (5) (6) (3) (4) (1) (2) [Swapped in 32-bit units + 16-bit units] 111: (8) (7) (6) (5) (4) (3) (2) (1) [Swapped in 32-bit units + 16-bit units + 8-bit units] Note: When YCbCr422 or RGB565 is selected as a frame buffer video-signal writing format, these bits should be set to 000 [Not swapped].
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	SC0_RES_TB_ADD_MOD	0	R/W	Top and Bottom Data Write Address Specification Method 0: A write address is specified in common for top and bottom data. 1: Separate write addresses are specified for top and bottom data.
6 to 4	SC0_RES_DS_WR_MD[2:0]	0	R/W	Frame Buffer Writing Mode for Image Processing 0: Normal 1: Horizontal mirroring 2: 90° rotation 3: 180° rotation 4: 270° rotation 5 to 7: Setting prohibited

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
3, 2	SC0_RES_MD [1:0]	0	R/W	Frame Buffer Video-Signal Writing Format 0: YCbCr422 (16 bits) 1: RGB565 (16 bits) 2: RGB888 (24 (32) bits) 3: YCbCr444 (24 (32) bits)
1	SC0_RES_ LOOP	0	R/W	Frame Buffer Write Mode Select 0: Frame mode 1: Line mode (read as ring buffer)
0	SC0_RES_ BST_MD	0	R/W	Transfer Burst Length for Frame Buffer Writing 0: 32-byte 1: 128-byte

Note: SC0_RES_LOOP and SC0_RES_BST_MD are updated when the SC0_SCL1_VEN_B bit in the SC0_SCL1 register update control register (SC0_SCL1_UPDATE) is 1.
SC0_RES_TB_ADD_MOD, SC0_RES_DS_WR_MD, and SC0_RES_MD are updated when the SC0_SCL1_VEN_A and SC0_SCL1_VEN_B bits in the SC0_SCL1 register update control register (SC0_SCL1_UPDATE) are 1.
SC0_RES_WRSWA is updated when the SC0_SCL1_UPDATE_A bit in the SC0_SCL1 register update control register (SC0_SCL1_UPDATE) is 1.

31.2.30 Write Address Register 1T (SC0_SCL1_WR2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SC0_RES_BASE[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC0_RES_BASE[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	SC0_RES_ BASE [31:0]	0	R/W	Frame Buffer Base Address Sets the start address of the frame buffer to store the frame data for the top field when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 1 or that for the top and bottom fields when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 0. For 32-byte transfer: The lower five bits should be fixed to 0 0000. For 128-byte transfer: The lower seven bits should be fixed to 000 0000.

Note: This register is updated when the SC0_SCL1_VEN_B bit in the SC0_SCL1 register update control register (SC0_SCL1_UPDATE) is 1.

chanhvinh.com

31.2.31 Write Address Register 2T (SC0_SCL1_WR3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	SC0_RES_LN_OFF[14:0]														
Initial value:	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SC0_RES_FLM_NUM[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 16	SC0_RES_LN_OFF [14:0]	2048	R/W	Frame Buffer Line Offset Address Sets the line offset address for calculating the line start address for the top field when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 1 or that for the top and bottom fields when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 0. Line 0: SC0_RES_BASE Line 1: SC0_RES_BASE + SC0_RES_LN_OFF × 1 : Line n: SC0_RES_BASE + SC0_RES_LN_OFF × n For 32-byte transfer: The lower five bits should be fixed to 0 0000. For 128-byte transfer: The lower seven bits should be fixed to 000 0000.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	SC0_RES_FLM_NUM [9:0]	1	R/W	Number of Frames of Buffer to be Written to Sets the number of frames for the top field when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 1 or that for the top and bottom fields when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 0 Number of frames defined by SC0_RES_FLM_NUM + 1 are used.

Note: This register is updated when the SC0_SCL1_VEN_B bit in the SC0_SCL1 register update control register (SC0_SCL1_UPDATE) is 1.

chanhvinh.com

31.2.32 Write Address Register 3T (SC0_SCL1_WR4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	SC0_RES_FLM_OFF[22:16]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC0_RES_FLM_OFF[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
22 to 0	SC0_RES_FLM_OFF [22:0]	524288	R/W	Frame Buffer Frame Offset Address Sets the frame offset address for calculating the start address of each frame for the top field when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 1 or that for the top and bottom fields when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 0. Buffer 0: SC0_RES_BASE Buffer 1: SC0_RES_BASE + SC0_RES_FLM_OFF × 1 : Buffer n: SC0_RES_BASE + SC0_RES_FLM_OFF × n For 32-byte transfer: The lower five bits should be fixed to 0 0000. For 128-byte transfer: The lower seven bits should be fixed to 000 0000.

Note: This register is updated when the SC0_SCL1_VEN_B bit in the SC0_SCL1 register update control register (SC0_SCL1_UPDATE) is 1.

chanhvinh.com

31.2.33 Frame Sub-Sampling Register (SC0_SCL1_WR5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	SC0_RES_INTER	—	—	SC0_RES_FS_RATE[1:0]		—	—	—	SC0_RES_FLD_SEL	—	—	—	SC0_RES_WENB
Initial value:	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R/W	R/W	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	SC0_RES_INTER	1	R/W	Field Operating Mode Select 0: Progressive 1: Interlace
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	SC0_RES_FS_RATE[1:0]	0	R/W	Writing Rate Sets the frame buffer writing rate to the vertical frequency of the input signal. 0: 1/1 an input signal (The SC0_RES_FLD_SEL setting is invalid.) 1: 1/2 an input signal 2: 1/4 an input signal 3: 1/8 an input signal
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SC0_RES_FLD_SEL	0	R/W	Write Field Select 0: Top field 1: Bottom field
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC0_RES_WENB	0	R/W	Frame Buffer Write Enable After making the setting to enable writing, writing starts from the second frame. 0: Frame buffer writing is disabled. 1: Frame buffer writing is enabled.

Note: SC0_RES_INTER, SC0_RES_FS_RATE[1:0], and SC0_RES_FLD_SEL are updated when the SC0_SCL1_VEN_A bit in the SC0_SCL1 register update control register (SC0_SCL1_UPDATE) is 1.

SC0_RES_WENB is updated when the SC0_SCL1_VEN_A and SC0_SCL1_VEN_B bits in the SC0_SCL1 register update control register (SC0_SCL1_UPDATE) are 1.

chanhvinh.com

31.2.34 Bit Reduction Register (SC0_SCL1_WR6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SC0_RES_DTH_ON	—	—	—	SC0_RES_BITDEC_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SC0_RES_DTH_ON	0	R/W	Dither Correction On/Off 0: Off (rounded off) 1: On (2 × 2 dither pattern)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC0_RES_BITDEC_ON	0	R/W	Bit Reduction On/Off 0: Off 1: On

Note: This register is updated when the SC0_SCL1_VEN_A bit in the SC0_SCL1 register update control register (SC0_SCL1_UPDATE) is 1.

31.2.35 Write Detection Register (SC0_SCL1_WR7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SC0_RES_OVERFLOW
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SC0_RES_FLM_CNT[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	SC0_RES_OVERFLOW	0	R	Line Buffer Overflow Detect 1: Line buffer has overflowed. 0: Line buffer has not overflowed.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	SC0_RES_FLM_CNT[9:0]	0	R	Frame Number of Frame Being Accessed Frame number of the frame being accessed in the top field when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 1 or that in the top or bottom field when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 0.

chanhvinh.com

31.2.36 Write Address Register 1B (SC0_SCL1_WR8)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SC0_RES_BASE_B[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC0_RES_BASE_B[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	SC0_RES_BASE_B [31:0]	0	R/W	Frame Buffer Base Address for Bottom Sets the start address of the frame buffer to store the frame data for the bottom field when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 1. For 32-byte transfer: The lower five bits should be fixed to 0_0000. For 128-byte transfer: The lower seven bits should be fixed to 000_0000.

Note: This register is updated when the SC0_SCL1_VEN_B bit in the SC0_SCL1 register update control register (SC0_SCL1_UPDATE) is 1.

chanhvinh.com

31.2.37 Write Address Register 2B (SC0_SCL1_WR9)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	SC0_RES_LN_OFF_B[14:0]														
Initial value:	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SC0_RES_FLM_NUM_B[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 16	SC0_RES_LN_OFF_B[14:0]	2048	R/W	Frame Buffer Line Offset Address for Bottom Sets the line offset address for calculating the line start address for the bottom field when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 1. Line 0: SC0_RES_BASE_B Line 1: SC0_RES_BASE_B + SC0_RES_LN_OFF_B × 1 : Line n: SC0_RES_BASE + SC0_RES_LN_OFF_B × n For 32-byte transfer: The lower five bits should be fixed to 0_0000. For 128-byte transfer: The lower seven bits should be fixed to 000_0000.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	SC0_RES_FLM_NUM_B[9:0]	1	R/W	Number of Frames of Buffer to be Written to for Bottom Field Number of frames defined by SC0_RES_FLM_NUM_B + 1 are used when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 1.

Note: This register is updated when the SC0_SCL1_VEN_B bit in the SC0_SCL1 register update control register (SC0_SCL1_UPDATE) is 1.

chanhvinh.com

31.2.38 Write Address Register 3B (SC0_SCL1_WR10)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	SC0_RES_FLM_OFF_B[22:16]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC0_RES_FLM_OFF_B[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
22 to 0	SC0_RES_FLM_OFF_B[22:0]	524288	R/W	Frame Buffer Frame Offset Address for Bottom Sets the frame offset address for calculating the start address of each frame for the bottom field when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 1. Buffer 0: SC0_RES_BASE_B Buffer 1: SC0_RES_BASE_B + SC0_RES_FLM_OFF_B × 1 : Buffer n: SC0_RES_BASE_B + SC0_RES_FLM_OFF_B × n For 32-byte transfer: The lower five bits should be fixed to 0_0000. For 128-byte transfer: The lower seven bits should be fixed to 000_0000.

Note: This register is updated when the SC0_SCL1_VEN_B bit in the SC0_SCL1 register update control register (SC0_SCL1_UPDATE) is 1.

31.2.39 Write Detection Register B (SC0_SCL1_WR11)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SC0_RES_FLM_CNT_B[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	SC0_RES_FLM_CNT_B[9:0]	0	R	Frame Number of Frame Being Accessed in Bottom Field Frame number of the frame being accessed in the bottom field when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 1.

chanhvinh.com

31.2.40 Status Monitor 1 Register (SC0_SCL1_MON1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SC0_PBUF_NUM[1:0]		—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	SC0_PBUF_NUM[1:0]	All 0	R	Write pointer indicating the pointer buffer number corresponding to the location currently being written to.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

31.2.41 Pointer Buffer 0 Register (SC0_SCL1_PBUF0)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SC0_PBUF0_ADD[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC0_PBUF0_ADD[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	SC0_PBUF0_ADD[31:0]	All 0	R	Start address of the write buffer pointed to by pointer buffer 0.

chanhvinh.com

31.2.42 Pointer Buffer 1 Register (SC0_SCL1_PBUF1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SC0_PBUF1_ADD[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC0_PBUF1_ADD[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	SC0_PBUF1_ ADD[31:0]	All 0	R	Start address of the write buffer pointed to by pointer buffer 1.

31.2.43 Pointer Buffer 2 Register (SC0_SCL1_PBUF2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SC0_PBUF2_ADD[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC0_PBUF2_ADD[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	SC0_PBUF2_ ADD[31:0]	All 0	R	Start address of the write buffer pointed to by pointer buffer 2.

chanhvinh.com

31.2.44 Pointer Buffer 3 Register (SC0_SCL1_PBUF3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SC0_PBUF3_ADD[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC0_PBUF3_ADD[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	SC0_PBUF3_ ADD[31:0]	All 0	R	Start address of the write buffer pointed to by pointer buffer 3.

31.2.45 Pointer Buffer and Field Information Register (SC0_SCL1_PBUF_FLD)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	SC0_FLD_INF3	—	—	—	—	—	—	—	SC0_FLD_INF2
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SC0_FLD_INF1	—	—	—	—	—	—	—	SC0_FLD_INF0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	SC0_FLD_ INF3	0	R	Top or bottom field information of the frame buffer pointed to by pointer buffer 3. 0: Bottom 1: Top
23 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	SC0_FLD_ INF2	0	R	Top or bottom field information of the frame buffer pointed to by pointer buffer 2. 0: Bottom 1: Top
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	SC0_FLD_ INF1	0	R	Top or bottom field information of the frame buffer pointed to by pointer buffer 1. 0: Bottom 1: Top
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
0	SC0_FLD_INF0	0	R	Top or bottom field information of the frame buffer pointed to by pointer buffer 0. 0: Bottom 1: Top

31.2.46 Pointer Buffer Control Register (SC0_SCL1_PBUF_CNT)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SC0_PBUF_RST
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	SC0_PBUF_RST	0	R/W	Reset Control for Pointer Buffer 0: Pointer buffer is not reset. 1: Pointer buffer is reset.
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: This register is updated when the SC0_SCL1_UPDATE_B bit in the SC0_SCL1 register update control register (SC0_SCL1_UPDATE) is 1.

chanhvinh.com

31.2.47 Graphics 0 Register Update Control Register (GR0_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GR0_UPDATE	—	—	—	GR0_P_VEN	—	—	—	GR0_IBUS_VEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/WC1	R	R	R	R/WC1	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	GR0_UPDATE	0	R/WC1	Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	GR0_P_VEN	0	R/WC1	Graphics Display Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR0_IBUS_VEN	0	R/WC1	Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

31.2.48 Frame Buffer Read Control Register (Graphics 0) (GR0_FLM_RD)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR0_R_ENB
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR0_R_ENB	0	R/W	Frame Buffer Read Enable 0: Frame buffer reading is disabled. 1: Frame buffer reading is enabled.

Note: This register is updated when the GR0_IBUS_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

chanhvinh.com

31.2.49 Frame Buffer Control Register 1 (Graphics 0) (GR0_FLM1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR0_FLD_SEL	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR0_LN_OFF_DIR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	GR0_FLM_SEL[1:0]		—	—	—	—	—	—	—	GR0_BST_MD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	GR0_FLD_SEL	0	R/W	Enables or disables top or bottom field selection for the next frame buffer to be read. 0: Field selection is disabled. 1: Top or bottom field can be selected.
30 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	GR0_LN_OFF_DIR	0	R/W	Selects the line offset address direction of the frame buffer. 0: Increments the address by the line offset address. 1: Decrements the address by the line offset address.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	GR0_FLM_SEL[1:0]	0	R/W	Selects a frame buffer address setting signal. 0: Links to scaling-down process. (This setting is prohibited when separate write addresses are specified for the top and bottom fields; that is, SC0_RES_TB_ADD_MOD = 1 in SC0_SCL1_WR1.) 1: Selects GR0_FLM_NUM. 2: Setting prohibited 3: Links to pointer buffer.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR0_BST_MD	0	R/W	Frame Buffer Burst Transfer Mode 0: 32-byte transfer 1: 128- byte transfer

Note: GR0_FLD_SEL is updated when the GR0_P_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.
GR0_LN_OFF_DIR is updated when the GR0_IBUS_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.
GR0_FLM_SEL is updated when the GR0_P_VEN and GR0_IBUS_VEN bits in the graphics 0 register update control register (GR0_UPDATE) are 1.
GR0_BST_MD is updated when the GR0_IBUS_VEN and GR0_P_VEN bits in the graphics 0 register update control register (GR0_UPDATE) are 1.

chanhvinh.com

31.2.50 Frame Buffer Control Register 2 (Graphics 0) (GR0_FLM2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR0_BASE[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR0_BASE[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	GR0_BASE [31:0]	0	R/W	Frame Buffer Base Address Sets the start address of the frame buffer where frame data is to be stored. GR0_BASE[4:3] and GR0_BASE[6:3] are referred to during 32-byte burst transfer and 128-byte burst transfer, respectively, to skip the start line data. The lower three bits should be fixed to 000.

Note: This register is updated when the GR0_IBUS_VEN and GR0_P_VEN bits in the graphics 0 register update control register (GR0_UPDATE) are 1.

chanhvinh.com

31.2.51 Frame Buffer Control Register 3 (Graphics 0) (GR0_FLM3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR0_FLD_NXT	GR0_LN_OFF[14:0]														
Initial value:	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	GR0_FLM_NUM[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	GR0_FLD_NXT	0	R/W	Top or Bottom Field Selection for Next Frame Buffer 0: Bottom 1: Top
30 to 16	GR0_LN_OFF[14:0]	2048	R/W	Frame Buffer Line Offset Address Sets the line offset address for calculating the start address of each line. Line 0: GR0_BASE Line 1: GR0_BASE + GR0_LN_OFF × 1 : Line n: GR0_BASE + GR0_LN_OFF × n For 32-byte transfer: The lower five bits should be fixed to 0 0000. For 128-byte transfer: The lower seven bits should be fixed to 000 0000.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	GR0_FLM_NUM[9:0]	1	R/W	Frame Number of Frame Buffer Manually set the frame number when GR0_FLM_SEL = 1.

Note: GR0_FLD_NXT is updated when the GR0_P_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.
GR0_LN_OFF[14:0] and GR0_FLM_NUM[9:0] are updated when the GR0_IBUS_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

chanhvinh.com

31.2.52 Frame Buffer Control Register 4 (Graphics 0) (GR0_FLM4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	GR0_FLM_OFF[22:16]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR0_FLM_OFF[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
22 to 0	GR0_FLM_OFF[22:0]	524288	R/W	Frame Buffer Frame Offset Address Specifies the frame offset address used for calculating the start address of each frame buffer when more than one buffer is used. Buffer 0: GR0_BASE Buffer 1: GR0_BASE + GR0_FLM_OFF × 1 : Buffer n: GR0_BASE + GR0_FLM_OFF × n For 32-byte transfer: The lower five bits should be fixed to 0 0000. For 128-byte transfer: The lower seven bits should be fixed to 000 0000.

Note: This register is updated when the GR0_IBUS_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

chanhvinh.com

31.2.53 Frame Buffer Control Register 5 (Graphics 0) (GR0_FLM5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR0_FLM_LNUM[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR0_FLM_LOOP[10:0]										
Initial value:	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR0_FLM_LNUM[10:0]	0	R/W	Sets number of lines in a frame Number of lines is (GR0_FLM_LNUM + 1).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR0_FLM_LOOP[10:0]	1023	R/W	Number of lines when reading the addresses repeatedly by returning to the start address after reaching the end address. (GR0_FLM_LOOP + 1) lines are read.

Note: This register is updated when the GR0_IBUS_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

chanhvinh.com

31.2.54 Frame Buffer Control Register 6 (Graphics 0) (GR0_FLM6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR0_FORMAT[3:0]				—	GR0_HW[10:0]										
Initial value:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR0_YCC_SWAP[2:0]			GR0_RDSWA[2:0]			—	GR0_CNV444_MD	—	—	GR0_STA_POS[5:0]					
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	GR0_FORMAT[3:0]	8	R/W	Sets the format of the frame buffer read signal. 0: RGB565 1: RGB888 2: αRGB1555 3: αRGB4444 4: αRGB8888 5: CLUT8 6: CLUT4 7: CLUT1 8: YCbCr422 9: YCbCr444 10: RGBα5551 11: RGBα8888 12 to 15: Setting prohibited
27	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
26 to 16	GR0_HW[10:0]	0	R/W	Sets the width of the horizontal valid period. The width is (GR0_HW + 1) pixels. Note: The set value should be equal to or more than two.
15 to 13	GR0_YCC_SWAP[2:0]	0	R/W	Controls swapping of data read from buffer in the YCbCr422 format. 0: Cb/Y0/Cr/Y1 1: Y0/Cb/Y1/Cr 2: Cr/Y0/Cb/Y1 3: Y0/Cr/Y1/Cb 4: Y1/Cr/Y0/Cb 5: Cr/Y1/Cb/Y0 6: Y1/Cb/Y0/Cr 7: Cb/Y1/Cr/Y0

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
12 to 10	GR0_ RDSWA [2:0]	0	R/W	8-Bit, 16-Bit, or 32-Bit Swap Setting These bits control swapping in frame buffer reading as follows. Bit 0 0: Swapped in 8-bit units. 1: Not swapped in 8-bit units. Bit 1 0: Swapped in 16-bit units. 1: Not swapped in 16-bit units. Bit 2 0: Swapped in 32-bit units. 1: Not swapped in 32-bit units. According to the setting of these bits, data is swapped as follows. Each number in parentheses ((1) to (8)) indicates 8-bit data. 000: (1) (2) (3) (4) (5) (6) (7) (8) [Not swapped] 001: (2) (1) (4) (3) (6) (5) (8) (7) [Swapped in 8-bit units] 010: (3) (4) (1) (2) (7) (8) (5) (6) [Swapped in 16-bit units] 011: (4) (3) (2) (1) (8) (7) (6) (5) [Swapped in 16-bit units + 8-bit units] 100: (5) (6) (7) (8) (1) (2) (3) (4) [Swapped in 32-bit units] 101: (6) (5) (8) (7) (2) (1) (4) (3) [Swapped in 32-bit units + 8-bit units] 110: (7) (8) (5) (6) (3) (4) (1) (2) [Swapped in 32-bit units + 16-bit units] 111: (8) (7) (6) (5) (4) (3) (2) (1) [Swapped in 32-bit units + 16-bit units + 8-bit units]
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	GR0_ CNV444_ MD	0	R/W	Sets the interpolation mode for YCbCr422 to YCbCr444 conversion. 0: Hold interpolation 1: Average interpolation
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5 to 0	GR0_STA_ POS[5:0]	0	R/W	Sets the amount of data to be skipped through. Specifically data amount equal to the amount indicated by GR0_STA_POS is skipped from the start of the line.

Note: GR0_YCC_SWAP, GR0_CNV444, and GR0_STA_POS are updated when GR0_P_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.
GR0_ RDSWA is updated when the GR0_UPDATE bit in the graphics 0 register update control register (GR0_UPDATE) is 1.
GR0_FORMAT and GR0_HW are updated when GR0_IBUS_VEN and GR0_P_VEN bits in the graphics 0 register update control register (GR0_UPDATE) are 1.

chanhvinh.com

31.2.55 Alpha Blending Control Register 1 (Graphics 0) (GR0_AB1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	GR0_GRC_DISP_ON	—	—	GR0_DISP_SEL[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	GR0_GRC_DISP_ON	0	R/W	Turns on/off frame-line display of the graphics image area. 0: Frame-line display off 1: Frame-line display on
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	GR0_DISP_SEL[1:0]	0	R/W	Selects the graphics display mode. 0: Background color display (GR0_BASE) 1: Lower-layer graphics display When displaying video image or enlarged graphics, select this setting. 2: Current graphics display When displaying graphics, select this setting. 3: Blended display of lower-layer graphics and current graphics* Note: * Select this setting whenever chroma-key processing is to proceed. Since only current graphics are to be displayed by chroma-key processing, set the α values for both pixels to be subject to chroma-keying and pixels not to be subject to chroma-keying to 255.

Note: This register is updated when GR0_P_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

chanhvinh.com

31.2.56 Alpha Blending Control Register 2 (Graphics 0) (GR0_AB2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR0_GRC_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR0_GRC_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR0_GRC_VS[10:0]	0	R/W	Vertical Start Position of Graphics Image Area. Note: The set value should be four or more (lines). GR0_GRC_VS + GR0_GRC_VW should be equal to or less than 2039 (lines).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR0_GRC_VW[10:0]	0	R/W	Vertical Width of Graphics Image Area.

Note: This register is updated when the GR0_P_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

31.2.57 Alpha Blending Control Register 3 (Graphics 0) (GR0_AB3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR0_GRC_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR0_GRC_HW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR0_GRC_HS[10:0]	0	R/W	Horizontal Start Position of Graphics Image Area. Note: The set value should be 16 or more (clock cycles). GR0_GRC_HS + GR0_GRC_HW should be equal to or less than 2015 (clock cycles).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR0_GRC_HW[10:0]	0	R/W	Horizontal Width of Graphics Image Area. Note: For displaying an image with 1- or 2-pixel horizontal width, set GR0_HW to 2 and GR0_GRC_HW to 1 (1 pixel) or 2 (2 pixels).

chanhvinh.com

Note: This register is updated when the GR0_P_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

31.2.58 Alpha Blending Control Register 7 (Graphics 0) (GR0_AB7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR0_CK_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR0_CK_ON	0	R/W	CLUT-Index/RGB-Index Chroma-Key Processing On/Off 0: Off 1: On

Note: This register is updated when the GR0_P_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

chanhvinh.com

31.2.59 Alpha Blending Control Register 8 (Graphics 0) (GR0_AB8)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR0_CK_KCLUT[7:0]								GR0_CK_KG[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR0_CK_KB[7:0]								GR0_CK_KR[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR0_CK_KCLUT[7:0]	0	R/W	CLUT Signal for CLUT-Index Chroma-Key Processing CLUT: Unsigned 8 bits (0 to 255 [LSB])
23 to 16	GR0_CK_KG[7:0]	0	R/W	G Signal for RGB-Index Chroma-Key Processing G: Unsigned 8 bits (0 to 255 [LSB])
15 to 8	GR0_CK_KB[7:0]	0	R/W	B Signal for RGB-Index Chroma-Key Processing B: Unsigned 8 bits (0 to 255 [LSB])
7 to 0	GR0_CK_KR[7:0]	0	R/W	R Signal for RGB-Index Chroma-Key Processing R: Unsigned 8 bits (0 to 255 [LSB])

Note: This register is updated when the GR0_P_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

31.2.60 Alpha Blending Control Register 9 (Graphics 0) (GR0_AB9)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR0_CK_A[7:0]								GR0_CK_G[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR0_CK_B[7:0]								GR0_CK_R[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR0_CK_A[7:0]	0	R/W	Replaced Alpha Signal after RGB-Index Chroma-Key Processing α: Unsigned 8 bits (0 to 255 [LSB]) Note: These bits should always be set to 255 to display the current graphics only.
23 to 16	GR0_CK_G[7:0]	0	R/W	Replaced G Signal after RGB-Index Chroma-Key Processing G: Unsigned 8 bits (0 to 255 [LSB])
15 to 8	GR0_CK_B[7:0]	0	R/W	Replaced B Signal after RGB-Index Chroma-Key Processing B: Unsigned 8 bits (0 to 255 [LSB])
7 to 0	GR0_CK_R[7:0]	0	R/W	Replaced R Signal after RGB-Index Chroma-Key Processing R: Unsigned 8 bits (0 to 255 [LSB])

Note: This register is updated when the GR0_P_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

chanhvinh.com

31.2.61 Alpha Blending Control Register 10 (Graphics 0) (GR0_AB10)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR0_A0[7:0]								GR0_G0[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR0_B0[7:0]								GR0_R0[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR0_A0 [7:0]	0	R/W	CLUT1 α0 Signal Replaced with α signal when in the CLUT1 format and CLUT1 = 0. Replaced with α signal when in the αRGB1555/RGBα5551 format and α = 0. Note: These bits should always be set to 255 to display the current graphics only.
23 to 16	GR0_G0 [7:0]	0	R/W	CLUT1 G0 Signal Replaced with G signal when in the CLUT1 format and CLUT1 = 0.
15 to 8	GR0_B0 [7:0]	0	R/W	CLUT1 B0 Signal Replaced with B signal when in the CLUT1 format and CLUT1 = 0.
7 to 0	GR0_R0 [7:0]	0	R/W	CLUT1 R0 Signal Replaced with R signal when in the CLUT1 format and CLUT1 = 0.

Note: This register is updated when the GR0_P_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

chanhvinh.com

31.2.62 Alpha Blending Control Register 11 (Graphics 0) (GR0_AB11)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR0_A1[7:0]								GR0_G1[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR0_B1[7:0]								GR0_R1[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR0_A1 [7:0]	0	R/W	CLUT1 α1 Signal Replaced with α signal when in the CLUT1 format and CLUT1 = 1. Replaced with α signal when in the αRGB1555/RGBα5551 format and α = 1. Note: These bits should always be set to 255 to display the current graphics only.
23 to 16	GR0_G1 [7:0]	0	R/W	CLUT1 G1 Signal Replaced with G signal when in the CLUT1 format and CLUT1 = 1.
15 to 8	GR0_B1 [7:0]	0	R/W	CLUT1 B1 Signal Replaced with B signal when in the CLUT1 format and CLUT1 = 1.
7 to 0	GR0_R1 [7:0]	0	R/W	CLUT1 R1 Signal Replaced with R signal when in the CLUT1 format and CLUT1 = 1.

Note: This register is updated when the GR0_P_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

31.2.63 Background Color Control Register (Graphics 0) (GR0_BASE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	GR0_BASE_G[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR0_BASE_B[7:0]								GR0_BASE_R[7:0]							
Initial value:	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	GR0_BASE_ G[7:0]	0	R/W	Background Color G/Y Signal G/Y: Unsigned 8 bits (0 to 255 [LSB])
15 to 8	GR0_BASE_ B[7:0]	128	R/W	Background Color B/Cb Signal B: Unsigned 8 bits (0 to 255 [LSB]) Cb: 8 bits; 128 offset binary; unsigned (0 to 255 [LSB])

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	GR0_BASE_R[7:0]	128	R/W	Background Color R/Cr Signal R: Unsigned 8 bits (0 to 255 [LSB]) Cr: 8 bits; 128 offset binary; unsigned (0 to 255 [LSB])

Note: This register is updated when the GR0_P_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

31.2.64 CLUT Table Control Register (Graphics 0) (GR0_CLUT)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR0_CLT_SEL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	GR0_CLT_SEL	0	R/W	CLUT Table Select Signal 0: Selects CLUT table 0. Referring to the CLUT table 0 value to expand to αRGB8888 The CPU side can read-access or write-access to the CLUT table 1. 1: Selects CLUT table 1. Referring to the CLUT table 1 value to expand to αRGB8888 The CPU side can read-access or write-access to the CLUT table 0.
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: This register is updated when the GR0_P_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

chanhvinh.com

31.3 Usage Method

31.3.1 Scaling Setting Example for 525i Video Input and VGA-Size (640 x 480) Video Output

(1) Angles of View for Input and Output

This section describes an example of setting the signals of the input and output angles of view shown in Table 31.40.

Here, the over-scan rate is assumed to be 100%.

Table 31.40 Input and Output Angles for 525i Video Input and VGA-Size (640 x 480) Video Output

Input Signal	Output Signal	Signal Format	Rotation	Buffer Planes	Scaling Filter
1440 × 240	640 × 480	YCbCr	Normal	Two planes	2-tap linear

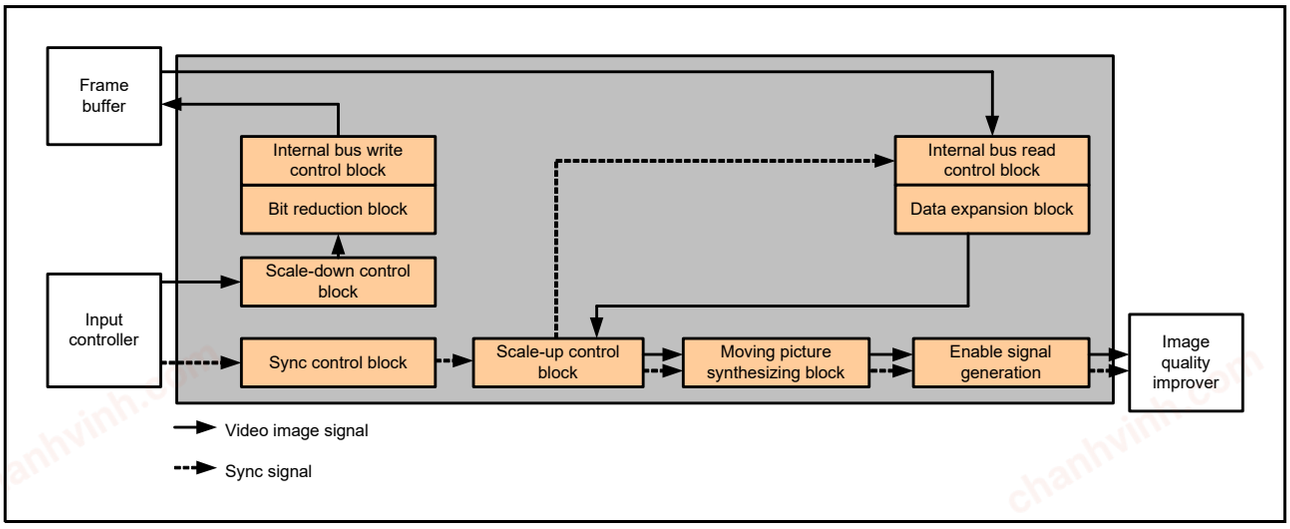


Figure 31.18 Signal Paths for Displaying Input Video Image

(2) Horizontal Scaling (Horizontal Scale Down, Scaling Filter: 2-Tap Linear)

The scaling rate for folding can be calculated as shown below.

RATIO_org = round (1440 ÷ 640 × 4096) = 9216

σ = (9216 × (640 − 1) − (1440 − 1) × 4096) ÷ (640 − 1) = −8.01

Horizontal scaling ratio = roundup (9216 − (−8.01)) = 9225

(3) Vertical Scaling (Vertical Scale Up, Scaling Filter: 2-Tap Linear)

The scaling rate for folding can be calculated as shown below.

RATIO_org = round (240 ÷ 480 × 4096) = 2048

σ = (2048 × (480 − 1) − (240 − 1) × 4096) ÷ (480 − 1) = 4.27

Vertical scaling ratio = round (2048 − (4.07)) = 2044

chanhvinh.com

(4) Setting Frame Buffer Access Area

Since video data is written to the frame buffer after scaled down, the write size is 640 × 240 pixels.

The frame buffer area required is 640 pixels or more for line offset and line offset × 240 pixels or more for frame offset.

Here, the frame buffer work area is assumed to be 1024 × 256 pixels.

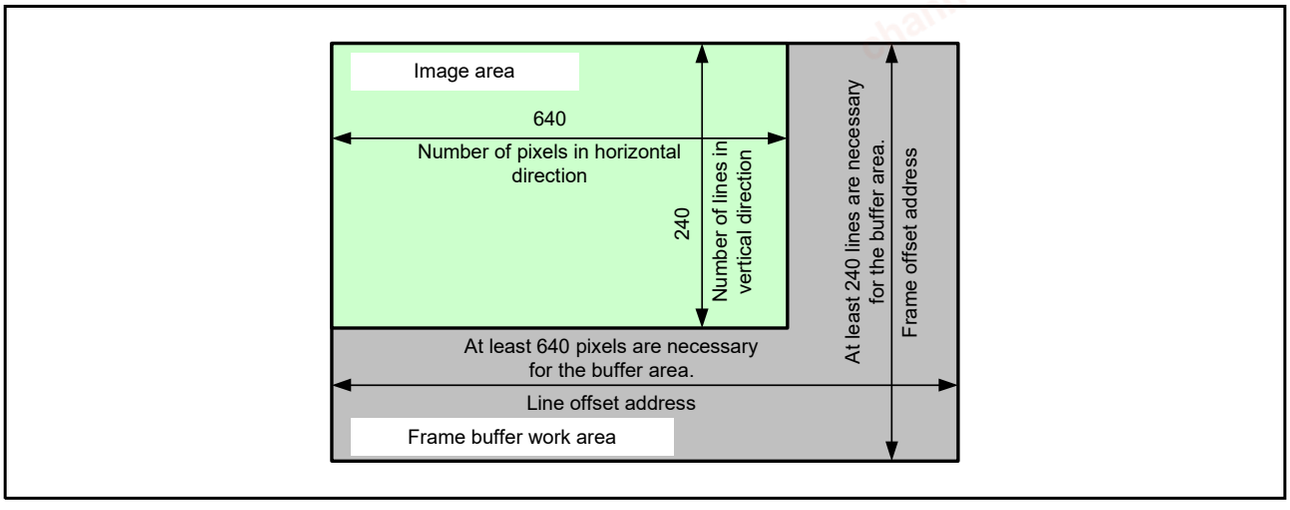


Figure 31.19 Frame Buffer Access Area Setting

Since the frame buffer is accessed in 64-bit units, YCbCr422 (16 bits) is accessed in 4-pixel units.

The line offset address values to be set are:

SC_RES_LN_OFF[14:0] = 1024 × 2 = 2048

GR_LN_OFF[14:0] = 1024 × 2 = 2048

The frame offset address values to be set are:

SC_RES_FLM_OFF[22:0] = SC_RES_LN_OFF[14:0] × 256 = 524288

GR_FLM_OFF[22:0] = GR_LN_OFF[14:0] × 256 = 524288

chanhvinh.com

(5) Register Setting Example

Table 31.41 Register Setting Example for 525i Video Input and VGA-Size Video Output

Register Name	Bit Name	Settings	Remarks
		Synchronization Control	
SC_SCL0_FRC3	SC_RES_VS_SEL	0	External Vsync selected
SC_SCL0_FRC4	SC_RES_FH[10:0]	799	Horizontal period width of output signal (period width = set value + 1)
		Size of Angle of View	
SC_SCL0_DS2	SC_RES_VS[10:0]	15	Vertical capture start position of input signal
SC_SCL0_DS2	SC_RES_VW[10:0]	240	Vertical capture width of input signal
SC_SCL0_DS3	SC_RES_HS[10:0]	244	Horizontal capture start position of input signal
SC_SCL0_DS3	SC_RES_HW[10:0]	1440	Horizontal capture width of input signal
SC_SCL0_FRC6	SC_RES_F_VS[10:0]	35	Vertical valid start position of full screen
SC_SCL0_FRC6	SC_RES_F_VW[10:0]	480	Vertical valid width of full screen
SC_SCL0_FRC7	SC_RES_F_HS[10:0]	144	Horizontal valid start position of full screen
SC_SCL0_FRC7	SC_RES_F_HW[10:0]	640	Horizontal valid width of full screen
SC_SCL0_US2	SC_RES_P_VS[10:0]	35	Vertical valid start position of output image
SC_SCL0_US2	SC_RES_P_VW[10:0]	480	Vertical valid width of output image
SC_SCL0_US3	SC_RES_P_HS[10:0]	144	Horizontal valid start position of output image
SC_SCL0_US3	SC_RES_P_HW[10:0]	640	Horizontal valid width of output image
		Scaling Setting	
SC_SCL0_DS4	SC_RES_DS_H_RATIO [15:0]	9224	Horizontal scaling-down because SC_RES_DS_H_RATIO is equal to or larger than 4096
SC_SCL0_DS1	SC_RES_DS_H_ON	1	Horizontal scaling-down on
SC_SCL0_US1	SC_RES_US_H_ON	0	Horizontal scaling-up off
SC_SCL0_US5	SC_RES_US_H_RATIO [15:0]	4096	Horizontal scaling-up off because SC_RES_US_H_RATIO is equal to or larger than 4096
SC_SCL0_DS1	SC_RES_DS_V_ON	0	Vertical scaling-down off
SC_SCL0_US1	SC_RES_US_V_ON	1	Vertical scaling-up on
SC_SCL0_DS6	SC_RES_V_RATIO[15:0]	2044	Vertical scaling-up because SC_RES_V_RATIO is smaller than 4096
SC_SCL0_DS7	SC_RES_OUT_VW[10:0]	240	Vertical valid input width because the vertical scaling-down function is off
SC_SCL0_DS7	SC_RES_OUT_HW[10:0]	640	Horizontal image size after horizontal scaling-down
SC_SCL0_US4	SC_RES_IN_VW[10:0]	240	Vertical width of frame buffer read
SC_SCL0_US4	SC_RES_IN_HW[10:0]	640	Horizontal width of frame buffer read
		IP Conversion Setting	
SC_SCL0_DS5	SC_RES_TOP_ INIPHASE[11:0]	2048	Top field adjusted by 0.5-line phase
SC_SCL0_DS5	SC_RES_BTM_ INIPHASE[11:0]	0	No phase adjustment for bottom field
SC_SCL0_FRC5	SC_RES_FLD_DLY_SEL	1	IP conversion with two planes of frame buffer used for vertical scaling-up
		Frame Buffer Write Setting	
SC_SCL1_WR1	SC_RES_DS_WR_MD [2:0]	0	Normal write mode for rotation control
SC_SCL1_WR1	SC_RES_MD[1:0]	0	Frame buffer write format YCbCr422 (16 bits)
SC_SCL1_WR2	SC_RES_BASE[31:0]	0	Frame buffer write start address (0 in setting example)
SC_SCL1_WR3	SC_RES_LN_OFF[14:0]	2048	Frame buffer write line offset

chanhvinh.com

Table 31.41 Register Setting Example for 525i Video Input and VGA-Size Video Output

Register Name	Bit Name	Settings	Remarks
SC_SCL1_WR3	SC_RES_FLM_NUM[9:0]	1	Two planes of frame buffer used
SC_SCL1_WR4	SC_RES_FLN_OFF[22:0]	524288	Frame buffer write frame offset
SC_SCL1_WR5	SC_RES_WENB	1	Frame buffer write enabled
Frame Buffer Read Setting			
GR_FLM1	GR_FLM_SEL[1:0]	0	Frame number for frame buffer write output
GR_FLM2	GR_BASE[31:0]	0	Conforming to frame buffer write setting
GR_FLM3	GR_LN_OFF[14:0]	2048	Conforming to frame buffer write setting
GR_FLM4	GR_FLM_OFF[22:0]	524288	Conforming to frame buffer write setting
GR_FLM6	GR_FORMAT[3:0]	8	Frame buffer read format YCbCr422
GR_FLM_RD	GR_R_ENB	1	Frame buffer read enabled
GR_FLM6	GR_CNV444_MD	1	Mean value interpolation in YCbCr422→ YCbCr444 conversion
Scaling-up Selection			
SC_SCL0_US8	SC_RES_IBUS_SYNC_SEL	0	Scaled-up video signal displayed
GR_AB1	GR_DISP_SEL[1:0]	1	Scaling display selected

31.3.2 Scaling Setting Example for Graphics Display

(1) Angle of View for Graphics Display

This section describes an example of setting the signals of the input and output angles of view shown in Table 31.42.

Table 31.42 Input and Output Angles of View for Graphics Display

Graphics Size	Output Signal	Graphics Signal Format
640 × 480	640 × 480	RGB888

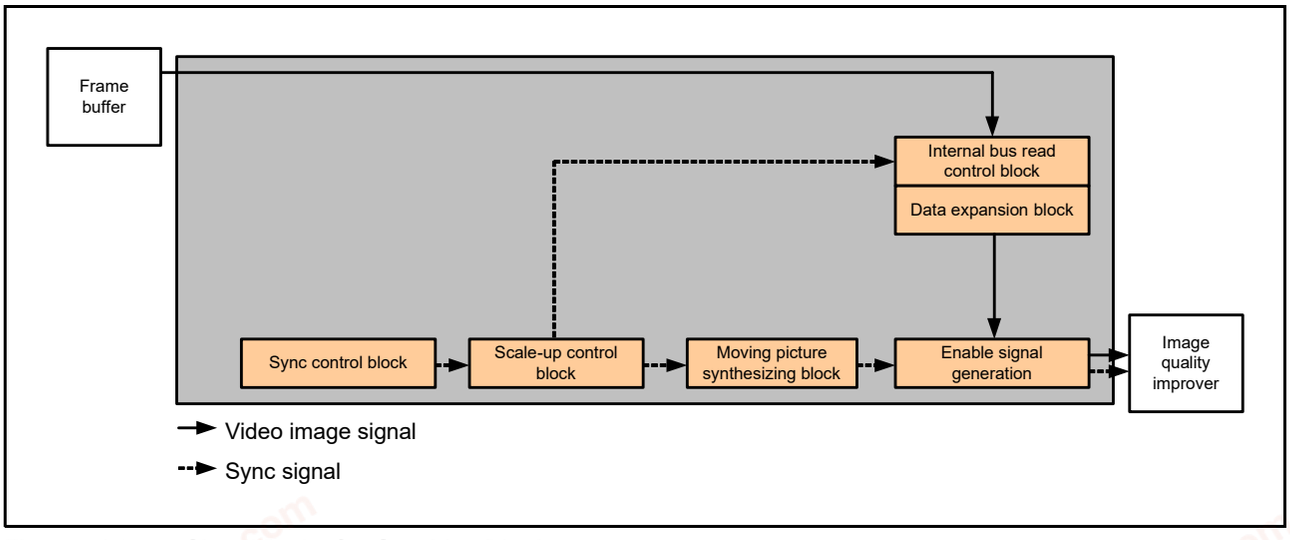


Figure 31.20 Signal Paths for Graphics Display

chanhvinh.com

(2) Setting Frame Buffer Access Area

In the frame buffer in which graphics data is stored, graphics data needs to be expanded in the area of 640 × 480 pixels or larger.

Here, the frame buffer area in which graphics data is expanded is assumed to be 640 × 480 pixels.

Since the frame buffer is accessed in 64-bit units, RGB888 (32 bits) is accessed in 2-pixel units.

The line offset address values to be set are:

GR_LN_OFF[14:0] = 640 × 4 = 2560

The frame offset address values to be set are:

GR_FLM_OFF[22:0] = GR_LN_OFF[14:0] × 480 = 1228800

(3) Register Setting Example

Table 31.43 Register Setting Example for Graphics Display

Register Name	Bit Name	Settings	Remarks
Synchronization Control			
SC_SCL0_FRC3	SC_RES_VS_SEL	1	Free-running Vsync selected (when the appropriate input signal is available, an external sync can also be selected)
SC_SCL0_FRC4	SC_RES_FV[10:0]	524	Vertical period width of output signal (period width = set value + 1)
SC_SCL0_FRC4	SC_RES_FH[10:0]	799	Horizontal period width of output signal (period width = set value + 1)
Size of Angle of View			
SC_SCL0_FRC6	SC_RES_F_VS[10:0]	35	Vertical valid start position of full screen
SC_SCL0_FRC6	SC_RES_F_VW[10:0]	480	Vertical valid width of full screen
SC_SCL0_FRC7	SC_RES_F_HS[10:0]	144	Horizontal valid start position of full screen
SC_SCL0_FRC7	SC_RES_F_HW[10:0]	640	Horizontal valid width of full screen
GR_AB2	GR_GRC_VS[10:0]	35	Vertical valid start position of graphics output
GR_AB2	GR_GRC_VW[10:0]	480	Vertical valid width of graphics output
GR_AB3	GR_GRC_HS[10:0]	144	Horizontal valid start position of graphics output
GR_AB3	GR_GRC_HW[10:0]	640	Horizontal valid width of graphics output
Frame Buffer Read Setting			
GR_FLM1	GR_FLM_SEL[1:0]	1	Frame number setting with register
GR_FLM3	GR_FLM_NUM[9:0]	0	Frame number of frame buffer (0 in setting example)
GR_FLM5	GR_FLM_LNUM[9:0]	479	Number of graphics lines (number of lines = set value + 1)
GR_FLM6	GR_HW[9:0]	639	Horizontal valid width of graphics (valid width = set value + 1)
GR_FLM2	GR_BASE[31:0]	0	Conforming to graphics expansion setting (0 in setting example)
GR_FLM3	GR_LN_OFF[14:0]	2560	Conforming to graphics expansion setting
Frame Buffer Read Setting			
GR_FLM4	GR_FLM_OFF[22:0]	1228800	Conforming to graphics expansion setting
GR_FLM6	GR_FORMAT[3:0]	1	Frame buffer read format RGB888
GR_FLM_RD	GR_R_ENB	1	Frame buffer read enabled

chanhvinh.com

Table 31.43 Register Setting Example for Graphics Display

Register Name	Bit Name	Settings	Remarks
	Scaling-up Selection		
SC_SCL0_US8	SC_RES_IBUS_SYNC_SEL	1	Graphics output displayed
GR_AB1	GR_DISP_SEL[1:0]	2	Graphics display selected

31.3.3 Scaling Setting Example for Scaled-up Graphics Display

(1) Angles of View for Input and Output

This section describes an example of setting the signals of the input and output angles of view shown in Table 31.44.

Table 31.44 Input and Output Angles of View for Scaled-up Graphics Display

Graphics Size	Output Signal	Graphics Signal Format
640 × 480	800 × 600	RGB565

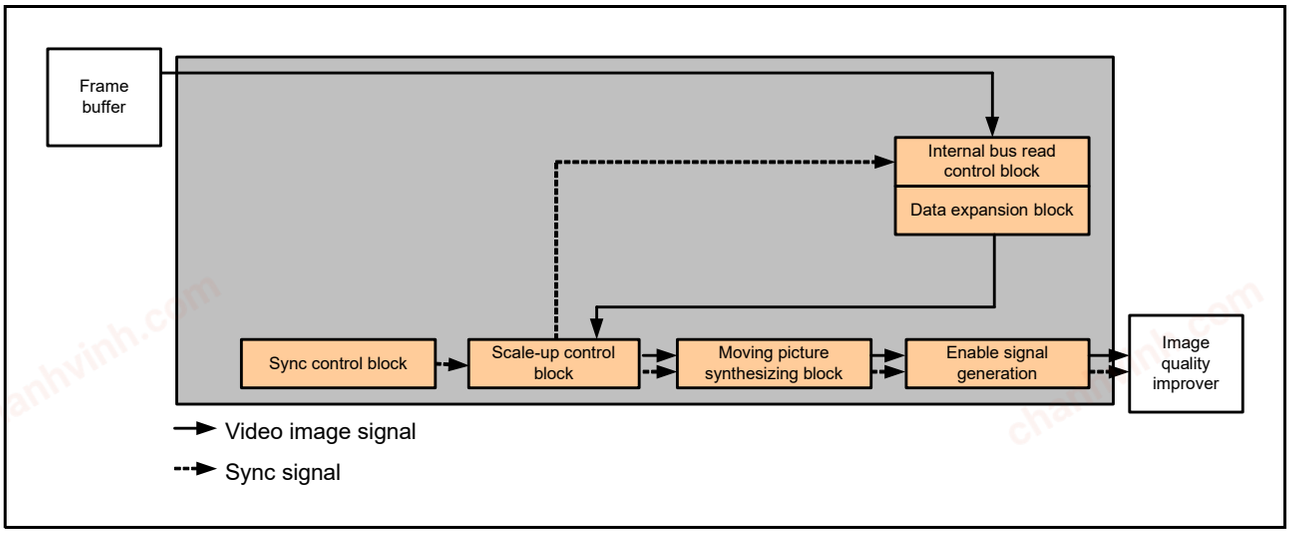


Figure 31.21 Signal Paths for Scaled-up Graphics Display

(2) Horizontal Scaling (Horizontal Scale Up, Scaling Filter: 2-Tap Linear)

The scaling rate for folding can be calculated as shown below.

RATIO_org = round (640 ÷ 800 × 4096) = 3277

σ = (3277 × (800 − 1) − (640 − 1) × 4096) ÷ (800 − 1) = 1.23

Horizontal scaling ratio = round (3277 − (1.23)) = 3276

chanhvinh.com

(3) Vertical Scaling (Vertical Scale Up, Scaling Filter: 2-Tap Linear)

The scaling rate for folding can be calculated as shown below.

RATIO_org = round (480 ÷ 600 × 4096) = 3277

σ = (3277 × (600 − 1) − (480 − 1) × 4096) ÷ (600 − 1) = 1.57

Vertical scaling ratio = round (3277 − (1.57)) = 3275

(4) Setting Frame Buffer Access Area

In the frame buffer in which graphics data is stored, graphics data needs to be expanded in the area of 640 × 480 pixels or larger.

Here, the frame buffer area in which graphics data is expanded is assumed to be 640 × 480 pixels.

Since the frame buffer is accessed in 64-bit units, RGB565 (16 bits) is accessed in 4-pixel units.

The line offset address values to be set are:

GR_LN_OFF[14:0] = 640 × 2 = 1280

The frame offset address values to be set are:

GR_FLM_OFF[22:0] = GR_LN_OFF[14:0] × 480 = 614400

chanhvinh.com

(5) Register Setting Example

Table 31.45 Register Setting Example for Scaled-up Graphics Display

Register Name	Bit Name	Settings	Remarks
Synchronization Control			
SC_SCL0_FRC3	SC_RES_VS_SEL	1	Free-running Vsync selected (when the appropriate input signal is available, an external sync can also be selected)
SC_SCL0_FRC4	SC_RES_FV[10:0]	668	Vertical period width of output signal (period width = set value + 1)
SC_SCL0_FRC4	SC_RES_FH[10:0]	1040	Horizontal period width of output signal (period width = set value + 1)
Size of Angle of View			
SC_SCL0_FRC6	SC_RES_F_VS[10:0]	27	Vertical valid start position of full screen
SC_SCL0_FRC6	SC_RES_F_VW[10:0]	600	Vertical valid width of full screen
SC_SCL0_FRC7	SC_RES_F_HS[10:0]	216	Horizontal valid start position of full screen
SC_SCL0_FRC7	SC_RES_F_HW[10:0]	800	Horizontal valid width of full screen
SC_SCL0_US2	SC_RES_P_VS[10:0]	27	Vertical valid start position of image output
SC_SCL0_US2	SC_RES_P_VW[10:0]	600	Vertical valid width of image output
SC_SCL0_US3	SC_RES_P_HS[10:0]	216	Horizontal valid start position of image output
SC_SCL0_US3	SC_RES_P_HW[10:0]	800	Horizontal valid width of image output
Scaling Setting			
SC_SCL0_US5	SC_RES_US_H_RATIO[15:0]	3276	Horizontal scaling-up because SC_RES_US_H_RATIO is smaller than 4096
SC_SCL0_DS6	SC_RES_V_RATIO[15:0]	3275	Vertical scaling-up because SC_RES_V_RATIO is smaller than 4096
SC_SCL0_US1	SC_RES_US_H_ON	1	Horizontal scaling-up on
SC_SCL0_US1	SC_RES_US_V_ON	1	Vertical scaling-up on
SC_SCL0_US4	SC_RES_IN_VW[10:0]	480	Vertical width of frame buffer read
SC_SCL0_US4	SC_RES_IN_HW[10:0]	640	Horizontal width of frame buffer read
Frame Buffer Read Setting			
GR_FLM1	GR_FLM_SEL[1:0]	1	Frame number setting with register
GR_FLM3	GR_FLM_NUM[9:0]	0	Frame number of frame buffer (0 in setting example)
GR_FLM2	GR_BASE[31:0]	0	Conforming to graphics expansion setting (0 in setting example)
GR_FLM3	GR_LN_OFF[14:0]	1280	Conforming to graphics expansion setting
GR_FLM4	GR_FLM_OFF[22:0]	614400	Conforming to graphics expansion setting
GR_FLM6	GR_FORMAT[3:0]	0	Frame buffer read format RGB565
GR_FLM_RD	GR_R_ENB	1	Frame buffer read enabled
Scaling-up Selection			
SC_SCL0_US8	SC_RES_IBUS_SYNC_SEL	0	Scaled-up video signal displayed
GR_AB1	GR_DISP_SEL[1:0]	1	Scaling display selected

chanhvinh.com

32. Video Display Controller 5 (4): Image Quality Improver

32.1 Image Quality Improver

32.1.1 Overview of Functions

The image quality improver subjects scaled YCbCr signals to black stretching, LTI/sharpness processing, and GBR conversion by using a color matrix.

The image quality improver does not act on RGB signals.

Figure 32.1 is a functional block diagram of the image quality improver. Image quality improver 0 is connected to scaler 0.

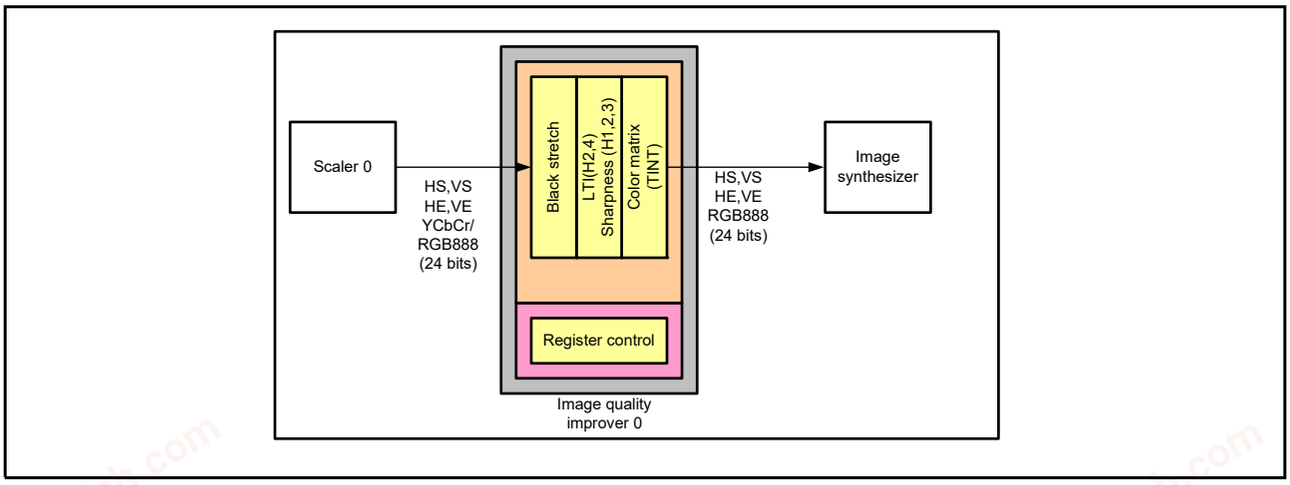


Figure 32.1 Functional Block Diagram of Image Quality Improver

32.1.2 Register Update Control

The control register for image quality improver controls the update timing entirely by vertical synchronous signals.

The vertical synchronous signal launched after the update control register is set to 1 is reflected in various registers, following which the update control register is automatically cleared to 0.

Note that registers for the improver can be identified by the number in the register name like ADJ0_xxxx. In the sections except for Register Description, however, the number is omitted like ADJ_xxxx for convenience sake.

Table 32.1 Register Update Control

Register Name	Bit Name	Initial Value	Description
ADJ_UPDATE	ADJ_VEN	0	Image Quality Improver Register Update 0: Register is not updated. 1: Register is updated by launch of vertical synchronous signal.

chanhvinh.com

32.1.3 Black Stretch

Black stretch refers to the black stretch correction of the Y signal of the input video signal of YCbCr format.

Correction of the Y signal is done by adjusting the time constant, depth (gain), and start point.

Figure 32.2 is a drawing illustrating black stretch correction.

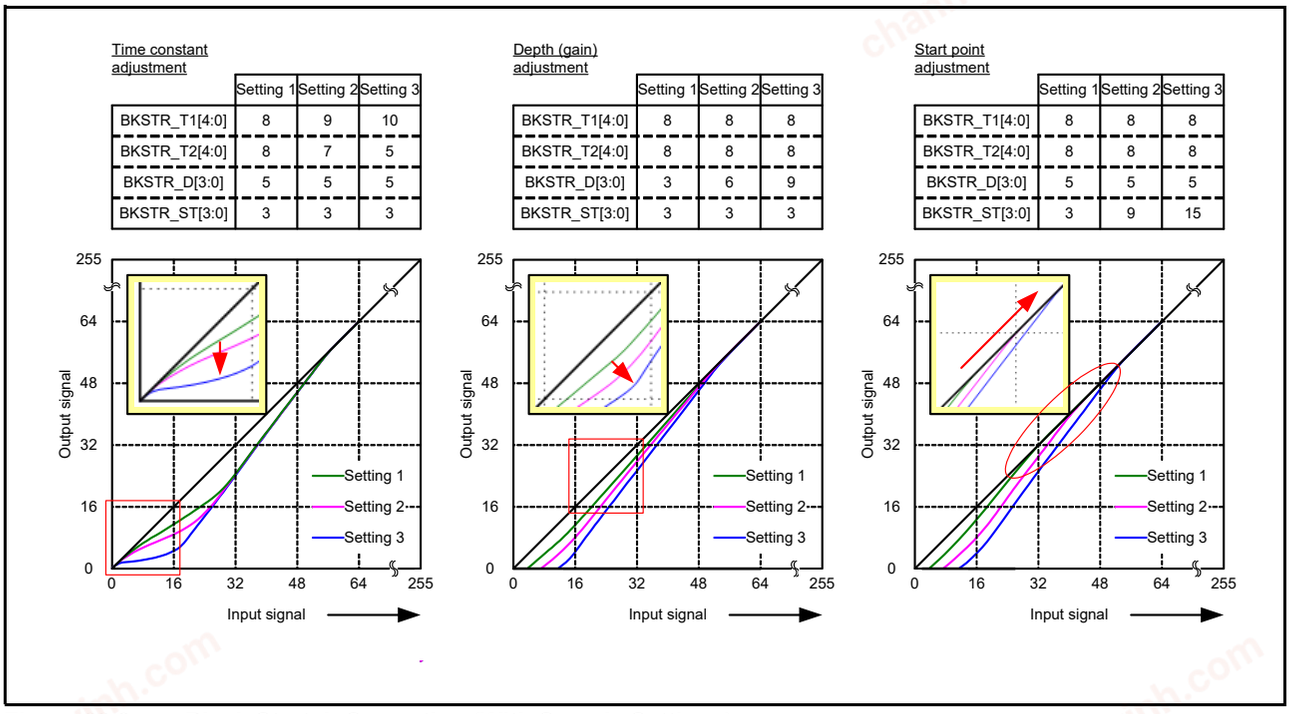


Figure 32.2 Black Stretch Correction (With Sample Settings)

Table 32.2 Black Stretch Control

Register Name	Bit Name	Initial Value	Description
ADJ_BKSTR_SET	BKSTR_ON	0	Black Stretch On/Off Control 0: Black Stretch Off 1: Black Stretch On
ADJ_BKSTR_SET	BKSTR_ST[3:0]	0	Black Stretch Start Point 0 (low) to 15 (high)
ADJ_BKSTR_SET	BKSTR_T1[4:0]	0	Black Stretch Time Constant (T1) 0 (small) to 31 (large)
ADJ_BKSTR_SET	BKSTR_T2[4:0]	0	Black Stretch Time Constant (T2) 0 (small) to 30 (large), 31: Setting prohibited
ADJ_BKSTR_SET	BKSTR_D[3:0]	0	Black Stretch Depth 0 (shallow) to 15 (deep)

chanhvinh.com

32.1.4 Enhancer

The enhancer subjects the scaled Y signal input to transient improvement (LTI) and sharpness processing in the horizontal direction.

(1) Enhancer Area Specification

The operating area of the enhancer is specified with reference to the rising edges of the Hsync signal and Vsync signal. ENH_HS should be set to four or greater clocks, and ENH_VS should be set to two or greater lines. Figure 32.3 shows enhancer area setting.

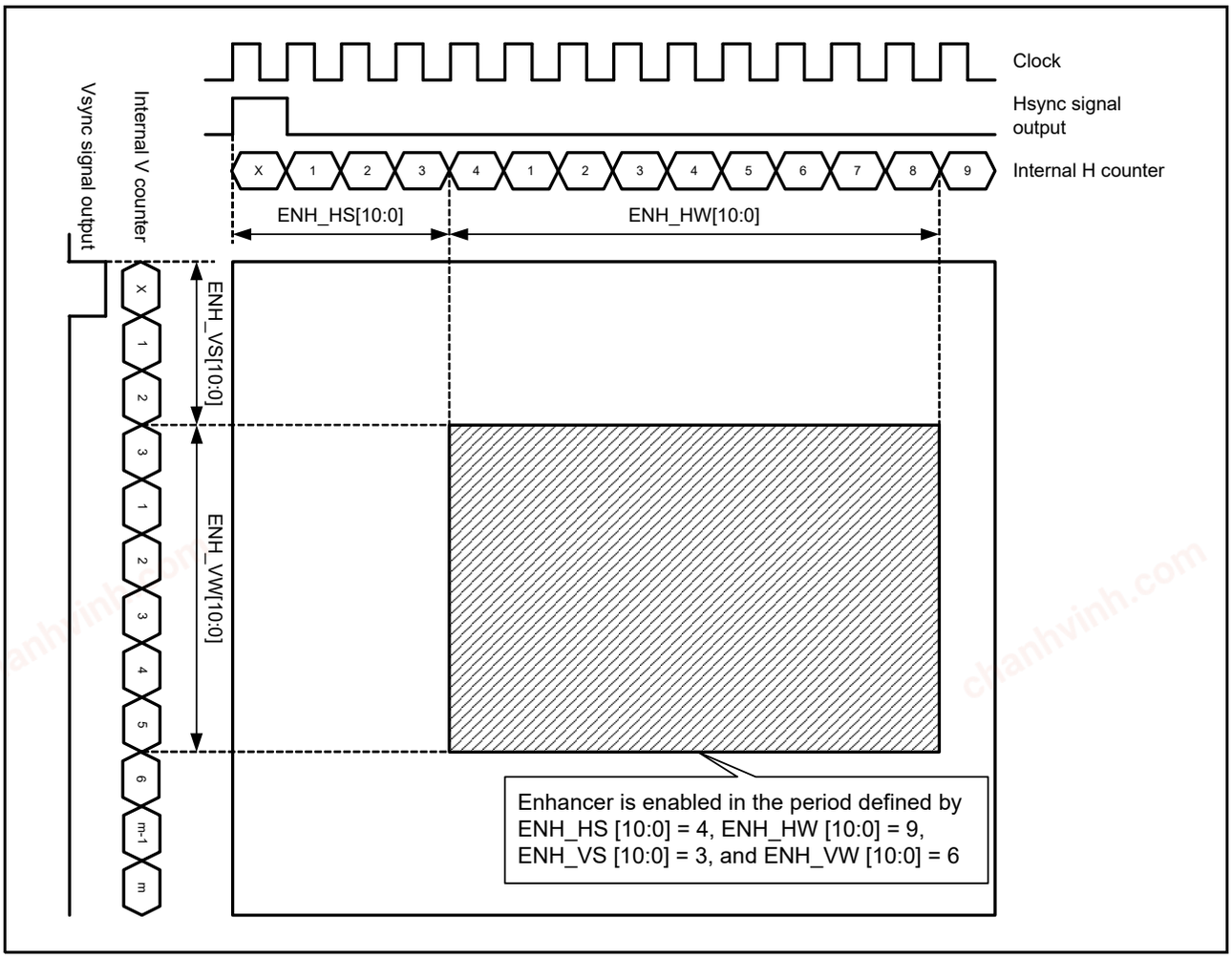


Figure 32.3 Period when Enhancer is Enabled

Setting ENH_DISP_ON to 1 displays the enhancer-enabled area with frame lines.

Table 32.3 Enhancer Area Control

Register Name	Bit Name	Initial Value	Description
ADJ_ENH_TIM1	ENH_MD	1	Operating Mode 0: RGB mode 1: YCbCr mode
ADJ_ENH_TIM2	ENH_VS[10:0]	0	Start Position of Vertical Valid Image Area in Enhancer-Enabled Area Note: Set to 2 or greater lines.
ADJ_ENH_TIM2	ENH_VW[10:0]	0	Width of Vertical Valid Image Area in Enhancer-Enabled Area

chanhvinh.com

Table 32.3 Enhancer Area Control

Register Name	Bit Name	Initial Value	Description
ADJ_ENH_TIM3	ENH_HS[10:0]	0	Start Position of Horizontal Valid Image Area in Enhancer-Enabled Area Note: Set to 4 or greater clocks.
ADJ_ENH_TIM3	ENH_HW[10:0]	0	Width of Horizontal Valid Image Area in Enhancer-Enabled Area
ADJ_ENH_TIM1	ENH_DISP_ON	0	Frame Line Display in Enhancer-Enabled Area 0: Off 1: On

chanhvinh.com

(2) LTI (Luminance Transient Improvement)

The enhancer subjects the Y signal input to transient improvement in the horizontal direction.

Transient improvement of the blanking signal is turned off.

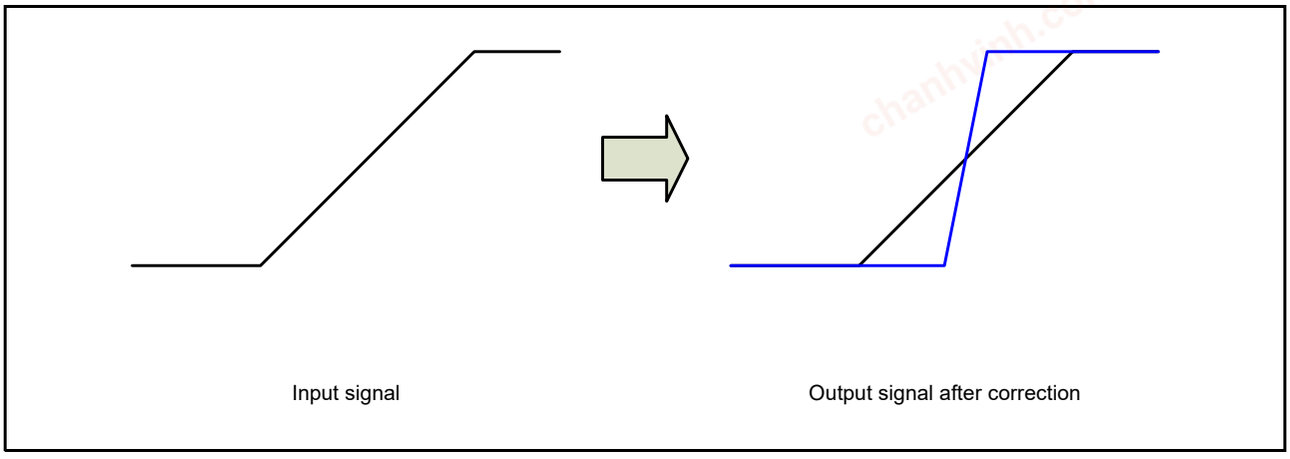


Figure 32.4 LTI Correction

After edge detection of the image, the LTI can be independently controlled in the two horizontal bands.

In LTI, the median filter is inserted after edge detection of the image.

In LTI (H4), the reference pixels of the median filter can be selected.

However, under normal operations, half the tap data (second adjacent pixel) at edge detection is used as reference.

Table 32.4 Reference Pixel Table for LTI

LTI Band	Reference Pixel for Edge Detection	LPF Application	Median Filter Reference Pixels
Horizontal LTI (H2)	Second adjacent pixel used as reference	LPF not applied or LPF (1,2,1)	Adjacent pixel used as reference
Horizontal LTI (H4)	Fourth adjacent pixel used as reference	LPF (1,2,1)	Adjacent pixel or second adjacent pixel used as reference

In LTI, the detection result can be subjected to a coring process.

The core value set in the register is subtracted from the edge detection result, and LTI correction is performed on the coring output after subtraction.

chanhvinh.com

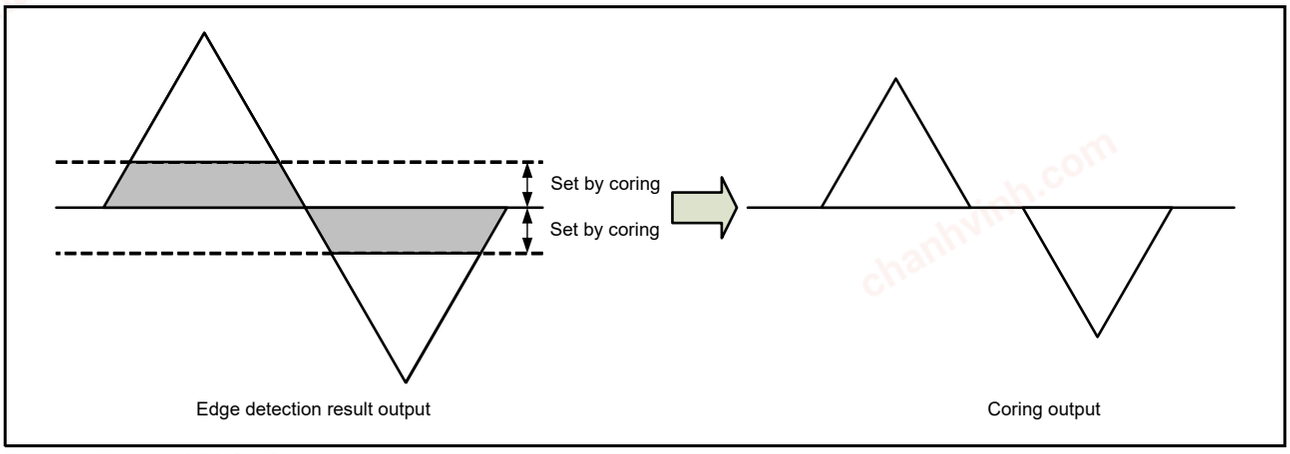


Figure 32.5 LTI Coring

Table 32.5 LTI Control

Register Name	Bit Name	Initial Value	Description
ADJ_ENH_LTI1	LTI_H_ON	0	LTI On/Off Control 0: LTI off 1: LTI on
ADJ_ENH_LTI1	LTI_H2_INC_ZERO [7:0]	10	Median Filter LTI Correction Threshold LTI correction is disabled when: |right TAP value – center TAP value| < LTI1_H2_INC_ZERO or |left TAP value – center TAP value| < LTI1_H2_INC_ZERO.
ADJ_ENH_LTI1	LTI_H2_LPF_SEL	0	LPF Selection for Folding Prevention Before H2 Edge Detection 0: LPF not selected 1: LPF selected
ADJ_ENH_LTI1	LTI_H2_GAIN[7:0]	0	LTI Edge Amplitude Value Gain 0 (0 times) to 64 (+1 times) to 255 (+ approx. 4 times)
ADJ_ENH_LTI1	LTI_H2_CORE[7:0]	0	LTI Coring (Maximum core value of 255) Amplitude smaller than or equal to the value of LTI_H2_CORE is cored from the edge amplitude value. (A core value setting of 128 remains unchanged.)
ADJ_ENH_LTI2	LTI_H4_INC_ZERO[7:0]	10	Median Filter LTI Correction Threshold LTI correction is disabled when: |right TAP value – center TAP value| < LTI1_H4_INC_ZERO or |left TAP value – center TAP value| < LTI1_H4_INC_ZERO.
ADJ_ENH_LTI2	LTI_H4_MEDIAN_TAP_ SEL	0	Median Filter Reference Pixel Select 0: Second adjacent pixel selected as reference 1: Adjacent pixel selected as reference
ADJ_ENH_LTI2	LTI_H4_GAIN[7:0]	0	LTI Edge Amplitude Value Gain 0 (0 times) to 64 (+1 times) to 255 (+ approx. 4 times)
ADJ_ENH_LTI2	LTI_H4_CORE[7:0]	0	LTI Coring (Maximum core value of 255) Amplitude less than or equal to the value of LTI_H4_CORE is cored from the edge amplitude value. (A core value setting of 128 remains unchanged.)

chanhvinh.com

(3) Sharpness Process

The enhancer performs edge enhancement on the Y signal input by adding overshoot and undershoot to the original signal. Edge enhancement of the blanking signal is turned off.

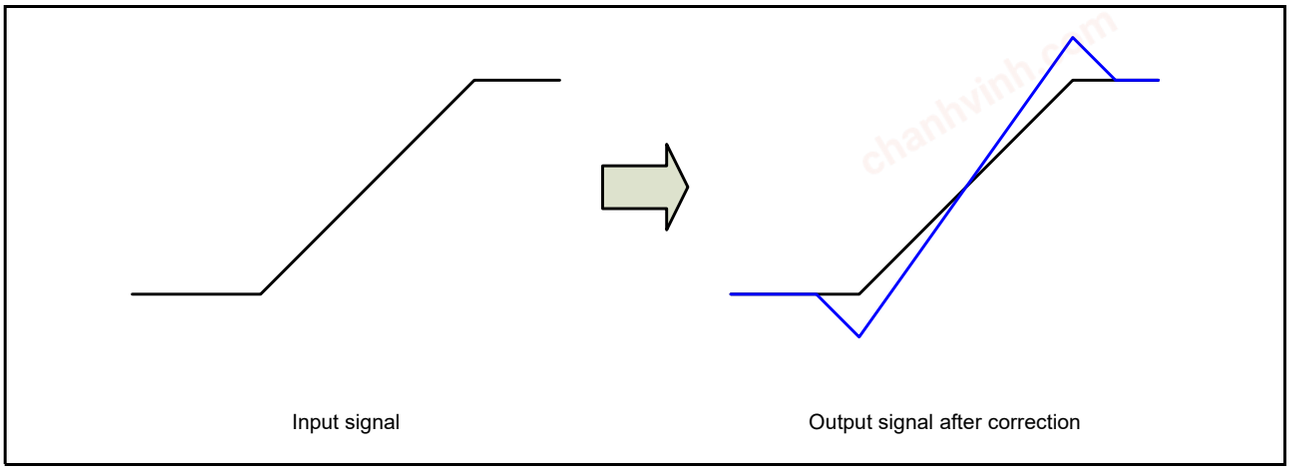


Figure 32.6 Sharpness Correction

After edge detection of the image, the sharpness can be independently controlled in the three horizontal bands.

In horizontal sharpness, a 3-tap low-pass filter (LPF) is inserted before edge detection to prevent folding. The LPF can be turned on or off by register setting.

Table 32.6 Reference Pixel Table for Sharpness

Sharpness Band	Reference Pixel for Edge Detection	LPF Application
Horizontal sharpness (H1)	Adjacent pixel used as reference	LPF not applied
Horizontal sharpness (H2)	Second adjacent pixel used as reference	LPF not applied or LPF (1,2,1)
Horizontal sharpness (H3)	Third adjacent pixel used as reference	LPF (1,2,1)

The edge amplitude of the edge to be enhanced is adjusted according to the value of SHP_CORE.

Edge enhancement is accomplished when the edge detection result of the image is greater than the value of SHP_CORE.

In edge enhancement, a correction value is output by multiplying (edge amplitude value - SHP_CORE) by sharpness gain.

Sharpness is turned off when the edge detection result of the image is smaller than the value of SHP_CORE.

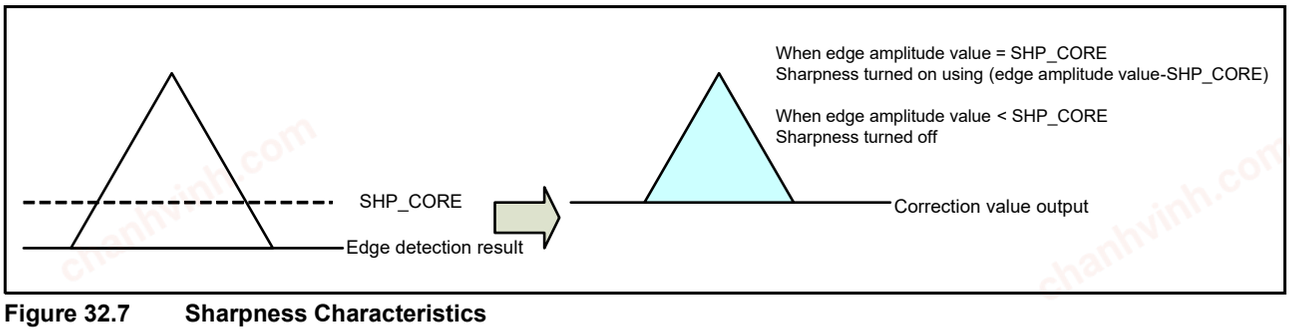


Figure 32.7 Sharpness Characteristics

chanhvinh.com

Table 32.7 Sharpness Control

Register Name	Bit Name	Initial Value	Description
ADJ_ENH_SHP1	SHP_H_ON	0	Sharpness On/Off Control 0: Horizontal sharpness off 1: Horizontal sharpness on
ADJ_ENH_SHP3	SHP_H2_LPF_SEL	0	LPF Selection for Folding Prevention Before H2 Edge Detection 0: LPF not selected 1: LPF selected
ADJ_ENH_SHP2	SHP_H1_CLIP_O[7:0]	0	Sharpness Correction Value Clipping (on the Overshoot Side) Correction value clipped according to SHP_H1_CLIP_O
ADJ_ENH_SHP2	SHP_H1_CLIP_U[7:0]	0	Sharpness Correction Value Clipping (on the Undershoot Side) Correction value clipped according to SHP_H1_CLIP_U
ADJ_ENH_SHP2	SHP_H1_GAIN_O[7:0]	0	Sharpness Edge Amplitude Value Gain (on the Overshoot Side) 0 (0 times) to 64 (+1 times) to 255 (+ approx. 4 times) Sharpness correction value = SHP_H1_GAIN_O × (edge amplitude value – SHP_H1_CORE)
ADJ_ENH_SHP2	SHP_H1_GAIN_U[7:0]	0	Sharpness Edge Amplitude Value Gain (on the Undershoot Side) 0 (0 times) to 64 (+1 times) to 255 (+ approx. 4 times) Sharpness correction value = SHP_H1_GAIN_U × (edge amplitude value – SHP_H1_CORE)
ADJ_ENH_SHP1	SHP_H1_CORE[6:0]	0	Active Sharpness Range Edge amplitude value ≥ SHP_H1_CORE: Sharpness processing on Edge amplitude value < SHP_H1_CORE: Sharpness processing off Sharpness processing is always on when the edge detection value is 128 or greater.
ADJ_ENH_SHP4	SHP_H2_CLIP_O[7:0]	0	Sharpness Correction Value Clipping (on the Overshoot Side) Correction value clipped according to SHP_H2_CLIP_O
ADJ_ENH_SHP4	SHP_H2_CLIP_U[7:0]	0	Sharpness Correction Value Clipping (on the Undershoot Side) Correction value clipped according to SHP_H2_CLIP_U
ADJ_ENH_SHP4	SHP_H2_GAIN_O[7:0]	0	Sharpness Edge Amplitude Value Gain (on the Overshoot Side) 0 (0 times) to 64 (+1 times) to 255 (+ approx. 4 times) Sharpness correction value = SHP_H2_GAIN_O × (edge amplitude value – SHP_H2_CORE)
ADJ_ENH_SHP4	SHP_H2_GAIN_U[7:0]	0	Sharpness Edge Amplitude Value Gain (on the Undershoot Side) 0 (0 times) to 64 (+1 times) to 255 (+ approx. 4 times) Sharpness correction value = SHP_H2_GAIN_U × (edge amplitude value – SHP_H2_CORE)
ADJ_ENH_SHP3	SHP_H2_CORE[6:0]	0	Active Sharpness Range Edge amplitude value ≥ SHP_H2_CORE: Sharpness processing on Edge amplitude value < SHP_H2_CORE: Sharpness processing off Sharpness processing is always on when the edge detection value is 128 or greater.
ADJ_ENH_SHP6	SHP_H3_CLIP_O[7:0]	0	Sharpness Correction Value Clipping (on the Overshoot Side) Correction value clipped according to SHP_H3_CLIP_O
ADJ_ENH_SHP6	SHP_H3_CLIP_U[7:0]	0	Sharpness Correction Value Clipping (on the Undershoot Side) Correction value clipped according to SHP_H3_CLIP_U
ADJ_ENH_SHP6	SHP_H3_GAIN_O[7:0]	0	Sharpness Edge Amplitude Value Gain (on the Overshoot Side) 0 (0 times) to 64 (+1 times) to 255 (+ approx. 4 times) Sharpness correction value = SHP_H3_GAIN_O × (edge amplitude value – SHP_H3_CORE)
ADJ_ENH_SHP6	SHP_H3_GAIN_U[7:0]	0	Sharpness Edge Amplitude Value Gain (on the Undershoot Side) 0 (0 times) to 64 (+1 times) to 255 (+ approx. 4 times) Sharpness correction value = SHP_H3_GAIN_U × (edge amplitude value – SHP_H3_CORE)
ADJ_ENH_SHP5	SHP_H3_CORE[6:0]	0	Active Sharpness Range Edge amplitude value ≥ SHP_H3_CORE: Sharpness processing on Edge amplitude value < SHP_H3_CORE: Sharpness processing off Sharpness processing is always on when the edge detection value is 128 or greater.

chanhvinh.com

32.1.5 Color Matrix

Color matrix is performed by adjusting the offset of each input signal and nine-axis gain. This allows YCbCr to GBR conversion.

(1) GBR to GBR Conversion

YGIN_A = YGIN + ADJ_MTX_YG − 128

CBBIN_A = CBBIN + ADJ_MTX_B − 128

CRRIN_A = CRRIN + ADJ_MTX_R − 128

YGOUT = (ADJ_MTX_GG × YGIN_A + ADJ_MTX_GB × CBBIN_A + ADJ_MTX_GR × CRRIN_A) ÷ 256

CBBOUT = (ADJ_MTX_BG × YGIN_A + ADJ_MTX_BB × CBBIN_A + ADJ_MTX_BR × CRRIN_A) ÷ 256

CRROUT = (ADJ_MTX_RG × YGIN_A + ADJ_MTX_RB × CBBIN_A + ADJ_MTX_RR × CRRIN_A) ÷ 256

(2) YCbCr to GBR Conversion

YGIN_A = YGIN + ADJ_MTX_YG − 128

CBBIN_A = CBBIN − 128

CRRIN_A = CRRIN − 128

YGOUT = (ADJ_MTX_GG × YGIN_A + ADJ_MTX_GB × CBBIN_A + ADJ_MTX_GR × CRRIN_A) ÷ 256

CBBOUT = (ADJ_MTX_BG × YGIN_A + ADJ_MTX_BB × CBBIN_A + ADJ_MTX_BR × CRRIN_A) ÷ 256

CRROUT = (ADJ_MTX_RG × YGIN_A + ADJ_MTX_RB × CBBIN_A + ADJ_MTX_RR × CRRIN_A) ÷ 256

Table 32.8 Matrix Coefficients (Standard Values) of SMPTE 293M

	YGIN		CBBIN		CRRIN	
	Coefficient	**Bit Setting**	**Coefficient**	**Bit Setting**	**Coefficient**	**Bit Setting**
YGOUT	1.000	ADJ_MTX_GG = 256	–0.344	ADJ_MTX_GB = 1960	–0.714	ADJ_MTX_GR = 1865
CBBOUT	1.000	ADJ_MTX_BG = 256	1.772	ADJ_MTX_BB = 454	0.000	ADJ_MTX_BR = 0
CRROUT	1.000	ADJ_MTX_RG = 256	0.000	ADJ_MTX_RB = 0	1.402	ADJ_MTX_RR = 359

chanhvinh.com

Table 32.9 Color Matrix Control

Register Name	Bit Name	Initial Value	Description
ADJ_MTX_MODE	ADJ_MTX_MD[1:0]	2	Operating Mode 0: GBR → GBR 1: Setting prohibited 2: YCbCr → GBR 3: Setting prohibited
ADJ_MTX_YG_ADJ0	ADJ_MTX_YG[7:0]	128	Y/G Signal Offset (DC) Adjustment Unsigned (0 (-128) to 128(0) to 255 (+127) [LSB])
ADJ_MTX_CBB_ADJ0	ADJ_MTX_B[7:0]	128	B Signal Offset (DC) Adjustment Unsigned (0 (-128) to 128 (0) to 255 (+127) [LSB])
ADJ_MTX_CRR_ADJ0	ADJ_MTX_R[7:0]	128	R Signal Offset (DC) Adjustment Unsigned (0 (-128) to 128 (0) to 255 (+127) [LSB])
ADJ_MTX_YG_ADJ0	ADJ_MTX_GG[10:0]	256	Gain Adjustment of Y/G Signal of G Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
ADJ_MTX_YG_ADJ1	ADJ_MTX_GB[10:0]	1960	Gain Adjustment of Cb/B Signal of G Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
ADJ_MTX_YG_ADJ1	ADJ_MTX_GR[10:0]	1865	Gain Adjustment of Cr/R Signal of G Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
ADJ_MTX_CBB_ADJ0	ADJ_MTX_BG[10:0]	256	Gain Adjustment of Y/G Signal of B Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
ADJ_MTX_CBB_ADJ1	ADJ_MTX_BB[10:0]	454	Gain adjustment of Cb/B signal of B Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
ADJ_MTX_CBB_ADJ1	ADJ_MTX_BR[10:0]	0	Gain Adjustment of Cr/R Signal of B Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
ADJ_MTX_CRR_ADJ0	ADJ_MTX_RG[10:0]	256	Gain Adjustment of Y/G Signal of R Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
ADJ_MTX_CRR_ADJ1	ADJ_MTX_RB[10:0]	0	Gain Adjustment of Cb/B Signal of R Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
ADJ_MTX_CRR_ADJ1	ADJ_MTX_RR[10:0]	359	Gain Adjustment of Cr/R Signal of R Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

chanhvinh.com

32.2 Register Description

Table 32.10 shows the register configuration.

[Symbols used in Register Description]

Initial value : Register value after a reset
— : Undefined value
R/W : Readable/writable. The written value can be read.
R/WC0 : Read and write. Bit is initialized if 0 is written, and ignored if 1 is written.
R/WC1 : Read and write. Bit is initialized if 1 is written, and ignored if 0 is written.
R : Read-only. The write value should always be 0.
—/W : Write-only. Read value is undefined.

Table 32.10 Image Quality Improver Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Register update control register in image quality improver (image quality improver 0)	ADJ0_UPDATE	R/WC1	H'0000 0000	H'FCFF 7680	32
Black stretch register (image quality improver 0)	ADJ0_BKSTR_SET	R/W	H'0000 0000	H'FCFF 7684	32
Enhancer timing adjustment register 1 (image quality improver 0)	ADJ0_ENH_TIM1	R/W	H'0000 0010	H'FCFF 7688	32
Enhancer timing adjustment register 2 (image quality improver 0)	ADJ0_ENH_TIM2	R/W	H'0023 01E0	H'FCFF 768C	32
Enhancer timing adjustment register 3 (image quality improver 0)	ADJ0_ENH_TIM3	R/W	H'0091 0280	H'FCFF 7690	32
Enhancer sharpness register 1 (image quality improver 0)	ADJ0_ENH_SHP1	R/W	H'0000 0000	H'FCFF 7694	32
Enhancer sharpness register 2 (image quality improver 0)	ADJ0_ENH_SHP2	R/W	H'0000 0000	H'FCFF 7698	32
Enhancer sharpness register 3 (image quality improver 0)	ADJ0_ENH_SHP3	R/W	H'0000 0000	H'FCFF 769C	32
Enhancer sharpness register 4 (image quality improver 0)	ADJ0_ENH_SHP4	R/W	H'0000 0000	H'FCFF 76A0	32
Enhancer sharpness register 5 (image quality improver 0)	ADJ0_ENH_SHP5	R/W	H'0000 0000	H'FCFF 76A4	32
Enhancer sharpness register 6 (image quality improver 0)	ADJ0_ENH_SHP6	R/W	H'0000 0000	H'FCFF 76A8	32
Enhancer LTI register 1 (image quality improver 0)	ADJ0_ENH_LTI1	R/W	H'000A 0000	H'FCFF 76AC	32
Enhancer LTI register 2 (image quality improver 0)	ADJ0_ENH_LTI2	R/W	H'000A 0000	H'FCFF 76B0	32
Matrix mode register in image quality improver (image quality improver 0)	ADJ0_MTX_MODE	R/W	H'0000 0002	H'FCFF 76B4	32
Matrix YG control register 0 in image quality improver (image quality improver 0)	ADJ0_MTX_YG_ADJ0	R/W	H'0080 0100	H'FCFF 76B8	32
Matrix YG control register 1 in image quality improver (image quality improver 0)	ADJ0_MTX_YG_ADJ1	R/W	H'07A8 0749	H'FCFF 76BC	32
Matrix CBB control register 0 in image quality improver (image quality improver 0)	ADJ0_MTX_CBB_ADJ0	R/W	H'0080 0100	H'FCFF 76C0	32
Matrix CBB control register 1 in image quality improver (image quality improver 0)	ADJ0_MTX_CBB_ADJ1	R/W	H'01C6 0000	H'FCFF 76C4	32

chanhvinh.com

Table 32.10 Image Quality Improver Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Matrix CRR control register 0 in image quality improver (image quality improver 0)	ADJ0_MTX_CRR_ADJ0	R/W	H'0080 0100	H'FCFF 76C8	32
Matrix CRR control register 1 in image quality improver (image quality improver 0)	ADJ0_MTX_CRR_ADJ1	R/W	H'0000 0167	H'FCFF 76CC	32

32.2.1 Register Update Control Register in Image Quality Improver (ADJ0_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADJ0_VEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ADJ0_VEN	0	R/WC1	Image Quality Improver Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

chanhvinh.com

32.2.2 Black Stretch Register (ADJ0_BKSTR_SET)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	BKSTR_ON	BKSTR_ST[3:0]				BKSTR_D[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	BKSTR_T1[4:0]					—	—	—	BKSTR_T2[4:0]				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	BKSTR_ON	0	R/W	Black Stretch On/Off Control 0: Black stretch off 1: Black stretch on
23 to 20	BKSTR_ST [3:0]	0	R/W	Black Stretch Start Point Setting values: 0 (low) to 15 (high)
19 to 16	BKSTR_D [3:0]	0	R/W	Depth of Black Stretch Setting Values: 0 (shallow) to 15 (deep)
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12 to 8	BKSTR_T1 [4:0]	0	R/W	Black Stretch Time Constant (T1) Setting Values: 0 (small) to 31 (large)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4 to 0	BKSTR_T2 [4:0]	0	R/W	Black Stretch Time Constant (T2) Setting Values: 0 (small) to 31 (large)

Note: This register is updated when ADJ0_VEN in ADJ0_UPDATE is 1.

chanhvinh.com

32.2.3 Enhancer Timing Adjustment Register 1 (ADJ0_ENH_TIM1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	ENH_MD	—	—	—	ENH_DISP_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	ENH_MD	1	R/W	Operating Mode 0: RGB mode 1: YCbCr mode
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ENH_DISP_ON	0	R/W	Frame Line Display On/Off of Enhancer-Enabled Area 0: Display off 1: Display on

Note: This register is updated when ADJ0_VEN in ADJ0_UPDATE is 1.

32.2.4 Enhancer Timing Adjustment Register 2 (ADJ0_ENH_TIM2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	ENH_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ENH_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	ENH_VS[10:0]	35	R/W	Start Position of Vertical Valid Image Area in Enhancer-Enabled Area Note: Set to 2 or greater lines.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	ENH_VW[10:0]	480	R/W	Width of Vertical Valid Image Area in Enhancer-Enabled Area

Note: This register is updated when ADJ0_VEN in ADJ0_UPDATE is 1.

chanhvinh.com

32.2.5 Enhancer Timing Adjustment Register 3 (ADJ0_ENH_TIM3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	ENH_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ENH_HW[10:0]										
Initial value:	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	ENH_HS[10:0]	145	R/W	Start Position of Horizontal Valid Image Area in Enhancer-Enabled Area Note: Set to 4 or greater clocks.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	ENH_HW[10:0]	640	R/W	Width of Horizontal Valid Image Area in Enhancer-Enabled Area

Note: This register is updated when ADJ0_VEN in ADJ0_UPDATE is 1.

32.2.6 Enhancer Sharpness Register 1 (ADJ0_ENH_SHP1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SHP_H_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	SHP_H1_CORE[6:0]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	SHP_H_ON	0	R/W	Sharpness On/Off Control 0: Horizontal sharpness off 1: Horizontal sharpness on
15 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6 to 0	SHP_H1_CORE[6:0]	0	R/W	Active Sharpness Range Edge amplitude value ≥ SHP_H1_CORE: Sharpness processing on Edge amplitude value < SHP_H1_CORE: Sharpness processing off Sharpness processing is always on when the edge detection value is 128 or greater.

Note: This register is updated when ADJ0_VEN in ADJ0_UPDATE is 1.

chanhvinh.com

32.2.7 Enhancer Sharpness Register 2 (ADJ0_ENH_SHP2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SHP_H1_CLIP_O[7:0]								SHP_H1_CLIP_U[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SHP_H1_GAIN_O[7:0]								SHP_H1_GAIN_U[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	SHP_H1_CLIP_O [7:0]	0	R/W	Sharpness Correction Value Clipping (on the Overshoot Side) Correction value clipped according to SHP_H1_CLIP_O
23 to 16	SHP_H1_CLIP_U [7:0]	0	R/W	Sharpness Correction Value Clipping (on the Undershoot Side) Correction value clipped according to SHP_H1_CLIP_U
15 to 8	SHP_H1_GAIN_O [7:0]	0	R/W	Sharpness Edge Amplitude Value Gain (on the Overshoot Side) 0 (0 times) to 64 (+1 times) to 255 (+approx. 4 times) Sharpness correction value = SHP_H1_GAIN_O × (edge amplitude value – SHP_H1_CORE)
7 to 0	SHP_H1_GAIN_U [7:0]	0	R/W	Sharpness Edge Amplitude Value Gain (on the Undershoot Side) 0 (0 times) to 64 (+1 times) to 255 (+approx. 4 times) Sharpness correction value = SHP_H1_GAIN_U × (Edge amplitude value – SHP_H1_CORE)

Note: This register is updated when ADJ0_VEN in ADJ0_UPDATE is 1.

chanhvinh.com

32.2.8 Enhancer Sharpness Register 3 (ADJ0_ENH_SHP3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SHP_H2_LPF_SEL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	SHP_H2_CORE[6:0]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	SHP_H2_LPF_SEL	0	R/W	LPF Selection for Folding Prevention before H2 Edge Detection 0: LPF not selected 1: LPF selected
15 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6 to 0	SHP_H2_CORE[6:0]	0	R/W	Active Sharpness Range Edge amplitude value ≥ SHP_H2_CORE: Sharpness processing on Edge amplitude value < SHP_H2_CORE: Sharpness processing off Sharpness processing is always on when the edge detection value is 128 or greater.

Note: This register is updated when ADJ0_VEN in ADJ0_UPDATE is 1.

chanhvinh.com

32.2.9 Enhancer Sharpness Register 4 (ADJ0_ENH_SHP4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SHP_H2_CLIP_O[7:0]								SHP_H2_CLIP_U[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SHP_H2_GAIN_O[7:0]								SHP_H2_GAIN_U[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	SHP_H2_CLIP_O[7:0]	0	R/W	Sharpness Correction Value Clipping (on the Overshoot Side) Correction value clipped according to SHP_H2_CLIP_O
23 to 16	SHP_H2_CLIP_U[7:0]	0	R/W	Sharpness Correction Value Clipping (on the Undershoot Side) Correction value clipped according to SHP_H2_CLIP_U
15 to 8	SHP_H2_GAIN_O[7:0]	0	R/W	Sharpness Edge Amplitude Value Gain (on the Overshoot Side) 0 (0 times) to 64 (+1 times) to 255 (+ approx. 4 times) Sharpness correction value = SHP_H2_GAIN_O × (edge amplitude value – SHP_H2_CORE)
7 to 0	SHP_H2_GAIN_U[7:0]	0	R/W	Sharpness Edge Amplitude Value Gain (on the Undershoot Side) 0 (0 times) to 64 (+1 times) to 255 (+ approx. 4 times) Sharpness correction value = SHP_H2_GAIN_U × (edge amplitude value – SHP_H2_CORE)

Note: This register is updated when ADJ0_VEN in ADJ0_UPDATE is 1.

32.2.10 Enhancer Sharpness Register 5 (ADJ0_ENH_SHP5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	SHP_H3_CORE[6:0]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6 to 0	SHP_H3_CORE[6:0]	0	R/W	Active Sharpness Range Edge amplitude value ≥ SHP_H3_CORE: Sharpness processing on Edge amplitude value < SHP_H3_CORE: Sharpness processing off Sharpness processing is always on when the edge detection value is 128 or greater.

Note: This register is updated when ADJ0_VEN in ADJ0_UPDATE is 1.

chanhvinh.com

32.2.11 Enhancer Sharpness Register 6 (ADJ0_ENH_SHP6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SHP_H3_CLIP_O[7:0]								SHP_H3_CLIP_U[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SHP_H3_GAIN_O[7:0]								SHP_H3_GAIN_U[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	SHP_H3_ CLIP_O[7:0]	0	R/W	Sharpness Correction Value Clipping (on the Overshoot Side) Correction value clipped according to SHP_H3_CLIP_O
23 to 16	SHP_H3_ CLIP_U[7:0]	0	R/W	Sharpness Correction Value Clipping (on the Undershoot Side) Correction value clipped according to SHP_H3_CLIP_U
15 to 8	SHP_H3_ GAIN_O[7:0]	0	R/W	Sharpness Edge Amplitude Value Gain (on the Overshoot Side) 0 (0 times) to 64 (+1 times) to 255 (+ approx. 4 times) Sharpness correction value = SHP_H3_GAIN_O × (Edge amplitude value – SHP_H3_CORE)
7 to 0	SHP_H3_ GAIN_U[7:0]	0	R/W	Sharpness Edge Amplitude Value Gain (on the Undershoot Side) 0 (0 times) to 64 (+1 times) to 255 (+ approx. 4 times) Sharpness correction value = SHP_H3_GAIN_U × (Edge amplitude value – SHP_H3_CORE)

Note: This register is updated when ADJ0_VEN in ADJ0_UPDATE is 1.

chanhvinh.com

32.2.12 Enhancer LTI Register 1 (ADJ0_ENH_LTI1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LTI_H_ON	—	—	—	—	—	—	LTI_H2_LPF_SEL	LTI_H2_INC_ZERO[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
R/W:	R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LTI_H2_GAIN[7:0]								LTI_H2_CORE[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	LTI_H_ON	0	R/W	LTI On/Off Control 0: LTI off 1: LTI on
30 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	LTI_H2_LPF_SEL	0	R/W	LPF Selection for Folding Prevention before H2 Edge Detection 0: LPF not selected 1: LPF selected
23 to 16	LTI_H2_INC_ZERO[7:0]	10	R/W	Median Filter LTI Correction Threshold LTI correction is disabled when | right TAP value – center TAP value | < LTI_H2_INC_ZERO or | left TAP value – center TAP value | < LTI_H2_INC_ZERO
15 to 8	LTI_H2_GAIN[7:0]	0	R/W	LTI Edge Amplitude Value Gain 0 (0 times) to 64 (+ 1 times) to 255 (+ approx. 4 times)
7 to 0	LTI_H2_CORE[7:0]	0	R/W	LTI Coring (Maximum Core value of 255) Amplitude less than or equal to the value of LTI_H2_CORE is cored from the edge amplitude value. (A core value setting of 128 remains unchanged)

Note: This register is updated when ADJ0_VEN in ADJ0_UPDATE is 1.

chanhvinh.com

32.2.13 Enhancer LTI Register 2 (ADJ0_ENH_LTI2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	LTI_H4_MEDIAN_TAP_SEL	LTI_H4_INC_ZERO[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LTI_H4_GAIN[7:0]								LTI_H4_CORE[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	LTI_H4_MEDIAN_TAP_SEL	0	R/W	Median Filter Reference Pixel Select 0: Second adjacent pixel selected as reference 1: Adjacent pixel selected as reference
23 to 16	LTI_H4_INC_ZERO[7:0]	10	R/W	Median Filter LTI Correction Threshold LTI correction is disabled when | right TAP value – center TAP value | < LTI_H4_INC_ZERO or | left TAP value – center TAP value | < LTI_H4_INC_ZERO
15 to 8	LTI_H4_GAIN[7:0]	0	R/W	LTI Edge Amplitude Value Gain 0 (0 times) to 64 (+ 1 times) to 255 (+ approx. 4 times)
7 to 0	LTI_H4_CORE[7:0]	0	R/W	LTI Coring (Maximum Core value of 255) Amplitude less than or equal to the value of LTI_H4_CORE is cored from the edge amplitude value (A core value setting of 128 remains unchanged)

Note: This register is updated when ADJ0_VEN in ADJ0_UPDATE is 1.

chanhvinh.com

32.2.14 Matrix Mode Register in Image Quality Improver (ADJ0_MTX_MODE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADJ0_MTX_MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	ADJ0_MTX_MD[1:0]	2	R/W	Operating Mode 0: GBR → GBR 1: Setting prohibited 2: YCbCr → GBR 3: Setting prohibited

Note: This register is updated when ADJ0_VEN in ADJ0_UPDATE is 1.

32.2.15 Matrix YG Control Register 0 in Image Quality Improver (ADJ0_MTX_YG_ADJ0)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	ADJ0_MTX_YG[7:0]							
Initial value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ADJ0_MTX_GG[10:0]										
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	ADJ0_MTX_YG[7:0]	128	R/W	Y/G Signal Offset (DC) Adjustment Unsigned (0 (-128) to 128 (0) to 255 (+127) [LSB])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	ADJ0_MTX_GG[10:0]	256	R/W	Gain Adjustment of Y/G Signal of G Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when ADJ0_VEN in ADJ0_UPDATE is 1.

chanhvinh.com

32.2.16 Matrix YG Control Register 1 in Image Quality Improver (ADJ0_MTX_YG_ADJ1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	ADJ0_MTX_GB[10:0]										
Initial value:	0	0	0	0	0	1	1	1	1	0	1	0	1	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ADJ0_MTX_GR[10:0]										
Initial value:	0	0	0	0	0	1	1	1	0	1	0	0	1	0	0	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	ADJ0_MTX_GB [10:0]	1960	R/W	Gain Adjustment of Cb/B Signal of G Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	ADJ0_MTX_GR [10:0]	1865	R/W	Gain Adjustment of Cr/R Signal of G Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when ADJ0_VEN in ADJ0_UPDATE is 1.

chanhvinh.com

32.2.17 Matrix CBB Control Register 0 in Image Quality Improver (ADJ0_MTX_CBB_ADJ0)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	ADJ0_MTX_B[7:0]							
Initial value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ADJ0_MTX_BG[10:0]										
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	ADJ0_MTX_B [7:0]	128	R/W	B Signal Offset (DC) Adjustment Unsigned (0 (-128) to 128 (0) to 255 (+127) [LSB])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	ADJ0_MTX_BG [10:0]	256	R/W	Gain Adjustment of Y/G Signal of B Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when ADJ0_VEN in ADJ0_UPDATE is 1.

chanhvinh.com

32.2.18 Matrix CBB Control Register 1 in Image Quality Improver (ADJ0_MTX_CBB_ADJ1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	ADJ0_MTX_BB[10:0]										
Initial value:	0	0	0	0	0	0	0	1	1	1	0	0	0	1	1	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ADJ0_MTX_BR[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	ADJ0_MTX_BB [10:0]	454	R/W	Gain Adjustment of Cb/B Signal of B Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	ADJ0_MTX_BR [10:0]	0	R/W	Gain Adjustment of Cr/R Signal of B Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when ADJ0_VEN in ADJ0_UPDATE is 1.

chanhvinh.com

32.2.19 Matrix CRR Control Register 0 in Image Quality Improver (ADJ0_MTX_CRR_ADJ0)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	ADJ0_MTX_R[7:0]							
Initial value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ADJ0_MTX_RG[10:0]										
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	ADJ0_MTX_R [7:0]	128	R/W	R Signal Offset (DC) Adjustment Unsigned (0 (-128) to 128 (0) to 255 (+127) [LSB])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	ADJ0_MTX_RG [10:0]	256	R/W	Gain Adjustment of Y/G Signal of R Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when ADJ0_VEN in ADJ0_UPDATE is 1.

chanhvinh.com

32.2.20 Matrix CRR Control Register 1 in Image Quality Improver (ADJ0_MTX_CRR_ADJ1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	ADJ0_MTX_RB[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ADJ0_MTX_RR[10:0]										
Initial value:	0	0	0	0	0	0	0	1	0	1	1	0	0	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	ADJ0_MTX_RB [10:0]	0	R/W	Gain Adjustment of Cb/B Signal of R Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	ADJ0_MTX_RR [10:0]	359	R/W	Gain Adjustment of Cr/R Signal of R Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when ADJ0_VEN in ADJ0_UPDATE is 1.

chanhvinh.com

32.3 Usage Method

32.3.1 Black Stretch Usage Method

The degree of black stretch can be adjusted by setting the depth (BKSTR_D[3:0]) and the start point (BKSTR_ST[3:0]) of the black stretch. The variation in the black stretch time axis can be adjusted by setting the time constant (BKSTR_T1[4:0] and BKSTR_T2[4:0]). By setting the time constant, changes that occur abruptly due to swapping of the scene can be controlled.

Table 32.11 Black Stretch Setting Register

Register Name	Bit Name	Set Value
ADJ_BKSTR_SET	BKSTR_ON	When black stretch is on: 1
ADJ_BKSTR_SET	BKSTR_D[3:0]	Set the depth of black stretch. The depth increases as the value becomes larger.
ADJ_BKSTR_SET	BKSTR_ST[3:0]	Set the start point of black stretch. The stretching area becomes larger as the value becomes larger.
ADJ_BKSTR_SET	BKSTR_T1[4:0]	Set the time constant of black stretch in the positive direction. The changes are more delayed as the value becomes larger.
ADJ_BKSTR_SET	BKSTR_T2[4:0]	Set the time constant of black stretch in the negative direction. The changes are more delayed as the value becomes larger.

Note: ADJ_VEN in ADJ_UPDATE should be set to 1 after setting the registers.

32.3.2 LTI Processing of Enhancer

Figure 32.8 shows an example of LTI adjustment.

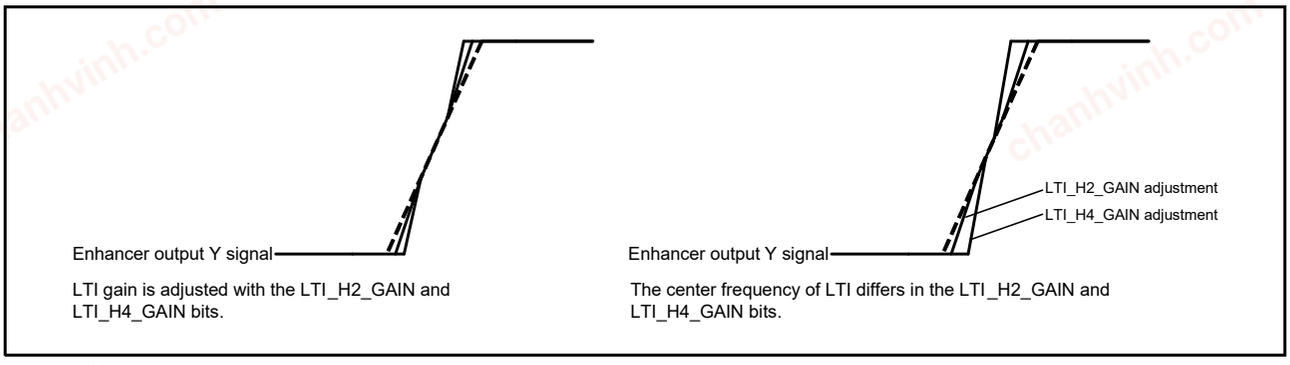


Figure 32.8 Example of LTI Adjustment

chanhvinh.com

32.3.3 Sharpness Processing of Enhancer

Figure 32.9 shows an example of sharpness adjustment.

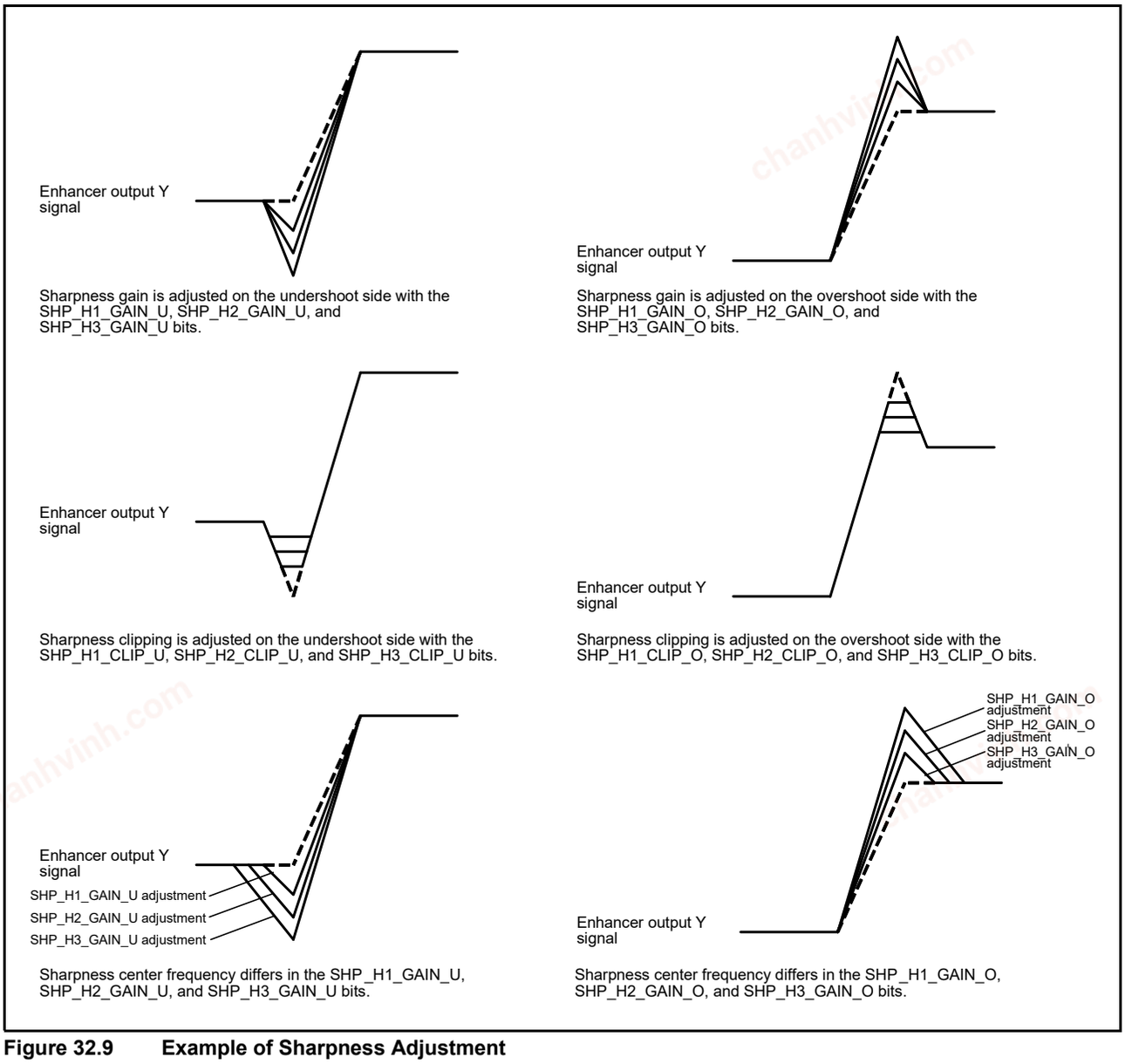


Figure 32.9 Example of Sharpness Adjustment

chanhvinh.com

32.3.4 Setting Method for Color Matrix Data Conversion

GBR signals are assumed to be input to the circuit subsequent to the image quality improver; therefore, the output from the color matrix circuit should be in the GBR format.

Table 32.12 shows an example of GBR conversion setting.

Table 32.12 Recommended Setting Values for Matrix Conversion

		GBR to GBR Conversion	YCBCR to GBR Conversion
Register Name	**Bit Name**	**Recommended Values**	**Recommended Values**
ADJ_MTX_MODE	ADJ_MTX_MD[1:0]	0	2
ADJ_MTX_YG_ADJ0	ADJ_MTX_YG[7:0]	128	128
ADJ_MTX_CBB_ADJ0	ADJ_MTX_B[7:0]	128	128
ADJ_MTX_CRR_ADJ0	ADJ_MTX_R[7:0]	128	128
ADJ_MTX_YG_ADJ0	ADJ_MTX_GG[10:0]	256	256
ADJ_MTX_YG_ADJ1	ADJ_MTX_GB[10:0]	0	1960
ADJ_MTX_YG_ADJ1	ADJ_MTX_GR[10:0]	0	1865
ADJ_MTX_CBB_ADJ0	ADJ_MTX_BG[10:0]	0	256
ADJ_MTX_CBB_ADJ1	ADJ_MTX_BB[10:0]	256	454
ADJ_MTX_CBB_ADJ1	ADJ_MTX_BR[10:0]	0	0
ADJ_MTX_CRR_ADJ0	ADJ_MTX_RG[10:0]	0	256
ADJ_MTX_CRR_ADJ1	ADJ_MTX_RB[10:0]	0	0
ADJ_MTX_CRR_ADJ1	ADJ_MTX_RR[10:0]	256	359

Note: ADJ_VEN in ADJ_UPDATE should be set to 1 after setting the registers.

chanhvinh.com

33. Video Display Controller 5 (5): Image Synthesizer

33.1 Image Synthesizer

33.1.1 Overview of Functions

The image synthesizer reads graphics data from the frame buffer and displays the synthesized image on the screen.

One video plane + two graphics planes, or three graphics planes can be selected for synthesis.

RGB565, RGB888, αRGB1555, αRGB4444, αRGB8888, RGBα5551, RGBα8888, CLUT8, CLUT4, CLUT1, YCbCr422 (for the graphics 0 process), and YCbCr444 (for the graphics 0 process) formats can be used for graphics data, and RGB565, RGB888, YCbCr422, and YCbCr444 formats for video data.

On each of the graphics planes, background color, lower-layer graphics, current graphics, or blended image (for the graphics 2 and 3 processes) of lower-layer graphics and current graphics can be displayed.

It is recommend that the frame buffer is placed in the on-chip large-capacity RAM. Both the on-chip large-capacity RAM and an external SDRAM can be used for the frame buffer, but the bus bandwidth might become short and display might not be possible if an external SDRAM is used for the frame buffer.

The functional block diagram of the image synthesizer is shown below.

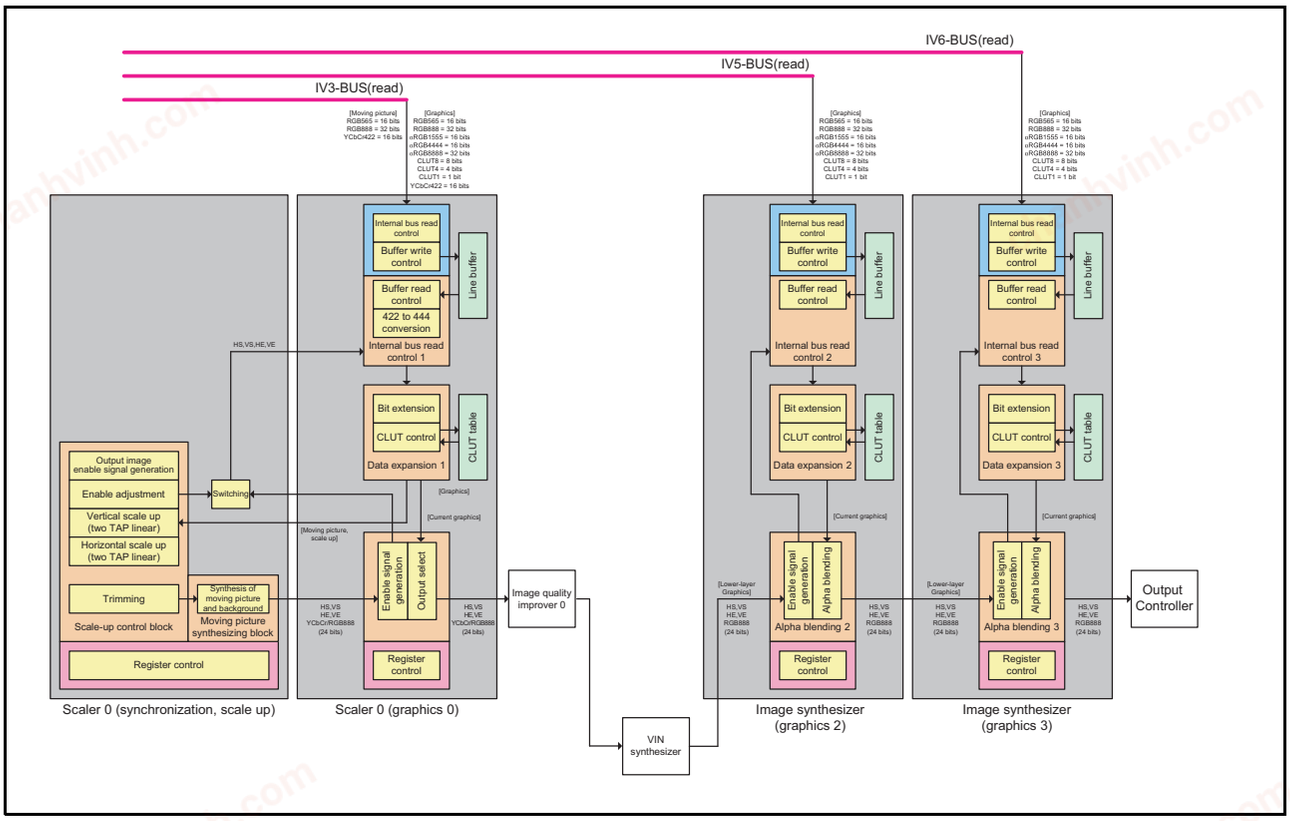


Figure 33.1 Functional Block Diagram of Image Synthesizer

chanhvinh.com

33.1.2 Graphics Data Read Control

Graphics data read can be controlled for the three processes: the graphics 0 process in the scaler 0, and the graphics 2 and 3 processes in the image synthesizer.

The register bits of each process can be identified by the number in the register name like GR0_xxxx, GR2_xxxx, and GR3_xxxx, respectively. In the sections except for Register Descriptions, however, the number is omitted like GR_xxxx for convenience sake.

In the VIN synthesizer, graphics data read is not controlled. In this manual, the name is omitted like GR_xxxx for convenience sake. The synthesizer does not have the read control register (GR_FLM).

(1) Updating Registers

The Vsync signal is used to control the update timing of the registers for graphics display and frame buffer read control, except for some of the registers.

After 1 is set to the bits in the update control register, the contents of the relevant registers are actually modified at the rising edge of the Vsync signal, when the update control register is automatically cleared to 0.

Table 33.1 Register Update Control

Register Name	Bit Name	Initial Value	Description
GR_UPDATE	GR_UPDATE	0	Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated.
GR_UPDATE	GR_P_VEN	0	Graphics Display Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
GR_UPDATE	GR_IBUS_VEN	0	Frame Buffer Read Register Update* 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

Note: * This bit is not supported for the VIN synthesizer.

(2) Frame Buffer Burst Transfer Mode

Either 32-byte or 128-byte transfer mode can be selected for accessing the frame buffer in which video data and graphics data are stored.

Table 33.2 Frame Buffer Burst Transfer Mode

Register Name	Bit Name	Initial Value	Description
GR_FLM1	GR_BST_MD	0	Frame Buffer Burst Transfer Mode 0: 32-byte transfer 1: 128- byte transfer

chanhvinh.com

(3) Frame Buffer Control Mode

More than one frame of data is read from the frame buffer.

For graphics data, set the GR_FLM_SEL[1:0] bits to 1, and set the specific display frame number with the GR_FLM_NUM[9:0] bits. For video data, select a mode with the GR_FLM_SEL[1:0] bits depending on the writing process used; the quantity of the frames used for video data is set in the writing process block.

Table 33.3 Frame Buffer Control Mode

Register Name	Bit Name	Initial Value	Description
GR_FLM1	GR_FLM_SEL[1:0]	0	Frame Buffer Address Setting Signal Select 0: Control linked to scaling-down process, or frame 0 selected.*1 1: Register GR_FLM_NUM selected. 2: Frame 0 selected. 3: Control linked to pointer buffer, or setting prohibited.*2
GR_FLM3	GR_FLM_NUM[9:0]	0	Frame Number of Frame Buffer Manually set the frame number when GR_FLM_SEL = 1.

Notes: 1. For the graphics 0 process, frame buffer control links to the scaling-down process. For the graphics 2 and 3 processes, frame 0 is selected.
2. For the graphics 0 process, frame buffer control links to the pointer buffer. For the graphics 2 and 3 processes, setting is prohibited.

(4) Frame Buffer Read Control

The following bit enables or disables read access to the frame buffer.

Table 33.4 Frame Buffer Read Control

Register Name	Bit Name	Initial Value	Description
GR_FLM_RD	GR_R_ENB	0	Frame Buffer Read Enable 0: Disables read access to the frame buffer. 1: Enables read access to the frame buffer.

chanhvinh.com

(5) Frame Buffer Size

The following bits set the size of the frame buffer to be read.

The numbers of horizontal pixels and of lines in the vertical direction are set with the GR_HW[10:0] and GR_FLM_LNUM[10:0] bits, respectively.

Table 33.5 Frame Buffer Size

Register Name	Bit Name	Initial Value	Description
GR_FLM6	GR_HW[10:0]	0	Sets the width of the horizontal valid period. The width is (GR_HW + 1) pixels. Note: Set to 2 or greater.
GR_FLM5	GR_FLM_LNUM[10:0]	0	Sets the number of lines in a frame The number of lines is (GR_FLM_LNUM + 1).

(6) Calculating Addresses in Frame Buffer

The data area in the frame buffer is defined using the addresses specified by GR_BASE[31:0], GR_LN_OFF[14:0], and GR_FLM_OFF[22:0] bits and the display frame number.

The GR_LN_OFF[14:0] and GR_FLM_OFF[22:0] bits should be set in units of 32/128 bytes (the lower 5/7 bits should be fixed to 0).

The GR_BASE[31:0] bits should be set in units of 64 bits to set the display data start position (the lower three bits should be fixed).

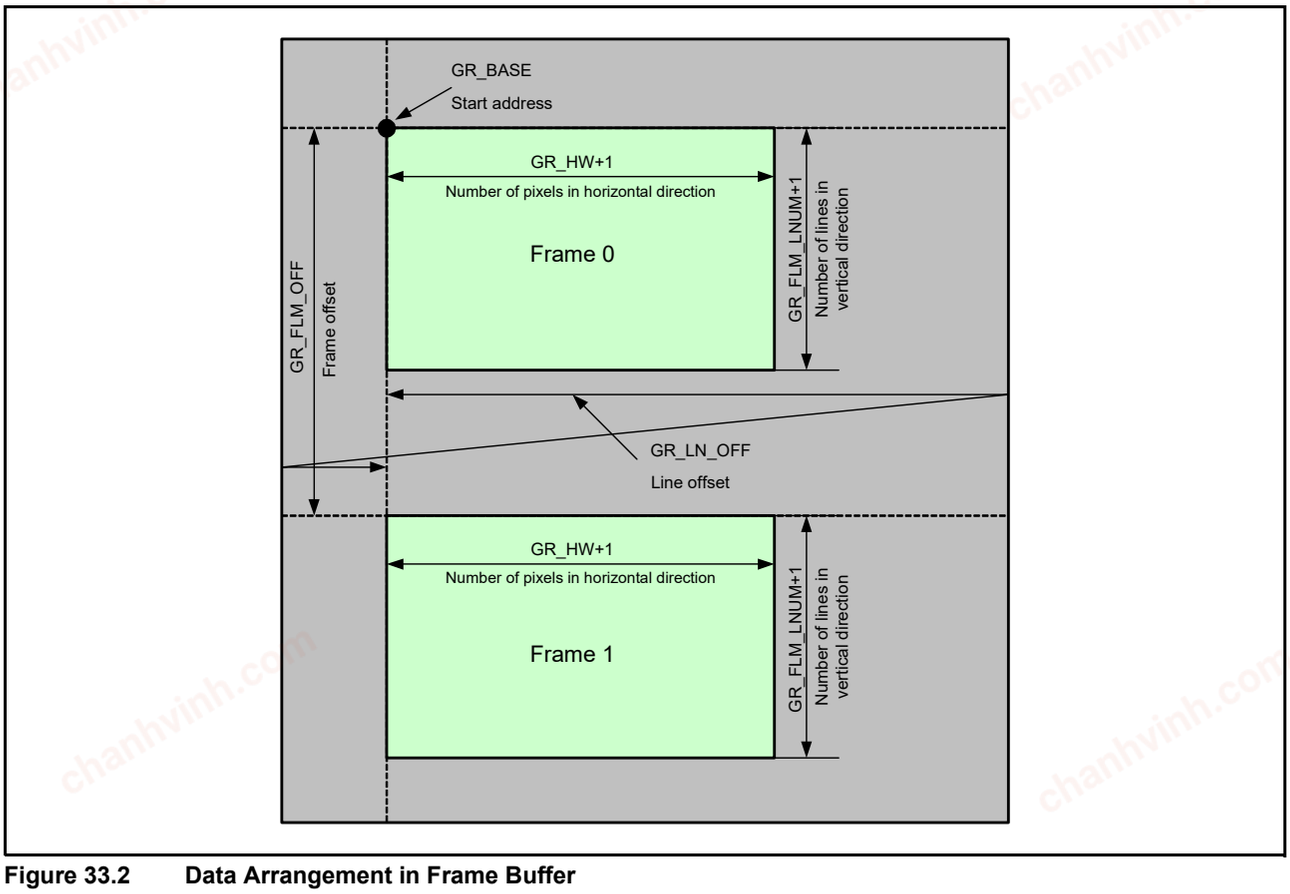


Figure 33.2 Data Arrangement in Frame Buffer

chanhvinh.com

Table 33.6 Calculation of Addresses in Frame Buffer

Register Name	Bit Name	Initial Value	Description
GR_FLM2	GR_BASE[31:0]	0	Frame Buffer Base Address Sets the start address of the frame buffer where frame data is to be stored. GR_BASE[4:3] and GR_BASE[6:3] are referred to during 32-byte burst transfer and 128-byte burst transfer, respectively, to skip the start line data. The lower 3 bits should be set to 000.
GR_FLM3	GR_LN_OFF[14:0]	0	Frame Buffer Line Offset Address Sets the line offset address for calculating the start address of each line. Line 0:GR_BASE Line 1:GR_BASE + GR_LN_OFF × 1 : Line n: GR_BASE + GR_LN_OFF × n For 32-byte transfer, the lower 5 bits should be fixed to 0_0000. For 128-byte transfer, the lower 7 bits should be fixed to 000_0000.
GR_FLM4	GR_FLM_OFF[22:0]	0	Frame Buffer Frame Offset Address (lower) Specifies the frame offset address used for calculating the start address of each frame buffer when more than one buffer is used. Buffer 0: GR_BASE Buffer 1: GR_BASE + GR_FLM_OFF × 1 : Buffer n: GR_BASE + GR_FLM_OFF × n For 32-byte transfer, the lower 5 bits should be fixed to 0_0000. For 128-byte transfer, the lower 7 bits should be fixed to 000_0000.

(7) Setting Frame Buffer Size Smaller than One Frame

Frame buffer size can be set in one-line units.

When the number of lines set with the GR_FLM_LOOP[10:0] bits is smaller than the value of the GR_FLM_LNUM[10:0] bits, data is again read from the start address of the frame buffer after the number of lines set with the (GR_FLM_LOOP[10:0] + 1) bits have been read.

Table 33.7 Setting of Frame Buffer Size Smaller than One Frame

Register Name	Bit Name	Initial Value	Description
GR_FLM5	GR_FLM_LOOP[10:0]	1023	Number of lines when reading the addresses repeatedly by returning to the start address after reaching the end address. The number of lines is (GR_FLM_LOOP + 1).

chanhvinh.com

(8) Line Offset Control for Frame Buffer

The following bit sets the line offset address direction of the frame buffer.

Table 33.8 Line Offset Address Direction Control for Frame Buffer

Register Name	Bit Name	Initial Value	Description
GR_FLM1	GR_LN_OFF_DIR	0	Selects the line offset address direction of the frame buffer. 0:Increments the address by the line offset address. 1:Decrements the address by the line offset address.

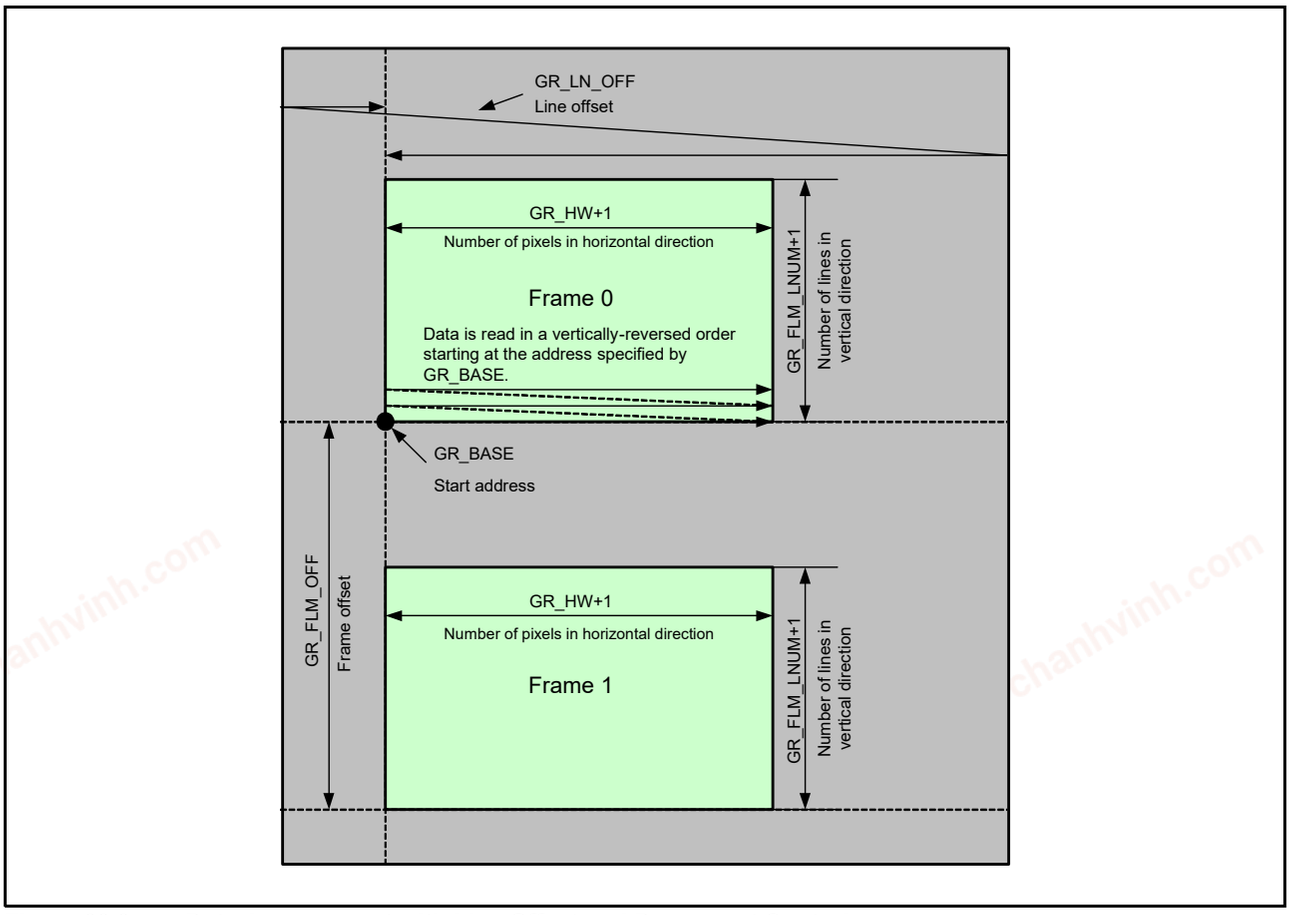


Figure 33.3 Data Arrangement with Line Offset and Decrement Control

chanhvinh.com

(9) Selecting Format of Frame Buffer Read Signal

Signal formats RGB565, RGB888, αRGB1555, αRGB4444, αRGB8888, RGBα5551, RGBα8888, CLUT8, CLUT4 and CLUT1 are supported for the graphics 0, 2, and 3 processes. The YCbCr422 and YCbCr444 formats are also supported for the graphics 0 process.

The GR_FORMAT[3:0] bits select a signal format.

Table 33.9 Format Selection for Frame Buffer Read Signal

Register Name	Bit Name	Initial Value	Description
GR_FLM6	GR_FORMAT[3:0]	0	Sets the format of the frame buffer read signal. 0: RGB565 1: RGB888 2: αRGB1555 3: αRGB4444 4: αRGB8888 5: CLUT8 6: CLUT4 7: CLUT1 8: YCbCr422 or setting prohibited * 9: YCbCr444 or setting prohibited * 10: RGBα5551 11: RGBα8888 12 to 15: Setting prohibited

Note: * Setting this value selects YCbCr422 and YCbCr444 for the graphics 0 process, and is prohibited for the graphics 2 and 3 processes.

(10) Endian Control

In the frame buffer, data is handled in 64-bit units, and endian of the data to be read can be controlled by setting the GR_RDSWA[2:0] bits. Bit 0 of these bits indicates whether 8-bit data is swapped. Bit 1 indicates whether 16-bit data is swapped. Bit 2 indicates whether 32-bit data is swapped. In the YCbCr422 format, data can be arranged with the GR_YCC_SWAP[2:0] bits.

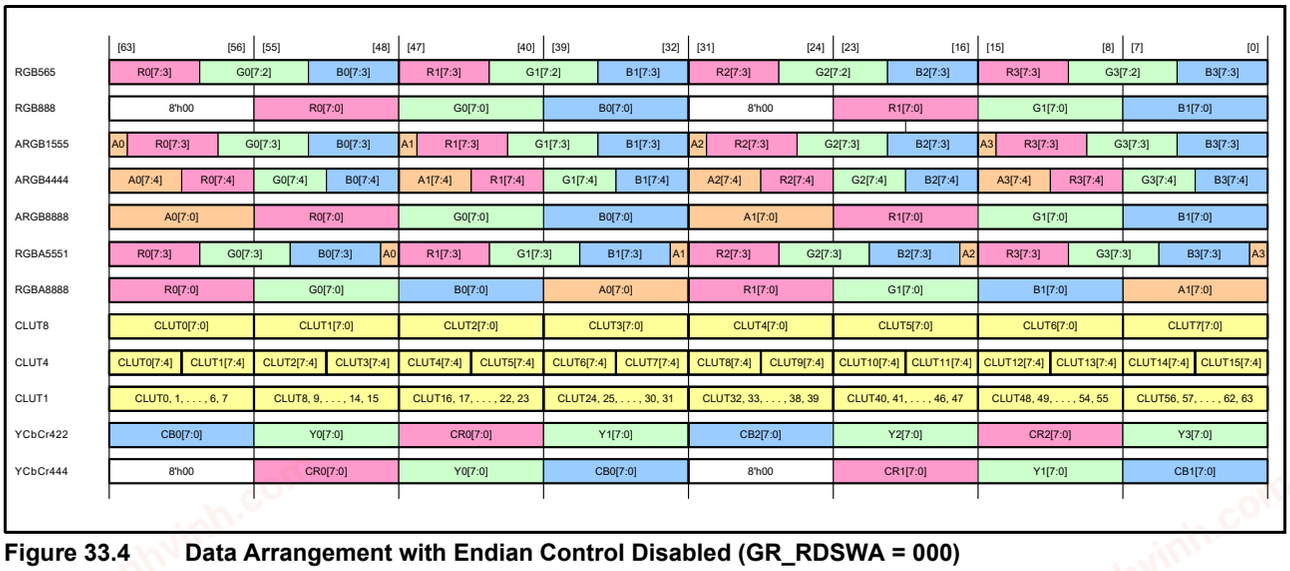


Figure 33.4 Data Arrangement with Endian Control Disabled (GR_RDSWA = 000)

chanhvinh.com

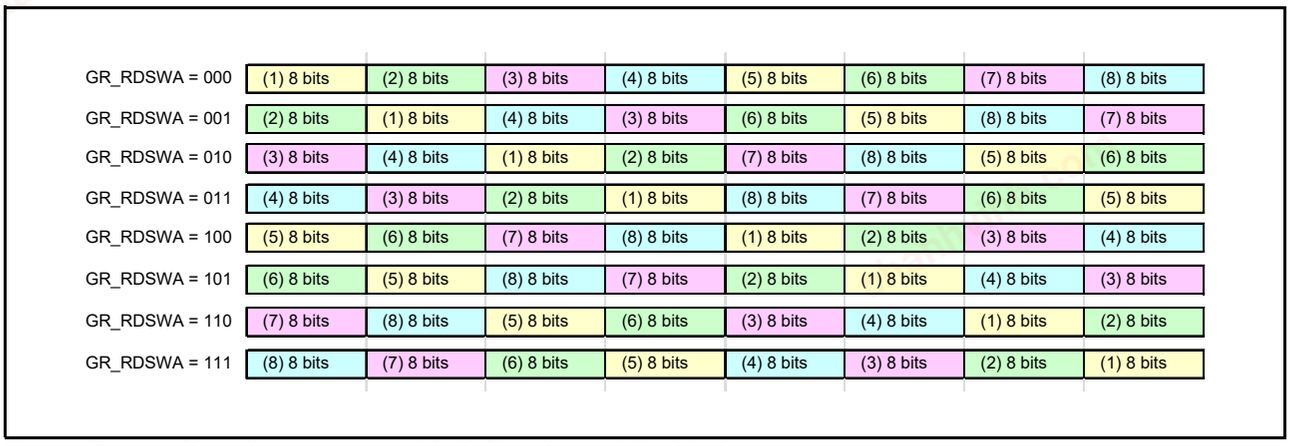


Figure 33.5 Data Arrangement with Endian Control Enabled

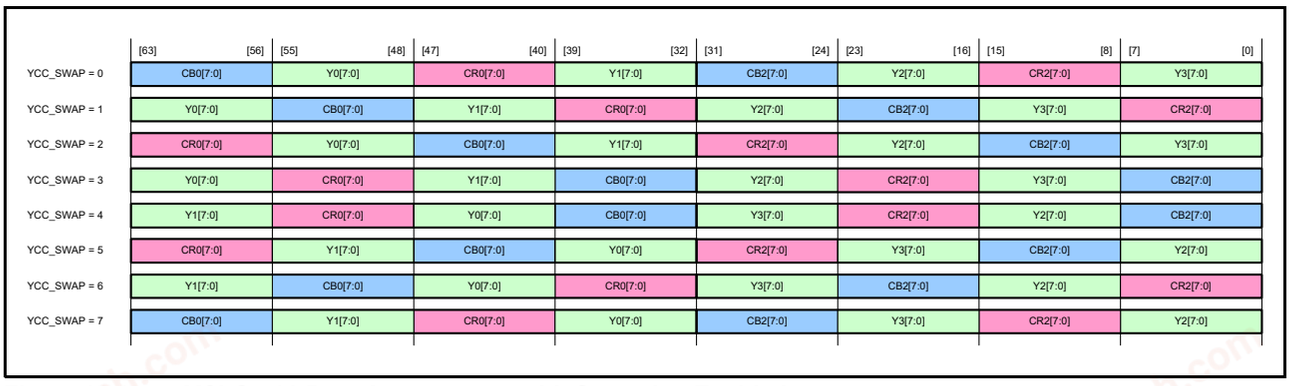


Figure 33.6 YCbCr422 Data Arrangement with Swapping Enabled

Table 33.10 Endian Control

Register Name	Bit Name	Initial Value	Description
GR_FLM6	GR_RDSWA[2:0]	0	Sets 8-, 16-, and 32-bit swap. These three bits specify the method for swapping the bits of frame buffer read data as follows. Bit 0 0: 8 bits are not swapped. 1: 8 bits are swapped. Bit 1 0: 16 bits are not swapped. 1: 16 bits are swapped. Bit 2 0: 32 bits are not swapped. 1: 32 bits are swapped. When eight bits are put together, they are swapped as follows. Each of (1) to (8) indicates eight-bit data. 000: (1) (2) (3) (4) (5) (6) (7) (8) [No swap] 001: (2) (1) (4) (3) (6) (5) (8) (7) [8-bit swap] 010: (3) (4) (1) (2) (7) (8) (5) (6) [16-bit swap] 011: (4) (3) (2) (1) (8) (7) (6) (5) [16-bit swap + 8-bit swap] 100: (5) (6) (7) (8) (1) (2) (3) (4) [32-bit swap] 101: (6) (5) (8) (7) (2) (1) (4) (3) [32-bit swap + 8-bit swap] 110: (7) (8) (5) (6) (3) (4) (1) (2) [32-bit swap + 16-bit swap] 111: (8) (7) (6) (5) (4) (3) (2) (1) [32-bit swap + 16-bit swap + 8-bit swap]

chanhvinh.com

Table 33.10 Endian Control

Register Name	Bit Name	Initial Value	Description
GR_FLM6	GR_YCC_SWAP [2:0]	0	Controls swapping of data read from buffer in the YCbCr422 format. * 0: Cb/Y0/Cr/Y1 1: Y0/Cb/Y1/Cr 2: Cr/Y0/Cb/Y1 3: Y0/Cr/Y1/Cb 4: Y1/Cr/Y0/Cb 5: Cr/Y1/Cb/Y0 6: Y1/Cb/Y0/Cr 7: Cb/Y1/Cr/Y0

Note: * These bits are supported for the graphics 0 process only.

(11) Display Start Pixel Setting for Read Data

When a horizontal offset is applied to display the image data in the frame buffer, the display start pixel is set with the GR_BASE[31:0] and GR_STA_POS[5:0] bits. Calculation of the values for the GR_BASE[31:0] and GR_STA_POS[5:0] bits depends on the signal format. The display start pixel can be calculated with the formulas in the table below, where H_OFF is a horizontal offset from the display start pixel.

Table 33.11 Calculation of Display Start Position for Various Signal Formats

Signal Format of Video/Graphics	Number of Bits per Pixel	Calculation Formula *1
RGB888 αRGB8888, RGBα8888 YCbCr422*2 YCbCr444*3	32	GR_BASE[31:3] = t (H_OFF ÷ 2) GR_STA_POS[5:0] = mod (H_OFF ÷ 2)
RGB565 αRGB1555, RGBα5551 αRGB4444	16	GR_BASE[31:3] = int (H_OFF ÷ 4) GR_STA_POS[5:0] = mod (H_OFF ÷ 4)
CLUT8	8	GR_BASE[31:3] = int (H_OFF ÷ 8) GR_STA_POS[5:0] = mod (H_OFF ÷ 8)
CLUT4	4	GR_BASE[31:3] = int (H_OFF ÷ 16) GR_STA_POS[5:0] = mod (H_OFF ÷ 16)
CLUT1	1	GR_BASE[31:3] = int (H_OFF ÷ 64) GR_STA_POS[5:0] = mod (H_OFF ÷ 64)

Notes: 1. The functions int() and mod() output a quotient and a remainder, respectively.
2. The YCbCr422 format is not supported for the graphics 2 and 3 processes.
In the YCbCr422 format, 32 bits are used for two pixels (Cb, Y0, Cr, and Y1 components). Therefore, the start position is controlled in units of 32 bits.
3. The YCbCr444 format is not supported for the graphics 2 and 3 processes.

chanhvinh.com

Table 33.12 Setting of Display Start Pixel of Read Data

Register Name	Bit Name	Initial Value	Description
GR_FLM6	GR_STA_POS[5:0]	0	Sets the amount of data to be skipped through. Specifically data amount equal to the amount indicated by GR_STA_POS is skipped from the start of the line.
GR_FLM2	GR_BASE[31:0]	0	Frame Buffer Base Address Sets the start address of the frame buffer where frame data is to be stored. GR_BASE[4:3] and GR_BASE[6:3] are referred to during 32-byte burst transfer and 128-byte burst transfer, respectively, to skip the start line data. The lower 3 bits should be fixed to 000.

(12) YCbCr422 to YCbCr444 Conversion

Data format for the graphics 0 process is converted from YCbCr422 to YCbCr444.

This function is not supported for the graphics 2 and 3 processes.

Table 33.13 YCbCr422 to YCbCr444 Conversion

Register Name	Bit Name	Initial Value	Description
GR_FLM6	GR_CNV444_MD	0	Sets the interpolation mode for YCbCr422 to YCbCr444 conversion. * 0: Hold interpolation 1: Average interpolation

Note: * This bit is not provided for the graphics 2 and 3 processes, for which the YCbCr422 format is not supported.

chanhvinh.com

(13) Bit Extension

When the value of the GR_FORMAT[3:0] bits is 0 to 3, the RGB565, RGB888, αRGB1555, and αRGB4444 formats are converted to the αRGB8888 format. When the value of the GR_FORMAT[3:0] bits is 10, the RGBα5551 format is converted to the RGBα8888 format. The RGBα5551 to RGBα8888 format conversion is omitted because it differs from the αRGB5551 to αRGB8888 format conversion only in the position of α.

- RGB565 to αRGB8888 Format Conversion

  After conversion, α[7:0] is fixed to 255.

  After conversion, R[7:0] = R[4:0] × 263 ÷ 32 (round off to an integer), approximation of #R[4:0] × 255 ÷ 31

  After conversion G[7:0] = G[5:0] × 259 ÷ 64 (round off to an integer), approximation of #G[5:0] × 255 ÷ 63

  After conversion, B[7:0] = B[4:0] × 263 ÷ 32 (round off to an integer), approximation of #B[4:0] × 255 ÷ 31

- RGB888 to αRGB8888 Format Conversion

  After conversion, α[7:0] is fixed to 255.

- αRGB1555 to αRGB8888 Format Conversion

  After conversion, α[7:0] is GR_A1 when α input is 1, and GR_A0 when 0.

  After conversion, R[7:0] = R[4:0] × 263 ÷ 32 (round off to an integer), approximation of #R[4:0] × 255 ÷ 31

  After conversion, G[7:0] = G[4:0] × 263 ÷ 32 (round off to an integer), approximation of #G[4:0] × 255 ÷ 31

  After conversion, B[7:0] = B[4:0] × 263 ÷ 32 (round off to an integer), approximation of #B[4:0] × 255 ÷ 31

- αRGB4444 to αRGB8888 Format Conversion

  After conversion, α[7:0] = α[3:0] × 17

  After conversion, R[7:0] = R[3:0] × 17

  After conversion, G[7:0] = G[3:0] × 17

  After conversion, B[7:0] = B[3:0] × 17

(14) Buffer Underflow Processing

When data read from the frame buffer cannot be completed due to bus-traffic related problems, an underflow interrupt signal is output.

chanhvinh.com

33.1.3 Setting Graphics Display Area

The graphics display area is set with the GR_GRC_HS[10:0], GR_GRC_HW[10:0], GR_GRC_VS[10:0], and GR_GRC_VW[10:0] bits based on the rising edges of the Hsync and Vsync signals.

Figure 33.7 shows the graphics display area.

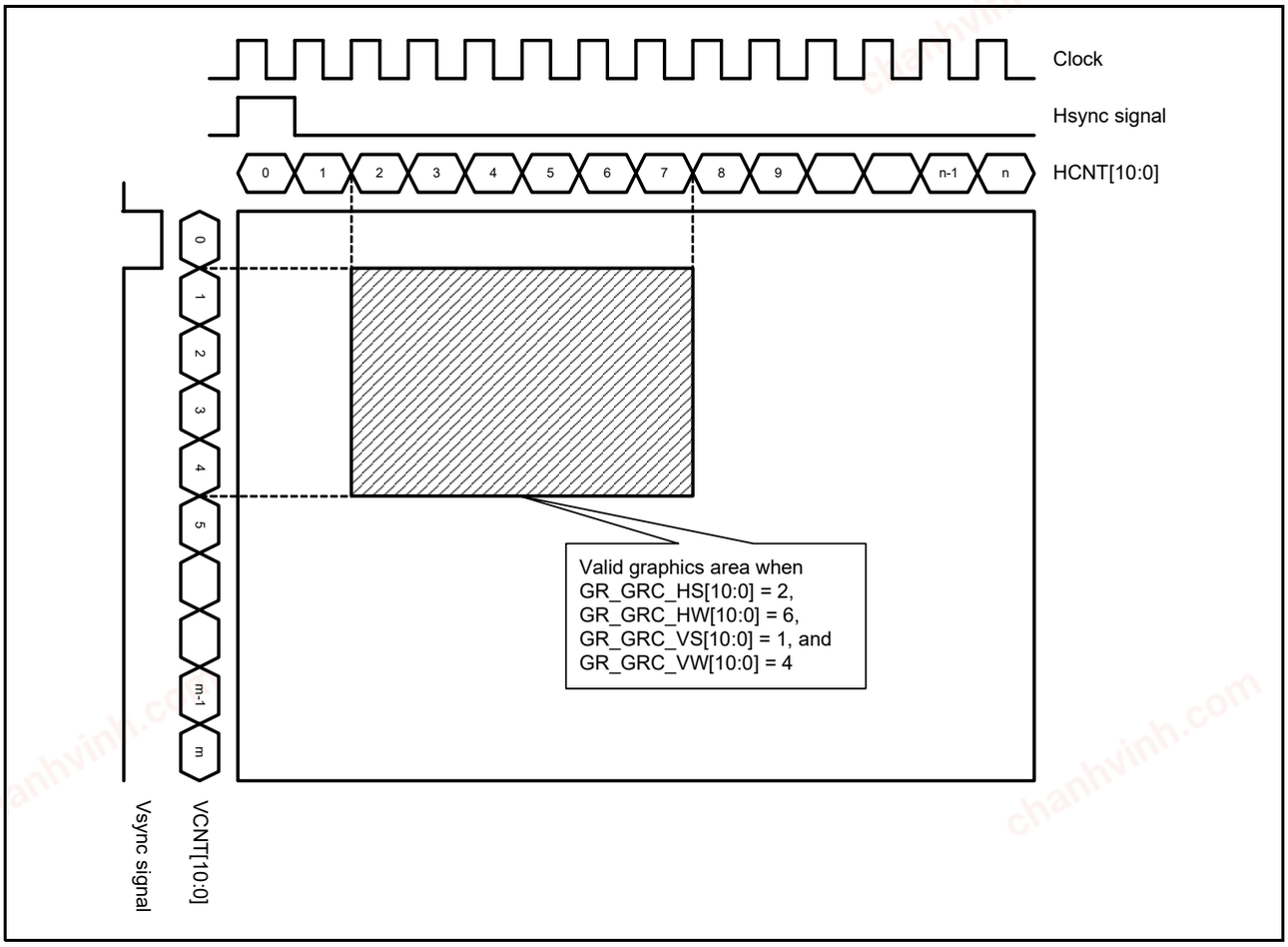


Figure 33.7 Graphics Display Area

The frame line of the graphics area can be displayed by setting the GR_GRC_DISP_ON bit to 1.

Table 33.14 Graphics Image Area Setting

Register Name	Bit Name	Initial Value	Description
GR_AB3	GR_GRC_HS[10:0]	0	Sets the horizontal start position of the graphics image area. Note: Set to 16 or greater clocks and the result of GR_GRC_HS + GR_GRC_HW should be smaller than or equal to 2015 clocks.
GR_AB3	GR_GRC_HW[10:0]	0	Sets the horizontal width of the graphics image area. Note: For displaying an image with 1- or 2-pixel horizontal width, set GR_HW to 2 and GR_GRC_HW to 1 (1-pixel) or 2 (2-pixel).
GR_AB2	GR_GRC_VS[10:0]	0	Sets the vertical start position of the graphics image area. Note: Set to 4 or greater lines and the result of GR_GRC_VS + GR_GRC_VW should be smaller than or equal to 2039 lines.
GR_AB2	GR_GRC_VW[10:0]	0	Sets the vertical width of the graphics image area.
GR_AB1	GR_GRC_DISP_ON	0	Turns on/off frame-line display of the graphics image area. 0: Frame-line display off 1: Frame-line display on

chanhvinh.com

33.1.4 Interrupt Generation at Specified Line

An interrupt signal can be generated at the line specified with the GR_LINE[10:0] bits.

Table 33.15 Interrupt Generation at Specified Line

Register Name	Bit Name	Initial Value	Description
GR_CLUT_INT	GR_LINE[10:0]	0	Line Interrupt Set * When the number of lines matches the value of the GR_LINE bits, an interrupt signal is output. This function is supported for the graphics 3 process only. This function supported for the graphics 3 process is enabled even when the graphics 3 process is not used.

Note: * This function is supported for the graphics 3 process only; these bits are not supported for the graphics 0 and 2 processes.

33.1.5 Formats of Frame Buffer Read Signals and Corresponding Alpha Blending Types

Setting the GR_FORMAT[3:0] bits selects the format of the signal read from the frame buffer.

Table 33.16 shows the signal formats and the corresponding alpha blending types. The priority of the alpha value is: alpha blending in rectangular area > chroma-key processing > alpha blending in pixel units.

Table 33.16 Formats of Frame Buffer Read Signal and Corresponding Alpha Blending Types

GR_FORMAT[3:0]	Signal Format	Alpha Blending in Rectangular Area	RGB-Index Chroma-Key Processing	CLUT-Index Chroma-Key Processing	Alpha Blending in Pixel Units
0	RGB565	Supported	Supported *1	Not supported	Not supported *2
1	RGB888	Supported	Supported	Not supported	Not supported *2
2	αRGB1555	Supported	Supported *1*3	Not supported	Supported *3
3	αRGB4444	Supported	Supported *1	Not supported	Supported
4	αRGB8888	Supported	Supported	Not supported	Supported
5	CLUT8	Supported	Not supported	Supported	Supported
6	CLUT4	Supported	Not supported	Supported	Supported
7	CLUT1	Supported *4	Not supported	Supported *4	Supported *4
8	YCbCr422	Not supported *5	Not supported *5	Not supported *5	Not supported *5
9	YCbCr444	Not supported *5	Not supported *5	Not supported *5	Not supported *5
10	RGBα5551	Supported	Supported *1*3	Not supported	Supported *3
11	RGBα8888	Supported	Supported	Not supported	Supported

Notes:
1. When each color component of the RGB signal read from the frame buffer is not 8 bits, it is converted to 8 bits by calculation in RGB-index chroma-key processing. (See section 33.1.2 (13) Bit Extension.)
2. Since α value is 255, the current graphics is always displayed.
3. α value for data read from the frame buffer is specified with one bit. This one-bit signal selects one of the two registers, each of which holds an 8-bit α value.
4. CLUT value for the frame buffer signal is specified with one bit. This one-bit signal selects one of the two registers, each of which holds the α, G, B, and R values (8 bits for each value). The CLUT table is not referenced.
5. YCbCr422 and YCbCr444 are supported for the graphics 0 process, but any type of blending and chroma-key processing cannot be used.

chanhvinh.com

33.1.6 Display Selection

The GR_DISP_SEL[1:0] bits select the graphics to be displayed from the background color, the lower-layer graphics, the current graphics, and the blended image of the lower-layer graphics and the current graphics. For blending, alpha blending in a rectangular area, multiplication with current alpha at alpha blending in a rectangular area, RGB-index chroma-key processing, CLUT-index chroma-key processing, alpha blending in one-pixel units, or premultiplication at alpha blending in one-pixel units can be selected.
This function is not supported for the VIN synthesizer.

Table 33.17 shows the settings for various display types.

Table 33.17 Settings for Various Display Types

GR_DISP_SEL [1:0]	GR_ARC_ON	GR_CK_ON	GR_ARC_MUL	GR_ACALC_MD	Processing for Graphics Area	Processing for the Area outside the Graphics Area
0	—	—	—	—	Background color	Background color
1	—	—	—	—	Lower-layer graphics	Lower-layer graphics
2	—	—	—	—	Current graphics	Background color
3	1	—	0	0	Alpha blending in a rectangular area*1	Lower-layer graphics
3	1	—	0	1	Setting prohibited	
3	1	—	1	0	Multiplication with current alpha at alpha blending in a rectangular area*2	Lower-layer graphics
3	1	—	1	1	Multiplication with current alpha at alpha blending in a rectangular area with alpha premultiplied*2	Lower-layer graphics
3	0	1	—	—	RGB-index or CLUT-index chroma-key processing	Lower-layer graphics
3	0	0	—	0	Alpha blending in one-pixel units*2	Lower-layer graphics
3	0	0	—	1	Premultiplication at alpha blending in one-pixel units*2	Lower-layer graphics

Notes: 1. The alpha blending in a rectangular area is not supported for the graphics 0 process in the scaler.
2. The multiplication with current alpha at alpha blending in a rectangular area and alpha blending function in one-pixel units are supported for the graphics 2 and 3 processes only.

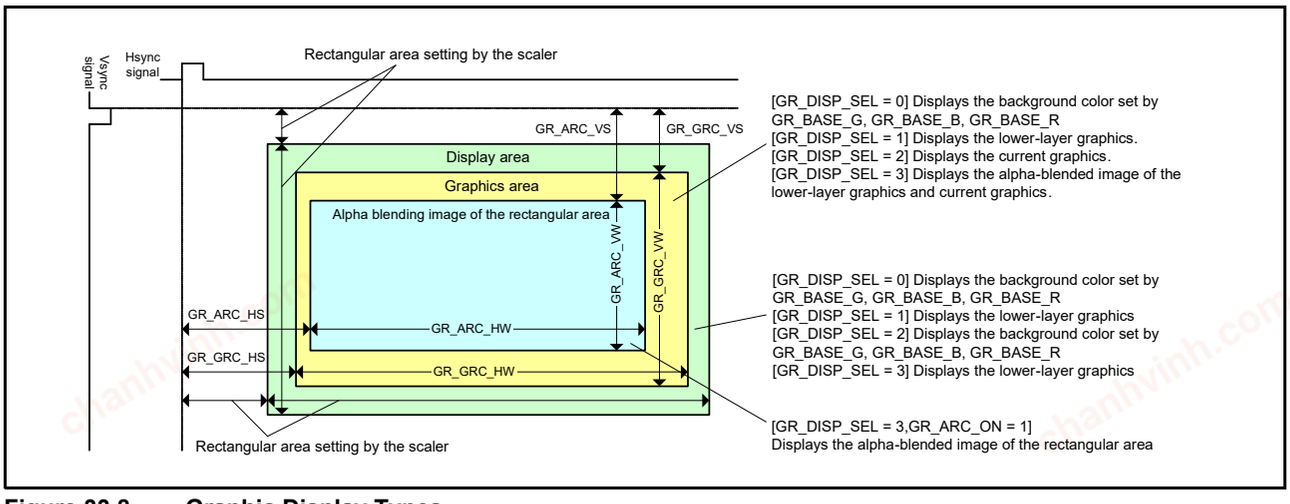


Figure 33.8 Graphic Display Types

chanhvinh.com

Figure 33.9 shows the graphics planes displayed when the GR_DISP_SEL bits are set to 3.

For correspondence between the lower-layer graphics and the current graphics, see Figure 33.1.

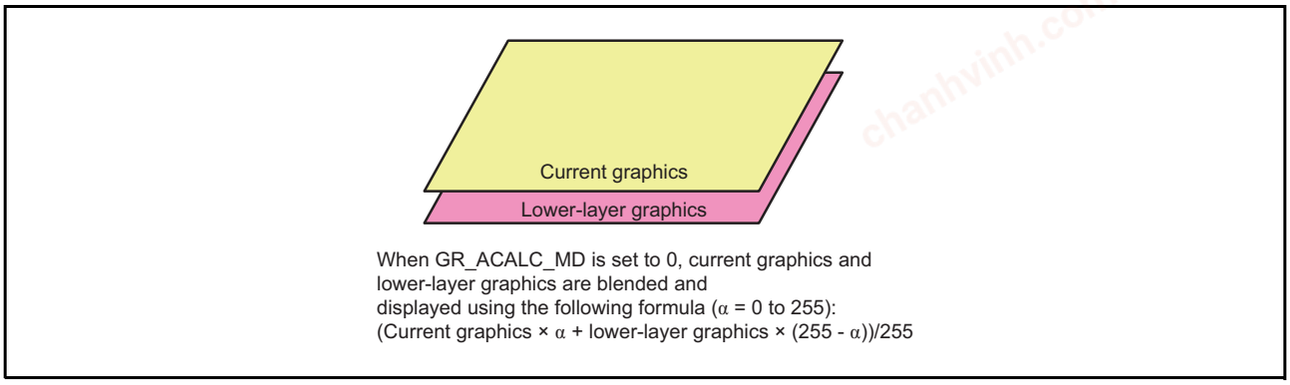


Figure 33.9 Graphics Planes with GR_DISP_SEL Set to 3

Table 33.18 Alpha Blending Setting

Register Name	Bit Name	Initial Value	Description
GR_AB1	GR_DISP_SEL [1:0]	0	Selects the graphics display mode. 0:Background color display 1:Lower-layer graphics display 2:Current graphics display 3:Blended display of lower-layer graphics and current graphics*1
GR_AB1	GR_ARC_ON	0	Turns on/off alpha blending in a rectangular area.*2 0: Off 1: On
GR_AB1	GR_ARC_MUL	0	Turns on/off multiplication processing with current alpha at alpha blending in a rectangular area *3 0: Off 1: On
GR_AB1	GR_ACALC_MD	0	Turns on/off premultiplication processing at alpha blending in one-pixel units *3 0: Off 1: On
GR_AB7	GR_CK_ON	0	Turns on/off CLUT-index/RGB-index chroma-key processing. 0: Off 1: On

Notes: 1. The graphics 0 process supports the chroma-key processing only. When performing chroma-key processing, set the α value for converting the pixels to be subjected to chroma-key processing, and the α value of the pixels not to be subjected to the chroma-key processing to 255 to display the current graphics only.
2. This function is supported only for the graphics 2 and 3 processes. This bit is not provided for the graphics 0 process.
3. This function is supported only for the graphics 2 and 3 processes. This bit is not provided for the graphics 0 process.

chanhvinh.com

33.1.7 Background Color Display Processing

The color set with the GR_BASE_G[7:0], GR_BASE_B[7:0], and GR_BASE_R[7:0] bits is displayed.

G output = GR_BASE_G

B output = GR_BASE_B

R output = GR_BASE_R

Table 33.19 Background Color Setting

Register Name	Bit Name	Initial Value	Description
GR_BASE	GR_BASE_G[7:0]	0	Background color G signal G: 8 bits; unsigned (0 to 255 [LSB])
GR_BASE	GR_BASE_B[7:0]	0	Background color B signal B: 8 bits; unsigned (0 to 255 [LSB])
GR_BASE	GR_BASE_R[7:0]	0	Background color R signal R: 8 bits; unsigned (0 to 255 [LSB])

33.1.8 Lower-Layer Graphics Display Processing

The lower-layer graphics are displayed as follows:

G output = G input of lower-layer graphics

B output = B input of lower-layer graphics

R output = R input of lower-layer graphics

33.1.9 Current Graphics Display Processing

The current graphics are displayed as follows:

G output = G input of current graphics

B output = B input of current graphics

R output = R input of current graphics

chanhvinh.com

33.1.10 Display with Alpha Blending in a Rectangular Area

The rectangular area subjected to alpha blending is set with the GR_ARC_HS[10:0], GR_ARC_HW[10:0], GR_ARC_VS[10:0], and GR_ARC_VW[10:0] bits based on the rising edges of the Hsync and Vsync signals. This function is not supported for the graphics 0 process.

Figure 33.10 shows the rectangular area setting for alpha blending.

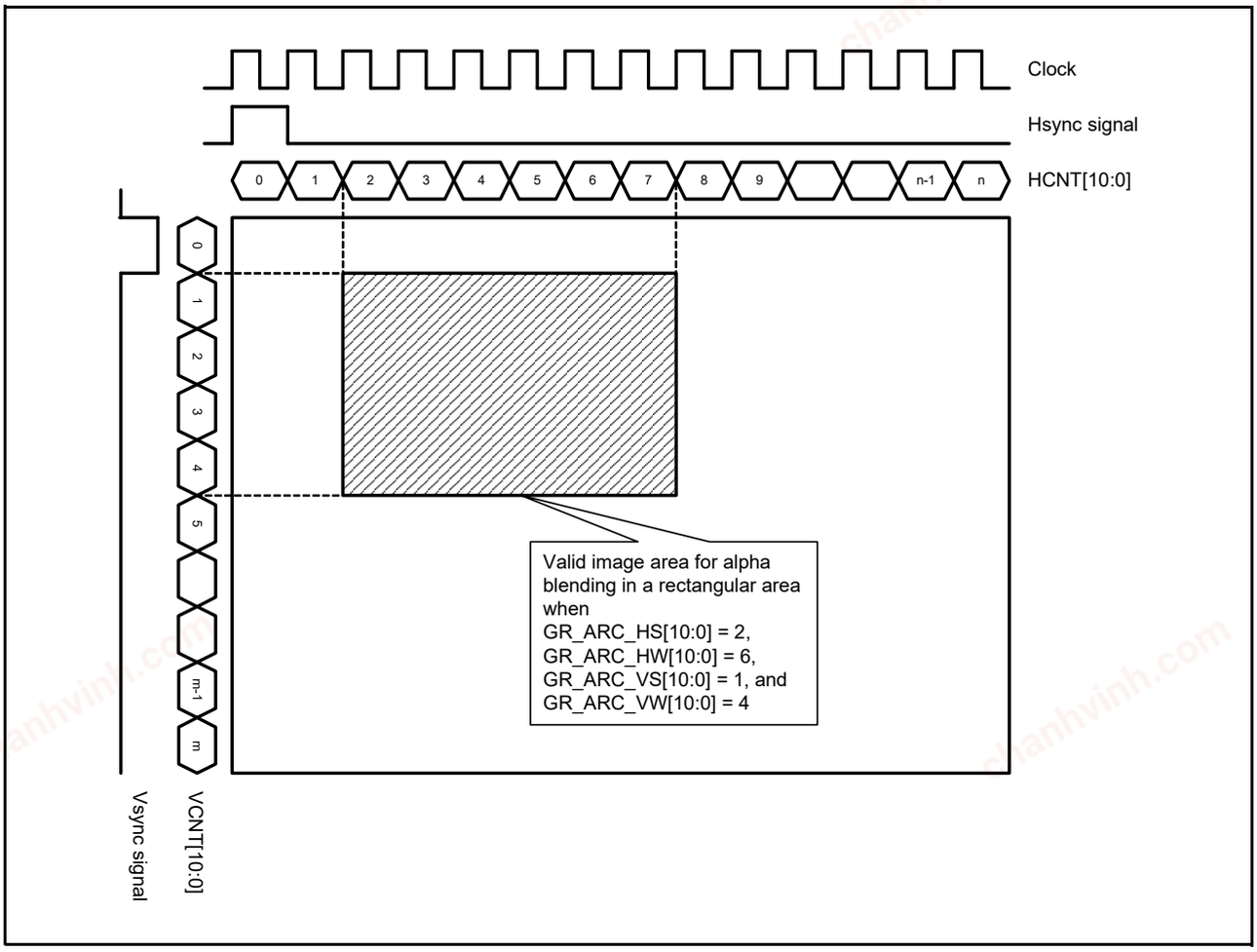


Figure 33.10 Rectangular Area Setting for Alpha Blending

The frame line of graphics area can be displayed by setting the GR_ARC_DISP_ON bit to 1.

chanhvinh.com

Table 33.20 Setting of Rectangular Area for Alpha Blending

Register Name	Bit Name	Initial Value	Description
GR_AB5	GR_ARC_HS[10:0]	0	Sets the horizontal start position of the valid image area for alpha blending in a rectangular area.
GR_AB5	GR_ARC_HW[10:0]	0	Sets the horizontal width of the valid image area for alpha blending in a rectangular area.
GR_AB4	GR_ARC_VS[10:0]	0	Sets the vertical start position of the valid image area for alpha blending in a rectangular area.
GR_AB4	GR_ARC_VW[10:0]	0	Sets the vertical width of the valid image area for alpha blending in a rectangular area.
GR_AB1	GR_ARC_DISP_ON	0	Turns on/off frame-line display of the valid image area for alpha blending in a rectangular area. 0: Frame-line display off 1: Frame-line display on

In alpha blending in a rectangular area, the current graphics are faded in or out by setting the fade-in or fade-out coefficients with the GR_ARC_DEF[7:0], GR_ARC_MODE, GR_ARC_COEF[7:0], and GR_ARC_RATE[7:0] bits.

First, the value of the GR_ARC_DEF[7:0] bits is assigned to the α value.

Then, each time the Vsync signal rises for the number of times set with the GR_ARC_RATE[7:0] bits + 1, the value of the GR_ARC_COEF[7:0] bit is added to or subtracted from the α value.

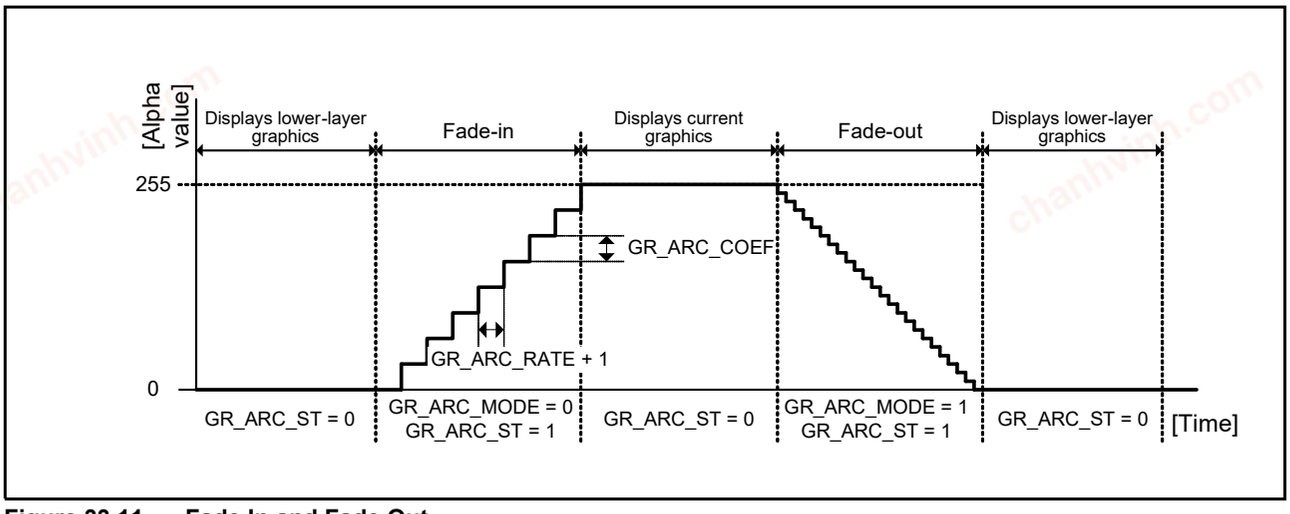


Figure 33.11 Fade In and Fade Out

Table 33.21 Setting for Alpha Blending in a Rectangular Area

Register Name	Bit Name	Initial Value	Description
GR_AB7	GR_ARC_DEF[7:0]	0	Sets the initial alpha value for alpha blending in a rectangular area.
GR_AB6	GR_ARC_MODE	0	Alpha Blending Mode in Rectangular Area 0: Addition 1: Subtraction
GR_AB6	GR_ARC_COEF[7:0]	0	Sets the alpha coefficient for alpha blending in a rectangular area. (0 to 255) [7:0]: Variation (absolute value)
GR_AB6	GR_ARC_RATE[7:0]	0	Sets the value obtained by subtracting 1 from the frame rate for alpha blending in a rectangular area.
GR_MON	GR_ARC_ST	—	Status Flag for Alpha Blending in Rectangular Area 0: Addition or subtraction has been completed. (α value is 0 or 255) 1: Addition or subtraction is in progress.

chanhvinh.com

The values specified with the following expressions are used in the alpha blending calculation described in section 33.1.14, Alpha Blending Calculation.

α value = Fade-in/out coefficient

G value = G input of the current graphics

B value = B input of the current graphics

R value = R input of the current graphics

33.1.11 RGB-Index Chroma-Key Processing

The pixels that satisfy all the expressions below are subjected to RGB-index chroma-key processing.

G input of the current graphics = GR_CK_KG

B input of the current graphics = GR_CK_KB

R input of the current graphics = GR_CK_KR

In RGB-index chroma-key processing, the values specified with the following expressions are used in the alpha blending calculation described in section 33.1.14, Alpha Blending Calculation. This function is not supported in the VIN synthesizer.

α value = GR_CK_A

G value = GR_CK_G

B value = GR_CK_B

R value = GR_CK_R

For the pixels that are not subjected to RGB-index chroma-key processing, the values specified with the following expressions are used in the alpha blending calculation described in section 33.1.14, Alpha Blending Calculation.

α value = α input of the current graphics

G value = G input of the current graphics

B value = B input of the current graphics

R value = R input of the current graphics

chanhvinh.com

Table 33.22 Setting for RGB-Index Chroma-Key Processing

Register Name	Bit Name	Initial Value	Description
GR_AB8	GR_CK_KG[7:0]	0	G Signal for RGB-Index Chroma-Key Processing G: 8 bits; unsigned (0 to 255 [LSB])
GR_AB8	GR_CK_KB[7:0]	0	B Signal for RGB-Index Chroma-Key Processing B: 8 bits; unsigned (0 to 255 [LSB])
GR_AB8	GR_CK_KR[7:0]	0	R Signal for RGB-Index Chroma-Key Processing R: 8 bits; unsigned (0 to 255 [LSB])
GR_AB9	GR_CK_A[7:0]	0	Replaced Alpha Signal after RGB-Index Chroma-Key Processing* α: 8 bits; unsigned (0 to 255 [LSB])
GR_AB9	GR_CK_G[7:0]	0	Replaced G Signal after RGB-Index Chroma-Key Processing G: 8 bits; unsigned (0 to 255 [LSB])
GR_AB9	GR_CK_B[7:0]	0	Replaced B Signal after RGB-Index Chroma-Key Processing B: 8 bits; unsigned (0 to 255 [LSB])
GR_AB9	GR_CK_R[7:0]	0	Replaced R Signal after RGB-Index Chroma-Key Processing R: 8 bits; unsigned (0 to 255 [LSB])

Note: * To use this function for the graphics 0 process, the alpha value should be set to 255.

33.1.12 CLUT-Index Chroma-Key Processing

The pixels that satisfy the expression below are subjected to CLUT-index chroma-key processing.

CLUT input of the current graphics = GR_CK_KCLUT

In CLUT-index chroma-key processing, the values specified with the following expressions are used in the alpha blending calculation described in section 33.1.14, Alpha Blending Calculation. This function is not supported in the VIN synthesizer.

α value = GR_CK_A

G value = GR_CK_G

B value = GR_CK_B

R value = GR_CK_R

For the pixels that are not subjected to CLUT-index chroma-key processing, the values specified with the following expressions are used in the alpha blending calculation described in section 33.1.14, Alpha Blending Calculation.

α value = α input of the current graphics

G value = G input of the current graphics

B value = B input of the current graphics

R value = R input of the current graphics

chanhvinh.com

Table 33.23 Setting for CLUT-Index Chroma-Key Processing

Register Name	Bit Name	Initial Value	Description
GR_AB8	GR_CK_ KCLUT[7:0]	0	CLUT Signal for CLUT-Index Chroma-Key Processing CLUT: 8 bits; unsigned (0 to 255 [LSB])
GR_AB10	GR_A0[7:0]	0	CLUT1 α0 Signal* Replaced with α signal when in the CLUT1 format and CLUT1= 0. Replaced with α signal when in the αRGB1555 format and α = 0.
GR_AB10	GR_G0[7:0]	0	CLUT1 G0 Signal Replaced with G signal when in the CLUT1 format and CLUT1 = 0.
GR_AB10	GR_B0[7:0]	0	CLUT1 B0 Signal Replaced with B signal when in the CLUT1 format and CLUT1 = 0.
GR_AB10	GR_R0[7:0]	0	CLUT1 R0 Signal Replaced with R signal when in the CLUT1 format and CLUT1 = 0.
GR_AB11	GR_A1[7:0]	0	CLUT1 α1 Signal* Replaced with α signal when in the CLUT1 format and CLUT1 = 1. Replaced with α signal when in the αRGB1555 format and α = 1.
GR_AB11	GR_G1[7:0]	0	CLUT1 G1 Signal Replaced with G signal when in the CLUT1 format and CLUT1 = 1.
GR_AB11	GR_B1[7:0]	0	CLUT1 B1 Signal Replaced with B signal when in the CLUT1 format and CLUT1 = 1.
GR_AB11	GR_R1[7:0]	0	CLUT1 R1 Signal Replaced with R signal when in the CLUT1 format and CLUT1 = 1.

Note: * To use this function for the graphics 0 process, the alpha value should be set to 255.

33.1.13 Display with Alpha Blending in One-Pixel Units

In the alpha blending in one-pixel units, the values specified with the following expressions are used in the alpha blending calculation described in section 33.1.14, Alpha Blending Calculation. This function is not supported in the graphics 0 process and the VIN synthesizer.

α value = α input of the current graphics

G value = G input of the current graphics

B value = B input of the current graphics

R value = R input of the current graphics

chanhvinh.com

33.1.14 Alpha Blending Calculation

Alpha blending of two input signals is performed using the α value as described below (rounded up if the result includes a decimal fraction).

[GR_ACALC_MD = 0]

G output = (G value × α value + G input of the lower-layer graphics × (255 − α value)) ÷ 256

B output = (B value × α value + B input of the lower-layer graphics × (255 − α value)) ÷ 256

R output = (R value × α value + R input of the lower-layer graphics × (255 − α value)) ÷ 256

[GR_ACALC_MD = 1 (premultiplication)]

G output = (G value + G input of the lower-layer graphics × (255 − α value)) ÷ 256

B output = (B value + B input of the lower-layer graphics × (255 − α value)) ÷ 256

R output = (R value + R input of the lower-layer graphics × (255 − α value)) ÷ 256

33.1.15 CLUT Table

When the signal format is CLUT8 or CLUT4, the format is converted to αRGB8888 based on the CLUT table. When the format is CLUT1, it is converted to αRGB8888 based on the register value.

Figure 33.12 shows data arrangement in the CLUT table.

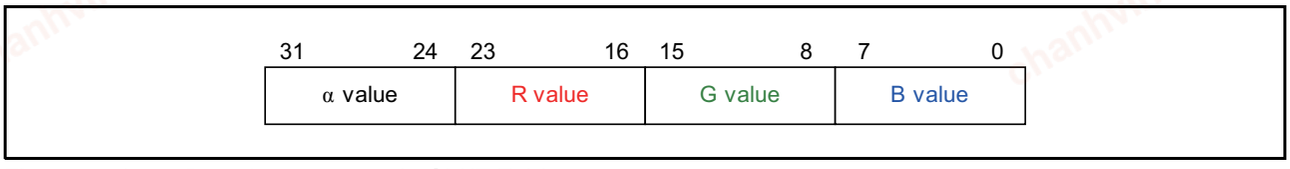


Figure 33.12 Data Arrangement in CLUT Table

The CLUT tables are arranged in the following addresses:

Graphics 0 CLUT table:	H'FCFF6000 to H'FCFF63FF (For CLUT4, addresses H'FCFF6000 to H'FCFF603F are valid.)
Graphics 2 CLUT table:	H'FCFF6800 to H'FCFF6BFF (For CLUT4, addresses H'FCFF6800 to H'FCFF683F are valid.)
Graphics 3 CLUT table:	H'FCFF6C00 to H'FCFF6FFF (For CLUT4, addresses H'FCFF6C00 to H'FCFF6C3F are valid.)

Two CLUT tables (CLUT table 0, CLUT table 1) on the different planes are allocated to the same address and one of the tables is selected with the GR_CLT_SEL bit. This allows rewriting one CLUT table when this module refers to the other CLUT table.
When switching the CLUT after overwriting the CLUT, please execute a dummy read of the CLUT table address.And switch the CLUT table by GR_CLT_SEL after dummy-read.switch the CLUT table by GR_CLT_SEL after dummy-read.

chanhvinh.com

Table 33.24 CLUT Table Selection

Register Name	Bit Name	Initial Value	Description
GR_CLUT	GR_CLT_SEL	0	CLUT Table Select Signal 0: Selects CLUT table 0. The format is converted to αRGB8888 based on the CLUT table 0. CLUT table 1 can be read from or written to by the CPU. 1: Selects CLUT table 1. The format is converted to αRGB8888 based on the CLUT table 1. CLUT table 0 can be read from or written to by the CPU.

33.1.16 Multiplication Processing with Current Alpha at Alpha Blending in Rectangular Area

In multiplication processing with current alpha at alpha blending in a rectangular area, the values specified with the following expressions are used in the alpha blending calculation described in section 33.1.14, Alpha Blending Calculation.

[GR_ARC_MUL = 0]

α value = Fade-in/out coefficient

G value = G input of the current graphics

B value = B input of the current graphics

R value = R input of the current graphics

[GR_ARC_MUL = 1 (multiplication)]

α value = Fade-in/out coefficient x α input of current graphics

G value = G input of the current graphics

B value = B input of the current graphics

R value = R input of the current graphics

33.1.17 Selection of Lower-Layer Graphics in VIN Synthesizer

Graphics 0 is allocated to the lower-layer graphics in the VIN synthesizer.

Table 33.25 Selection of Lower-Layer Plane in Scaler

Register Name	Bit Name	Initial Value	Description
GR_VIN_AB1	GR_VIN_SCL_UND_SEL	0	Specifies lower-layer plane in the scaler. 0: Selects graphics 0 as lower-layer graphics. 1: Setting prohibited

chanhvinh.com

33.2 Register Descriptions

Table 33.26 to Table 33.29 show the register configuration.

- Symbols used in Register Description:

Initial value:	Register value after a reset
—:	Undefined value
R/W:	Readable/writable. The written value can be read.
R/WC0:	Readable/writable. Writing 0 initializes the bit. Writing 1 is ignored.
R/WC1:	Readable/writable. Writing 1 initializes the bit. Writing 0 is ignored.
R:	Read-only. The write value should always be 0.
—/W:	Write-only. The read value is undefined.

Table 33.26 shows the register configuration for the graphics 2 process.

Table 33.27 shows the register configuration for the graphics 3 process.

Table 33.28 shows the CLUT table configuration.

Table 33.29 shows the register configuration for the VIN synthesizer.

The register configuration for the graphics 0 process is described in section 31, Video Display Controller 5 (3): Scaler.

chanhvinh.com

Table 33.26 Register Configuration of the Image Synthesizer (Graphics 2 Process)

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Graphics 2 register update control register	GR2_UPDATE	R/WC1	H'0000 0000	H'FCFF 7700	32
Frame buffer read control register (Graphics 2)	GR2_FLM_RD	R/W	H'0000 0000	H'FCFF 7704	32
Frame buffer control register 1 (Graphics 2)	GR2_FLM1	R/W	H'0000 0000	H'FCFF 7708	32
Frame buffer control register 2 (Graphics 2)	GR2_FLM2	R/W	H'0000 0000	H'FCFF 770C	32
Frame buffer control register 3 (Graphics 2)	GR2_FLM3	R/W	H'0000 0000	H'FCFF 7710	32
Frame buffer control register 4 (Graphics 2)	GR2_FLM4	R/W	H'0000 0000	H'FCFF 7714	32
Frame buffer control register 5 (Graphics 2)	GR2_FLM5	R/W	H'0000 03FF	H'FCFF 7718	32
Frame buffer control register 6 (Graphics 2)	GR2_FLM6	R/W	H'0000 0000	H'FCFF 771C	32
Alpha blending control register 1 (Graphics 2)	GR2_AB1	R/W	H'0000 0000	H'FCFF 7720	32
Alpha blending control register 2 (Graphics 2)	GR2_AB2	R/W	H'0000 0000	H'FCFF 7724	32
Alpha blending control register 3 (Graphics 2)	GR2_AB3	R/W	H'0000 0000	H'FCFF 7728	32
Alpha blending control register 4 (Graphics 2)	GR2_AB4	R/W	H'0000 0000	H'FCFF 772C	32
Alpha blending control register 5 (Graphics 2)	GR2_AB5	R/W	H'0000 0000	H'FCFF 7730	32
Alpha blending control register 6 (Graphics 2)	GR2_AB6	R/W	H'0000 0000	H'FCFF 7734	32
Alpha blending control register 7 (Graphics 2)	GR2_AB7	R/W	H'00FF 0000	H'FCFF 7738	32
Alpha blending control register 8 (Graphics 2)	GR2_AB8	R/W	H'0000 0000	H'FCFF 773C	32
Alpha blending control register 9 (Graphics 2)	GR2_AB9	R/W	H'0000 0000	H'FCFF 7740	32
Alpha blending control register 10 (Graphics 2)	GR2_AB10	R/W	H'0000 0000	H'FCFF 7744	32
Alpha blending control register 11 (Graphics 2)	GR2_AB11	R/W	H'0000 0000	H'FCFF 7748	32
Background color control register (Graphics 2)	GR2_BASE	R/W	H'0000 0000	H'FCFF 774C	32
CLUT table control register (Graphics 2)	GR2_CLUT	R/W	H'0000 0000	H'FCFF 7750	32
Status monitor register (Graphics 2)	GR2_MON	R	H'0000 0000	H'FCFF 7754	32

chanhvinh.com

Table 33.27 Register Configuration of the Image Synthesizer (Graphics 3 Process)

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Graphics 3 register update control register	GR3_UPDATE	R/WC1	H'0000 0000	H'FCFF 7780	32
Frame buffer read control register (Graphics 3)	GR3_FLM_RD	R/W	H'0000 0000	H'FCFF 7784	32
Frame buffer control register 1 (Graphics 3)	GR3_FLM1	R/W	H'0000 0000	H'FCFF 7788	32
Frame buffer control register 2 (Graphics 3)	GR3_FLM2	R/W	H'0000 0000	H'FCFF 778C	32
Frame buffer control register 3 (Graphics 3)	GR3_FLM3	R/W	H'0000 0000	H'FCFF 7790	32
Frame buffer control register 4 (Graphics 3)	GR3_FLM4	R/W	H'0000 0000	H'FCFF 7794	32
Frame buffer control register 5 (Graphics 3)	GR3_FLM5	R/W	H'0000 03FF	H'FCFF 7798	32
Frame buffer control register 6 (Graphics 3)	GR3_FLM6	R/W	H'0000 0000	H'FCFF 779C	32
Alpha blending control register 1 (Graphics 3)	GR3_AB1	R/W	H'0000 0000	H'FCFF 77A0	32
Alpha blending control register 2 (Graphics 3)	GR3_AB2	R/W	H'0000 0000	H'FCFF 77A4	32
Alpha blending control register 3 (Graphics 3)	GR3_AB3	R/W	H'0000 0000	H'FCFF 77A8	32
Alpha blending control register 4 (Graphics 3)	GR3_AB4	R/W	H'0000 0000	H'FCFF 77AC	32
Alpha blending control register 5 (Graphics 3)	GR3_AB5	R/W	H'0000 0000	H'FCFF 77B0	32
Alpha blending control register 6 (Graphics 3)	GR3_AB6	R/W	H'0000 0000	H'FCFF 77B4	32
Alpha blending control register 7 (Graphics 3)	GR3_AB7	R/W	H'00FF 0000	H'FCFF 77B8	32
Alpha blending control register 8 (Graphics 3)	GR3_AB8	R/W	H'0000 0000	H'FCFF 77BC	32
Alpha blending control register 9 (Graphics 3)	GR3_AB9	R/W	H'0000 0000	H'FCFF 77C0	32
Alpha blending control register 10 (Graphics 3)	GR3_AB10	R/W	H'0000 0000	H'FCFF 77C4	32
Alpha blending control register 11 (Graphics 3)	GR3_AB11	R/W	H'0000 0000	H'FCFF 77C8	32
Background color control register (Graphics 3)	GR3_BASE	R/W	H'0000 0000	H'FCFF 77CC	32
CLUT table and interrupt control register (Graphics 3)	GR3_CLUT_INT	R/W	H'0000 0000	H'FCFF 77D0	32
Status monitor register (Graphics 3)	GR3_MON	R	H'0000 0000	H'FCFF 77D4	32

chanhvinh.com

Table 33.28 CLUT Table Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Graphics 0 CLUT table	GR0_CLUTT	R/W	—	H'FCFF 6000 to H'FCFF 63FF	32
Graphics 2 CLUT table	GR2_CLUTT	R/W	—	H'FCFF 6800 to H'FCFF 6BFF	32
Graphics 3 CLUT table	GR3_CLUTT	R/W	—	H'FCFF 6C00 to H'FCFF 6FFF	32

Table 33.29 Register Configuration of the VIN Synthesizer

Name	Abbreviation	R/W	Initial Value	Address	Access Size
VIN synthesizer register update control register	GR_VIN_UPDATE	R/WC1	H'0000 0000	H'FCFF 7E00	32
Alpha blending control register 1 (VIN synthesizer)	GR_VIN _AB1	R/W	H'0000 0000	H'FCFF 7E20	32

chanhvinh.com

33.2.1 Graphics 2 Register Update Control Register (GR2_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GR2_UPDATE	—	—	—	GR2_P_VEN	—	—	—	GR2_IBUS_VEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/WC1	R	R	R	R/WC1	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	GR2_UPDATE	0	R/WC1	Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	GR2_P_VEN	0	R/WC1	Graphics Display Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR2_IBUS_VEN	0	R/WC1	Frame Buffer Read Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

33.2.2 Frame Buffer Read Control Register (Graphics 2) (GR2_FLM_RD)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR2_R_ENB
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR2_R_ENB	0	R/W	Frame Buffer Read Enable 0: Frame buffer reading is disabled. 1: Frame buffer reading is enabled.

Note: This register is updated when GR2_IBUS_VEN in GR2_UPDATE is 1.

chanhvinh.com

33.2.3 Frame Buffer Control Register 1 (Graphics 2) (GR2_FLM1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR2_LN_OFF_DIR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	GR2_FLM_SEL[1:0]		—	—	—	—	—	—	—	GR2_BST_MD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	GR2_LN_OFF_DIR	0	R/W	Selects the line offset address direction of the frame buffer. 0: Increments the address by the line offset address. 1: Decrements the address by the line offset address.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	GR2_FLM_SEL[1:0]	0	R/W	Selects a frame buffer address setting signal. 0: Selects frame 0. 1: Selects register GR2_FLM_NUM. 2: Selects frame 0. 3: Setting prohibited
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR2_BST_MD	0	R/W	Frame Buffer Burst Transfer Mode 0: 32-byte transfer 1: 128- byte transfer

Note: GR2_LN_OFF_DIR and GR2_FLM_SEL are updated when GR2_IBUS_VEN in GR2_UPDATE is 1.
GR2_BST_MD is updated when GR2_IBUS_VEN and GR2_P_VEN in GR2_UPDATE are 1.

chanhvinh.com

33.2.4 Frame Buffer Control Register 2 (Graphics 2) (GR2_FLM2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR2_BASE[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR2_BASE[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	GR2_BASE [31:0]	0	R/W	Frame Buffer Base Address (upper) Sets the start address of the frame buffer where frame data is to be stored. GR_BASE[4:3] and GR_BASE[6:3] are referred to during 32-byte burst transfer and 128-byte burst transfer, respectively, to skip the start line data. The lower 3 bits should be fixed to 000.

Note: This register is updated when GR2_IBUS_VEN and GR2_P_VEN in GR2_UPDATE are 1.

33.2.5 Frame Buffer Control Register 3 (Graphics 2) (GR2_FLM3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	GR2_LN_OFF[14:0]														
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	GR2_FLM_NUM[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 16	GR2_LN_ OFF[14:0]	0	R/W	Frame Buffer Line Offset Address Sets the line offset address for calculating the start address of each line. Line 0: GR2_BASE Line 1: GR2_BASE + GR2_LN_OFF × 1 : Line n: GR2_BASE + GR2_LN_OFF × n For 32 byte transfer, the lower 5 bits should be fixed to 0_0000. For 128 byte transfer, the lower 7 bits should be fixed to 000_0000.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	GR2_FLM_ NUM[9:0]	0	R/W	Frame Number of Frame Buffer Manually set the frame number when GR2_FLM_SEL = 1.

Note: This register is updated when GR2_IBUS_VEN in GR2_UPDATE is 1.

chanhvinh.com

33.2.6 Frame Buffer Control Register 4 (Graphics 2) (GR2_FLM4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	GR2_FLM_OFF[22:16]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR2_FLM_OFF[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
22 to 0	GR2_FLM_OFF[22:0]	0	R/W	Frame Buffer Frame Offset Address (upper) Specifies the frame offset address used for calculating the start address of each frame buffer when more than one buffer is used. Buffer 0: GR2_BASE Buffer 1: GR2_BASE + GR2_FLM_OFF × 1 : Buffer n: GR2_BASE + GR2_FLM_OFF × n For 32 byte transfer, the lower 5 bits should be fixed to 0_0000. For 128 byte transfer, the lower 7 bits should be fixed to 000_0000.

Note: This register is updated when GR2_IBUS_VEN in GR2_UPDATE is 1.

33.2.7 Frame Buffer Control Register 5 (Graphics 2) (GR2_FLM5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR2_FLM_LNUM[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR2_FLM_LOOP[10:0]										
Initial value:	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR2_FLM_LNUM[10:0]	0	R/W	Sets number of lines in a frame The number of lines is (GR2_FLM_LNUM + 1).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR2_FLM_LOOP[10:0]	1023	R/W	Number of lines when reading the addresses repeatedly by returning to the start address after reaching the end address. The number of lines is (GR2_FLM_LOOP + 1).

Note: This register is updated when GR2_IBUS_VEN in GR2_UPDATE is 1.

chanhvinh.com

33.2.8 Frame Buffer Control Register 6 (Graphics 2) (GR2_FLM6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR2_FORMAT[3:0]				—	GR2_HW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	GR2_RDSWA[2:0]			—	—	—	—	GR2_STA_POS[5:0]					
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	GR2_FORMAT [3:0]	0	R/W	Sets the format of the frame buffer read signal. 0: RGB565 1: RGB888 2: αRGB1555 3: αRGB4444 4: αRGB8888 5: CLUT8 6: CLUT4 7: CLUT1 8: Setting prohibited 9: Setting prohibited 10: RGBα5551 11: RGBα8888 12 to 15: Setting prohibited
27	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
26 to 16	GR2_HW [10:0]	0	R/W	Sets the width of the horizontal valid period. The width is (GR2_HW + 1) pixels. Note: Set to 2 or greater.
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12 to 10	GR2_RDSWA [2:0]	0	R/W	Sets 8-, 16-, and 32-bit swap. These three bits specify the method for swapping the bits of frame buffer read data as follows. Bit 0 0: 8 bits are not swapped. 1: 8 bits are swapped. Bit 1 0: 16 bits are not swapped. 1: 16 bits are swapped. Bit 2 0: 32 bits are not swapped. 1: 32 bits are swapped. When eight bits are put together, they are swapped as follows. Each of (1) to (8) indicates eight-bit data. 000: (1) (2) (3) (4) (5) (6) (7) (8) [No swap] 001: (2) (1) (4) (3) (6) (5) (8) (7) [8-bit swap] 010: (3) (4) (1) (2) (7) (8) (5) (6) [16-bit swap] 011: (4) (3) (2) (1) (8) (7) (6) (5) [16-bit swap + 8-bit swap] 100: (5) (6) (7) (8) (1) (2) (3) (4) [32-bit swap] 101: (6) (5) (8) (7) (2) (1) (4) (3) [32-bit swap + 8-bit swap] 110: (7) (8) (5) (6) (3) (4) (1) (2) [32-bit swap + 16-bit swap] 111: (8) (7) (6) (5) (4) (3) (2) (1) [32-bit swap + 16-bit swap + 8-bit swap]
9 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
5 to 0	GR2_ STA_POS [5:0]	0	R/W	Sets the amount of data to be skipped through. Specifically data amount equal to the amount indicated by GR2_STA_POS is skipped from the start of the line.

Note: GR2_STA_POS is updated when GR2_P_VEN in GR2_UPDATE is 1. GR2_RDSWA is updated when GR2_UPDATE in GR2_UPDATE is 1.
GR2_FORMAT and GR2_HW are updated when GR2_IBUS_VEN and GR2_P_VEN in GR2_UPDATE are 1.

33.2.9 Alpha Blending Control Register 1 (Graphics 2) (GR2_AB1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR2_ARC_MUL	GR2_ACALC_MD	—	GR2_ARC_ON	—	—	—	GR2_ARC_DISP_ON	—	—	—	GR2_GRC_DISP_ON	—	—	GR2_DISP_SEL[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15	GR2_ARC_MUL	0	R/W	Turns on/off multiplication processing with current alpha at alpha blending in a rectangular area. 0: Off 1: On
14	GR2_ACALC_MD	0	R/W	Turns on/off premultiplication processing at alpha blending in one-pixel units. 0: Off 1: On
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	GR2_ARC_ON	0	R/W	Turns on/off alpha blending in a rectangular area. 0: Off 1: On
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	GR2_ARC_ DISP_ON	0	R/W	Turns on/off frame-line display of the image area for alpha blending in a rectangular area. 0: Frame-line display off 1: Frame-line display on
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	GR2_GRC_ DISP_ON	0	R/W	Turns on/off frame-line display of the graphics image area. 0: Frame-line display off 1: Frame-line display on
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
1, 0	GR2_DISP_SEL[1:0]	0	R/W	Selects the graphics display mode. 0: Background color display 1: Lower-layer graphics display 2: Current graphics display 3: Blended display of lower-layer graphics and current graphics

Note: This register is updated when GR2_P_VEN in GR2_UPDATE is 1.

33.2.10 Alpha Blending Control Register 2 (Graphics 2) (GR2_AB2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR2_GRC_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR2_GRC_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR2_GRC_VS[10:0]	0	R/W	Sets the vertical start position of the graphics image area. Note: Set to 4 or greater lines and the result of GR2_GRC_VS + GR2_GRC_VW should be smaller than or equal to 2039 lines.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR2_GRC_VW[10:0]	0	R/W	Sets the vertical width of the graphics image area.

Note: This register is updated when GR2_P_VEN in GR2_UPDATE is 1.

chanhvinh.com

33.2.11 Alpha Blending Control Register 3 (Graphics 2) (GR2_AB3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR2_GRC_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR2_GRC_HW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR2_GRC_HS[10:0]	0	R/W	Sets the horizontal start position of the graphics image area. Note: Set to 16 or greater clocks and the result of GR2_GRC_HS + GR2_GRC_HW should be smaller than or equal to 2015 clocks.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR2_GRC_HW[10:0]	0	R/W	Sets the horizontal width of the graphics image area. Note: For displaying an image with 1- or 2-pixel horizontal width, set GR2_HW to 2 and GR2_GRC_HW to 1 (1-pixel) or 2 (2-pixel).

Note: This register is updated when GR2_P_VEN in GR2_UPDATE is 1.

33.2.12 Alpha Blending Control Register 4 (Graphics 2) (GR2_AB4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR2_ARC_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR2_ARC_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR2_ARC_VS[10:0]	0	R/W	Sets the vertical start position of the valid image area for alpha blending in a rectangular area.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR2_ARC_VW[10:0]	0	R/W	Sets the vertical width of the valid image area for alpha blending in a rectangular area.

Note: This register is updated when GR2_P_VEN in GR2_UPDATE is 1.

chanhvinh.com

33.2.13 Alpha Blending Control Register 5 (Graphics 2) (GR2_AB5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR2_ARC_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR2_ARC_HW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR2_ARC_HS[10:0]	0	R/W	Sets the horizontal start position of the valid image area for alpha blending in a rectangular area.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR2_ARC_HW[10:0]	0	R/W	Sets the horizontal width of the valid image area for alpha blending in a rectangular area.

Note: This register is updated when GR2_P_VEN in GR2_UPDATE is 1.

33.2.14 Alpha Blending Control Register 6 (Graphics 2) (GR2_AB6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	GR2_ARC_MODE	GR2_ARC_COEF[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	GR2_ARC_RATE[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	GR2_ARC_MODE	0	R/W	Alpha Blending Mode in Rectangular Area 0: Addition 1: Subtraction
23 to 16	GR2_ARC_COEF[7:0]	0	R/W	Sets the alpha coefficient for alpha blending in a rectangular area. (0 to 255) [7:0]: Variation (absolute value)
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	GR2_ARC_RATE[7:0]	0	R/W	Sets the value obtained by subtracting 1 from the frame rate for alpha blending in a rectangular area.

Note: This bit is updated when GR2_P_VEN in GR2_UPDATE is 1.

chanhvinh.com

33.2.15 Alpha Blending Control Register 7 (Graphics 2) (GR2_AB7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	GR2_ARC_DEF[7:0]							
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR2_CK_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	GR2_ARC_DEF[7:0]	255	R/W	Sets the initial alpha value for alpha blending in a rectangular area.
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR2_CK_ON	0	R/W	Turns on/off CLUT-index/RGB-index chroma-key processing. 0: Off 1: On

Note: This register is updated when GR2_P_VEN in GR2_UPDATE is 1.

33.2.16 Alpha Blending Control Register 8 (Graphics 2) (GR2_AB8)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR2_CK_KCLUT[7:0]								GR2_CK_KG[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR2_CK_KB[7:0]								GR2_CK_KR[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR2_CK_KCLUT[7:0]	0	R/W	CLUT Signal for CLUT-Index Chroma-Key Processing CLUT: Unsigned 8 bits (0 to 255 [LSB])
23 to 16	GR2_CK_KG[7:0]	0	R/W	G Signal for RGB-Index Chroma-Key Processing G: Unsigned 8 bits (0 to 255 [LSB])
15 to 8	GR2_CK_KB[7:0]	0	R/W	B Signal for RGB-Index Chroma-Key Processing B: Unsigned 8 bits (0 to 255 [LSB])
7 to 0	GR2_CK_KR[7:0]	0	R/W	R Signal for RGB-Index Chroma-Key Processing R: Unsigned 8 bits (0 to 255 [LSB])

Note: This register is updated when GR2_P_VEN in GR2_UPDATE is 1.

chanhvinh.com

33.2.17 Alpha Blending Control Register 9 (Graphics 2) (GR2_AB9)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR2_CK_A[7:0]								GR2_CK_G[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR2_CK_B[7:0]								GR2_CK_R[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR2_CK_A [7:0]	0	R/W	Replaced Alpha Signal after RGB/CLUT-Index Chroma-Key Processing α: Unsigned 8 bits (0 to 255 [LSB])
23 to 16	GR2_CK_G [7:0]	0	R/W	Replaced G Signal after RGB/CLUT-Index Chroma-Key Processing G: Unsigned 8 bits (0 to 255 [LSB])
15 to 8	GR2_CK_B [7:0]	0	R/W	Replaced B Signal after RGB/CLUT-Index Chroma-Key Processing B: Unsigned 8 bits (0 to 255 [LSB])
7 to 0	GR2_CK_R [7:0]	0	R/W	Replaced R Signal after RGB/CLUT-Index Chroma-Key Processing R: Unsigned 8 bits (0 to 255 [LSB])

Note: This register is updated when GR2_P_VEN in GR2_UPDATE is 1.

33.2.18 Alpha Blending Control Register 10 (Graphics 2) (GR2_AB10)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR2_A0[7:0]								GR2_G0[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR2_B0[7:0]								GR2_R0[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR2_A0 [7:0]	0	R/W	CLUT1 α0 Signal Replaced with α signal when in the CLUT1 format and CLUT1= 0. Replaced with α signal when in the αRGB1555 or RGBα5551 format and α = 0.
23 to 16	GR2_G0 [7:0]	0	R/W	CLUT1 G0 Signal Replaced with G signal when in the CLUT1 format and CLUT1 = 0.
15 to 8	GR2_B0 [7:0]	0	R/W	CLUT1 B0 Signal Replaced with B signal when in the CLUT1 format and CLUT1 = 0.
7 to 0	GR2_R0 [7:0]	0	R/W	CLUT1 R0 Signal Replaced with R signal when in the CLUT1 format and CLUT1 = 0.

Note: This register is updated when GR2_P_VEN in GR2_UPDATE is 1.

chanhvinh.com

33.2.19 Alpha Blending Control Register 11 (Graphics 2) (GR2_AB11)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR2_A1[7:0]								GR2_G1[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR2_B1[7:0]								GR2_R1[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR2_A1 [7:0]	0	R/W	CLUT1 α1 Signal Replaced with α signal when in the CLUT1 format and CLUT1 = 1. Replaced with α signal when in the αRGB1555 or RGBα5551 format and α = 1.
23 to 16	GR2_G1 [7:0]	0	R/W	CLUT1 G1 Signal Replaced with G signal when in the CLUT1 format and CLUT1 = 1.
15 to 8	GR2_B1 [7:0]	0	R/W	CLUT1 B1 Signal Replaced with B signal when in the CLUT1 format and CLUT1 = 1.
7 to 0	GR2_R1 [7:0]	0	R/W	CLUT1 R1 Signal Replaced with R signal when in the CLUT1 format and CLUT1 = 1.

Note: This register is updated when GR2_P_VEN in GR2_UPDATE is 1.

33.2.20 Background Color Control Register (Graphics 2) (GR2_BASE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	GR2_BASE_G[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR2_BASE_B[7:0]								GR2_BASE_R[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	GR2_BASE_G [7:0]	0	R/W	Background Color G Signal G: Unsigned 8 bits (0 to 255 [LSB])
15 to 8	GR2_BASE_B [7:0]	0	R/W	Background Color B Signal B: Unsigned 8 bits (0 to 255 [LSB])
7 to 0	GR2_BASE_R [7:0]	0	R/W	Background Color R Signal R: Unsigned 8 bits (0 to 255 [LSB])

Note: This register is updated when GR2_P_VEN in GR2_UPDATE is 1.

chanhvinh.com

33.2.21 CLUT Table Control Register (Graphics 2) (GR2_CLUT)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR2_CLT_SEL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	GR2_CLT_SEL	0	R/W	CLUT Table Select Signal 0: Selects CLUT table 0. The format is converted to αRGB8888 based on the CLUT table 0. CLUT table 1 can be read from or written to by the CPU. 1: Selects CLUT table 1. The format is converted to αRGB8888 based on the CLUT table 1. CLUT table 0 can be read from or written to by the CPU.
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: This register is updated when GR2_P_VEN in GR2_UPDATE is 1.

33.2.22 Status Monitor Register (Graphics 2) (GR2_MON)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR2_ARC_ST
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR2_ARC_ST	0	R	Status Flag for Alpha Blending in Rectangular Area 0: Addition or subtraction has been completed. (α value is 0 or 255) 1: Addition or subtraction is in progress.

chanhvinh.com

33.2.23 Graphics 3 Register Update Control Register (GR3_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GR3_UPDATE	—	—	—	GR3_P_VEN	—	—	—	GR3_IBUS_VEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/WC1	R	R	R	R/WC1	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	GR3_UPDATE	0	R/WC1	Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	GR3_P_VEN	0	R/WC1	Graphics Display Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR3_IBUS_VEN	0	R/WC1	Frame Buffer Read Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

33.2.24 Frame Buffer Read Control Register (Graphics 3) (GR3_FLM_RD)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR3_R_ENB
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR3_R_ENB	0	R/W	Frame Buffer Read Enable 0: Frame buffer reading is disabled. 1: Frame buffer reading is enabled.

Note: This register is updated when GR3_IBUS_VEN in GR3_UPDATE is 1.

chanhvinh.com

33.2.25 Frame Buffer Control Register 1 (Graphics 3) (GR3_FLM1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR3_LN_OFF_DIR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	GR3_FLM_SEL[1:0]		—	—	—	—	—	—	—	GR3_BST_MD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	GR3_LN_OFF_DIR	0	R/W	Selects the line offset address direction of the frame buffer. 0: Increments the address by the line offset address. 1: Decrements the address by the line offset address.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	GR3_FLM_SEL[1:0]	0	R/W	Selects a frame buffer address setting signal. 0: Selects frame 0. 1: Selects register GR3_FLM_NUM. 2: Selects frame 0. 3: Setting prohibited
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR3_BST_MD	0	R/W	Frame Buffer Burst Transfer Mode 0: 32-byte transfer 1: 128- byte transfer

Note: GR3_LN_OFF_DIR and GR3_FLM_SEL are updated when GR3_IBUS_VEN in GR3_UPDATE is 1.
GR3_BST_MD is updated when GR3_IBUS_VEN and GR3_P_VEN in GR3_UPDATE are 1.

chanhvinh.com

33.2.26 Frame Buffer Control Register 2 (Graphics 3) (GR3_FLM2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR3_BASE[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR3_BASE[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	GR3_BASE [31:0]	0	R/W	Frame Buffer Base Address (upper) Sets the start address of the frame buffer where frame data is to be stored. GR_BASE[4:3] and GR_BASE[6:3] are referred to during 32-byte burst transfer and 128-byte burst transfer, respectively, to skip the start line data. The lower 3 bits should be fixed to 000.

Note: This register is updated when GR3_IBUS_VEN and GR3_P_VEN in GR3_UPDATE are 1.

33.2.27 Frame Buffer Control Register 3 (Graphics 3) (GR3_FLM3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	GR3_LN_OFF[14:0]														
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	GR3_FLM_NUM[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 16	GR3_LN_ OFF[14:0]	0	R/W	Frame Buffer Line Offset Address Sets the line offset address for calculating the start address of each line. Line 0: GR3_BASE Line 1: GR3_BASE + GR3_LN_OFF × 1 : Line n: GR3_BASE + GR3_LN_OFF × n For 32 byte transfer, the lower 5 bits should be fixed to 0_0000. For 128 byte transfer, the lower 7 bits should be fixed to 000_0000.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	GR3_FLM_ NUM[9:0]	0	R/W	Frame Number of Frame Buffer Manually set the frame number when GR3_FLM_SEL = 1.

Note: This register is updated when GR3_IBUS_VEN in GR3_UPDATE is 1.

chanhvinh.com

33.2.28 Frame Buffer Control Register 4 (Graphics 3) (GR3_FLM4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	GR3_FLM_OFF[22:16]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR3_FLM_OFF[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
22 to 0	GR3_FLM_OFF[22:0]	0	R/W	Frame Buffer Frame Offset Address (upper) Specifies the frame offset address used for calculating the start address of each frame buffer when more than one buffer is used. Buffer 0: GR3_BASE Buffer 1: GR3_BASE + GR3_FLM_OFF × 1 : Buffer n: GR3_BASE + GR3_FLM_OFF × n For 32 byte transfer, the lower 5 bits should be fixed to 0_0000. For 128 byte transfer, the lower 7 bits should be fixed to 000_0000.

Note: This register is updated when GR3_IBUS_VEN in GR3_UPDATE is 1.

33.2.29 Frame Buffer Control Register 5 (Graphics 3) (GR3_FLM5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR3_FLM_LNUM[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR3_FLM_LOOP[10:0]										
Initial value:	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR3_FLM_LNUM[10:0]	0	R/W	Sets number of lines in a frame The number of lines is (GR3_FLM_LNUM + 1).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR3_FLM_LOOP[10:0]	1023	R/W	Number of lines when reading the addresses repeatedly by returning to the start address after reaching the end address. The number of lines is (GR3_FLM_LOOP + 1).

Note: This register is updated when GR3_IBUS_VEN in GR3_UPDATE is 1.

chanhvinh.com

33.2.30 Frame Buffer Control Register 6 (Graphics 3) (GR3_FLM6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR3_FORMAT[3:0]				—	GR3_HW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	GR3_RDSWA[2:0]			—	—	—	—	GR3_STA_POS[5:0]					
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	GR3_ FORMAT [3:0]	0	R/W	Sets the format of the frame buffer read signal. 0: RGB565 1: RGB888 2: αRGB1555 3: αRGB4444 4: αRGB8888 5: CLUT8 6: CLUT4 7: CLUT1 8: Setting prohibited 9: Setting prohibited 10: RGBα5551 11: RGBα8888 12 to 15: Setting prohibited
27	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
26 to 16	GR3_HW [10:0]	0	R/W	Sets the width of the horizontal valid period. The width is (GR3_HW + 1) pixels. Note: Set to 2 or greater.
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12 to 10	GR3_ RDSWA [2:0]	0	R/W	Sets 8-, 16-, and 32-bit swap. These three bits specify the method for swapping the bits of frame buffer read data as follows. Bit 0 0: 8 bits are not swapped. 1: 8 bits are swapped. Bit 1 0: 16 bits are not swapped. 1: 16 bits are swapped. Bit 2 0: 32 bits are not swapped. 1: 32 bits are swapped. When eight bits are put together, they are swapped as follows. Each of (1) to (8) indicates eight-bit data. 000: (1) (2) (3) (4) (5) (6) (7) (8) [No swap] 001: (2) (1) (4) (3) (6) (5) (8) (7) [8-bit swap] 010: (3) (4) (1) (2) (7) (8) (5) (6) [16-bit swap] 011: (4) (3) (2) (1) (8) (7) (6) (5) [16-bit swap + 8-bit swap] 100: (5) (6) (7) (8) (1) (2) (3) (4) [32-bit swap] 101: (6) (5) (8) (7) (2) (1) (4) (3) [32-bit swap + 8-bit swap] 110: (7) (8) (5) (6) (3) (4) (1) (2) [32-bit swap + 16-bit swap] 111: (8) (7) (6) (5) (4) (3) (2) (1) [32-bit swap + 16-bit swap + 8-bit swap]
9 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
5 to 0	GR3_STA_POS[5:0]	0	R/W	Sets the amount of data to be skipped through. Specifically data amount equal to the amount indicated by GR3_STA_POS is skipped from the start of the line.

Note: GR3_STA_POS is updated when GR3_P_VEN in GR3_UPDATE is 1. GR3_RDSWA is updated when GR3_UPDATE in GR3_UPDATE is 1.
GR3_FORMAT and GR3_HW are updated when GR3_IBUS_VEN and GR3_P_VEN in GR3_UPDATE are 1.

33.2.31 Alpha Blending Control Register 1 (Graphics 3) (GR3_AB1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR3_ARC_MUL	GR3_ACALC_MD	—	GR3_ARC_ON	—	—	—	GR3_ARC_DISP_ON	—	—	—	GR3_GRC_DISP_ON	—	—	GR3_DISP_SEL[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15	GR3_ARC_MUL	0	R/W	Turns on/off multiplication processing with current alpha at alpha blending in a rectangular area. 0: Off 1: On
14	GR3_ACALC_MD	0	R/W	Turns on/off premultiplication processing at alpha blending in one-pixel units. 0: Off 1: On
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	GR3_ARC_ON	0	R/W	Turns on/off alpha blending in a rectangular area. 0: Off 1: On
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	GR3_ARC_DISP_ON	0	R/W	Turns on/off frame-line display of the image area for alpha blending in a rectangular area. 0: Frame-line display off 1: Frame-line display on
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	GR3_GRC_DISP_ON	0	R/W	Turns on/off frame-line display of the graphics image area. 0: Frame-line display off 1: Frame-line display on
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	GR3_DISP_SEL[1:0]	0	R/W	Selects the graphics display mode. 0: Background color display 1: Lower-layer graphics display 2: Current graphics display 3: Blended display of lower-layer graphics and current graphics

Note: This register is updated when GR3_P_VEN in GR3_UPDATE is 1.

chanhvinh.com

33.2.32 Alpha Blending Control Register 2 (Graphics 3) (GR3_AB2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR3_GRC_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR3_GRC_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR3_GRC_VS[10:0]	0	R/W	Sets the vertical start position of the graphics image area. Note: Set to 4 or greater lines and the result of GR3_GRC_VS + GR3_GRC_VW should be smaller than or equal to 2039 lines.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR3_GRC_VW[10:0]	0	R/W	Sets the vertical width of the graphics image area.

Note: This register is updated when GR3_P_VEN in GR3_UPDATE is 1.

33.2.33 Alpha Blending Control Register 3 (Graphics 3) (GR3_AB3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR3_GRC_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR3_GRC_HW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR3_GRC_HS[10:0]	0	R/W	Sets the horizontal start position of the graphics image area. Note: Set to 16 or greater clocks and the result of GR3_GRC_HS + GR3_GRC_HW should be smaller than or equal to 2015 clocks.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR3_GRC_HW[10:0]	0	R/W	Sets the horizontal width of the graphics image area. Note: For displaying an image with 1- or 2-pixel horizontal width, set GR3_HW to 2 and GR3_GRC_HW to 1 (1-pixel) or 2 (2-pixel).

Note: All the bits assigned to this address are updated when GR3_P_VEN in GR3_UPDATE is 1.

chanhvinh.com

33.2.34 Alpha Blending Control Register 4 (Graphics 3) (GR3_AB4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR3_ARC_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR3_ARC_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR3_ARC_VS[10:0]	0	R/W	Sets the vertical start position of the valid image area for alpha blending in a rectangular area.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR3_ARC_VW[10:0]	0	R/W	Sets the vertical width of the valid image area for alpha blending in a rectangular area.

Note: This register is updated when GR3_P_VEN in GR3_UPDATE is 1.

33.2.35 Alpha Blending Control Register 5 (Graphics 3) (GR3_AB5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR3_ARC_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR3_ARC_HW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR3_ARC_HS[10:0]	0	R/W	Sets the horizontal start position of the valid image area for alpha blending in a rectangular area.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR3_ARC_HW[10:0]	0	R/W	Sets the horizontal width of the valid image area for alpha blending in a rectangular area.

Note: This register is updated when GR3_P_VEN in GR3_UPDATE is 1.

chanhvinh.com

33.2.36 Alpha Blending Control Register 6 (Graphics 3) (GR3_AB6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	GR3_ARC_MODE	GR3_ARC_COEF[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	GR3_ARC_RATE[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	GR3_ARC_MODE	0	R/W	Alpha Blending Mode in Rectangular Area 0: Addition 1: Subtraction
23 to 16	GR3_ARC_COEF[7:0]	0	R/W	Sets the alpha coefficient for alpha blending in a rectangular area. (0 to 255) [7:0]: Variation (absolute value)
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	GR3_ARC_RATE[7:0]	0	R/W	Sets the value obtained by subtracting 1 from the frame rate for alpha blending in a rectangular area.

Note: This register is updated when GR3_P_VEN in GR3_UPDATE is 1.

33.2.37 Alpha Blending Control Register 7 (Graphics 3) (GR3_AB7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	GR3_ARC_DEF[7:0]							
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR3_CK_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	GR3_ARC_DEF[7:0]	255	R/W	Sets the initial alpha value for alpha blending in a rectangular area.
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR3_CK_ON	0	R/W	Turns on/off CLUT-index/RGB-index chroma-key processing. 0: Off 1: On

Note: This register is updated when GR3_P_VEN in GR3_UPDATE is 1.

chanhvinh.com

33.2.38 Alpha Blending Control Register 8 (Graphics 3) (GR3_AB8)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR3_CK_KCLUT[7:0]								GR3_CK_KG[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR3_CK_KB[7:0]								GR3_CK_KR[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR3_CK_KCLUT[7:0]	0	R/W	CLUT Signal for CLUT-Index Chroma-Key Processing CLUT: Unsigned 8 bits (0 to 255 [LSB])
23 to 16	GR3_CK_KG[7:0]	0	R/W	G Signal for RGB-Index Chroma-Key Processing G: Unsigned 8 bits (0 to 255 [LSB])
15 to 8	GR3_CK_KB[7:0]	0	R/W	B Signal for RGB-Index Chroma-Key Processing B: Unsigned 8 bits (0 to 255 [LSB])
7 to 0	GR3_CK_KR[7:0]	0	R/W	R Signal for RGB-Index Chroma-Key Processing R: Unsigned 8 bits (0 to 255 [LSB])

Note: This register is updated when GR3_P_VEN in GR3_UPDATE is 1.

33.2.39 Alpha Blending Control Register 9 (Graphics 3) (GR3_AB9)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR3_CK_A[7:0]								GR3_CK_G[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR3_CK_B[7:0]								GR3_CK_R[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR3_CK_A[7:0]	0	R/W	Replaced Alpha Signal after RGB/CLUT-Index Chroma-Key Processing α: Unsigned 8 bits (0 to 255 [LSB])
23 to 16	GR3_CK_G[7:0]	0	R/W	Replaced G Signal after RGB/CLUT-Index Chroma-Key Processing G: Unsigned 8 bits (0 to 255 [LSB])
15 to 8	GR3_CK_B[7:0]	0	R/W	Replaced B Signal after RGB/CLUT-Index Chroma-Key Processing B: Unsigned 8 bits (0 to 255 [LSB])
7 to 0	GR3_CK_R[7:0]	0	R/W	Replaced R Signal after RGB/CLUT-Index Chroma-Key Processing R: Unsigned 8 bits (0 to 255 [LSB])

Note: This register is updated when GR3_P_VEN in GR3_UPDATE is 1.

chanhvinh.com

33.2.40 Alpha Blending Control Register 10 (Graphics 3) (GR3_AB10)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR3_A0[7:0]								GR3_G0[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR3_B0[7:0]								GR3_R0[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR3_A0 [7:0]	0	R/W	CLUT1 α0 Signal Replaced with α signal when in the CLUT1 format and CLUT1 = 0. Replaced with α signal when in the αRGB1555 or RGBα5551 format and α = 0.
23 to 16	GR3_G0 [7:0]	0	R/W	CLUT1 G0 Signal Replaced with G signal when in the CLUT1 format and CLUT1 = 0.
15 to 8	GR3_B0 [7:0]	0	R/W	CLUT1 B0 Signal Replaced with B signal when in the CLUT1 format and CLUT1 = 0.
7 to 0	GR3_R0 [7:0]	0	R/W	CLUT1 R0 Signal Replaced with R signal when in the CLUT1 format and CLUT1 = 0.

Note: This register is updated when GR3_P_VEN in GR3_UPDATE is 1.

33.2.41 Alpha Blending Control Register 11 (Graphics 3) (GR3_AB11)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR3_A1[7:0]								GR3_G1[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR3_B1[7:0]								GR3_R1[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR3_A1 [7:0]	0	R/W	CLUT1 α1 Signal Replaced with α signal when in the CLUT1 format and CLUT1 = 1. Replaced with α signal when in the αRGB1555 or RGBα5551 format and α = 1.
23 to 16	GR3_G1 [7:0]	0	R/W	CLUT1 G1 Signal Replaced with G signal when in the CLUT1 format and CLUT1 = 1.
15 to 8	GR3_B1 [7:0]	0	R/W	CLUT1 B1 Signal Replaced with B signal when in the CLUT1 format and CLUT1 = 1.
7 to 0	GR3_R1 [7:0]	0	R/W	CLUT1 R1 Signal Replaced with R signal when in the CLUT1 format and CLUT1 = 1.

Note: This register is updated when GR3_P_VEN in GR3_UPDATE is 1.

chanhvinh.com

33.2.42 Background Color Control Register (Graphics 3) (GR3_BASE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	GR3_BASE_G[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR3_BASE_B[7:0]								GR3_BASE_R[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	GR3_BASE_G [7:0]	0	R/W	Background Color G Signal G: Unsigned 8 bits (0 to 255 [LSB])
15 to 8	GR3_BASE_B [7:0]	0	R/W	Background Color B Signal B: Unsigned 8 bits (0 to 255 [LSB])
7 to 0	GR3_BASE_R [7:0]	0	R/W	Background Color R Signal R: Unsigned 8 bits (0 to 255 [LSB])

Note: This register is updated when GR3_P_VEN in GR3_UPDATE is 1.

33.2.43 CLUT Table and Interrupt Control Register (Graphics 3) (GR3_CLUT_INT)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR3_CLT_SEL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR3_LINE[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	GR3_CLT_ SEL	0	R/W	CLUT Table Select Signal 0: Selects CLUT table 0. The format is converted to αRGB8888 based on the CLUT table 0. CLUT table 1 can be read from or written to by the CPU. 1: Selects CLUT table 1. The format is converted to αRGB8888 based on the CLUT table 1. CLUT table 0 can be read from or written to by the CPU.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
10 to 0	GR3_LINE [10:0]	0	R/W	Line Interrupt Set When number of lines matches the value of the GR3_LINE bits, an interrupt signal is output. This function is enabled even when the graphics 3 process is not used.

Note: This register is updated when GR3_P_VEN in GR3_UPDATE is 1.

33.2.44 Status Monitor Register (Graphics 3) (GR3_MON)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR3_LIN_STAT[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR3_ARC_ST
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR3_LIN_ STAT[10:0]	0	R	Line Position of Image being Currently Read
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR3_ARC_ ST	0	R	Status Flag for Alpha Blending in Rectangular Area 0: Addition or subtraction has been completed. (α value is 0 or 255) 1: Addition or subtraction is in progress.

chanhvinh.com

33.2.45 VIN Synthesizer Register Update Control Register (GR_VIN_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GR_VIN_UPDATE	—	—	—	GR_VIN_P_VEN	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/WC1	R	R	R	R/WC1	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	GR_VIN_UPDATE	0	R/WC1	Graphics Display Register Update 0: Registers are not updated. 1: Registers are updated.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	GR_VIN_P_VEN	0	R/WC1	Graphics Display Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

33.2.46 Alpha Blending Control Register 1 (VIN Synthesizer) (GR_VIN_AB1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	GR_VIN_SCL_UND_SEL	GR_VIN_DISP_SEL[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	GR_VIN_SCL_UND_SEL	0	R/W	Selection of Lower-Layer Plane in Scaler 0: Selects graphics 0 as lower-layer graphics. 1: Setting prohibited
1, 0	GR_VIN_DISP_SEL[1:0]	0	R/W	Selects the graphics display mode. 0: Background color display (fixed to black) 1: Lower-layer graphics display 2: Setting prohibited 3: Setting prohibited

Notes: GR_VIN_SCL_UND_SEL is updated when GR_VIN_UPDATE in GR_VIN_UPDATE is 1. The other bits of this register are updated when GR_VIN_P_VEN in GR_VIN_UPDATE is 1.

chanhvinh.com

33.3 Usage Method

33.3.1 Mute Image

The initial values of the GR0_DISP_SEL[1:0], GR2_DISP_SEL[1:0], GR3_DISP_SEL[1:0], and GR_VIN_DISP_SEL[1:0] bits are all 0. Accordingly, in the initial setting, a background color is displayed both inside and outside the graphics area for the graphics 0, 2, and 3 processes and the VIN synthesizer. Since the default background color is black, the black mute image is displayed in the initial state. Note that the background color for the VIN synthesizer is fixed to black.

33.3.2 Alpha Blending in Rectangular Area

The alpha coefficient and the frame rate can be changed during fade in and fade out by modifying the GR_ARC_MODE, GR_ARC_COEF[7:0] and GR_ARC_RATE[7:0] bits, respectively.

chanhvinh.com

34. Video Display Controller 5 (7): Output Controller

34.1 Output Controller

34.1.1 Overview of Functions

The output controller subjects RGB signals output from the image synthesizer to brightness adjustment, contrast adjustment, gamma correction of individual RGB, dither process, and output format conversion.

Figure 34.1 shows the function block diagram of the output controller.

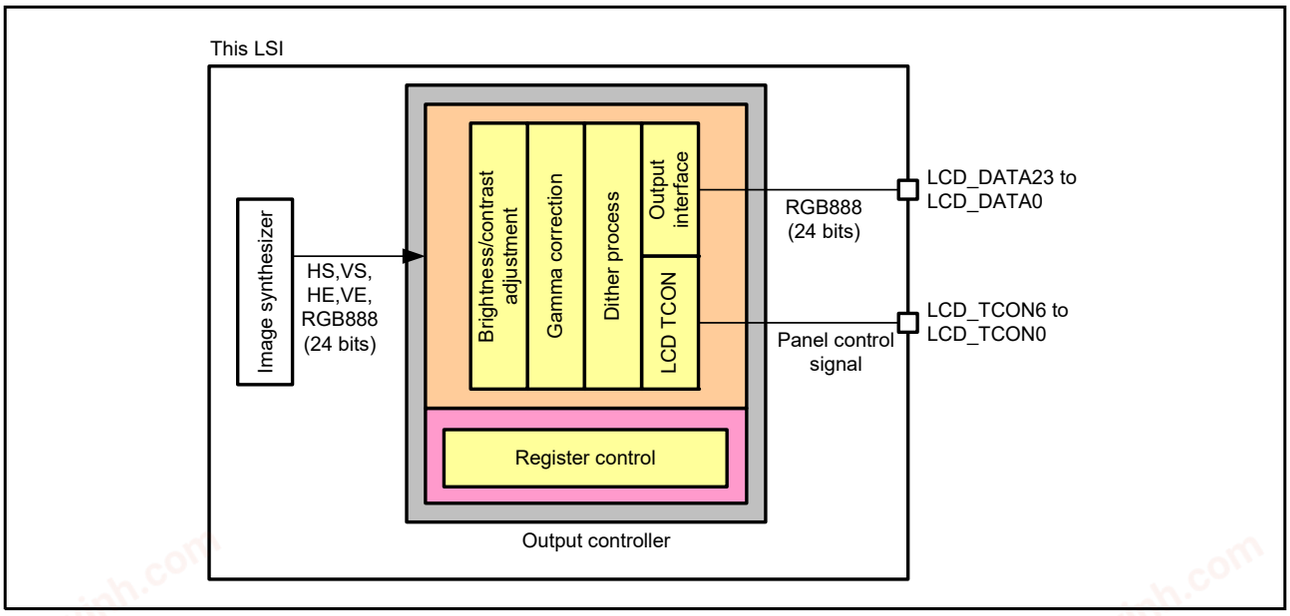


Figure 34.1 Functional Block Diagram of Output Controller

chanhvinh.com

34.1.2 Register Update Control

The Vsync signal is used to control the update timing of all the registers of the output controller.

After 1 is set to the bits in the update control register, the contents of the relevant registers are actually modified at the rising edge of the Vsync signal, when the update control register is automatically cleared to 0.

Table 34.1 Register Update Control

Register Name	Bit Name	Initial Value	Description
OUT_UPDATE	OUTCNT_VEN	0	Brightness/Contrast Control, Dither Process, Output Interface Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
GAM_G_UPDATE	GAM_G_VEN	0	Gamma Correction (G) Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
GAM_B_UPDATE	GAM_B_VEN	0	Gamma Correction (B) Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
GAM_R_UPDATE	GAM_R_VEN	0	Gamma Correction (R) Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
TCON_UPDATE	TCON_VEN	0	LCD TCON Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

34.1.3 Route Selection

The processing sequence of the brightness/contrast control and gamma correction control can be swapped according to the settings of the register.

Table 34.2 Route Selection

Register Name	Bit Name	Initial Value	Description
OUT_CLK_PHASE	OUTCNT_ FRONT_GAM	0	Correction Circuit Sequence Control 0: Brightness → contrast → gamma correction 1: Gamma correction → brightness → contrast

chanhvinh.com

34.1.4 Panel Brightness Adjustment

Brightness (DC) adjustment is individually performed for RGB signals from the image synthesizer.

(BRT_R/G/BOUT after brightness adjustment has many bits to prevent overflow or underflow. The overflow or underflow process is performed at contrast calculation.)

(1) Calculation formulas for brightness (DC) adjustment

BRT_GOUT = GIN + PBRT_G − 512

BRT_BOUT = BIN + PBRT_B − 512

BRT_ROUT = RIN + PBRT_R − 512

Table 34.3 Brightness (DC) Adjustment

Register Name	Bit Name	Initial Value	Description
OUT_BRIGHT1	PBRT_G[9:0]	512	Brightness (DC) Adjustment of G Signal Unsigned (0 (-512) to 512 (0) to 1023 (+511) [LSB], 512 [LSB] with offset)
OUT_BRIGHT2	PBRT_B[9:0]	512	Brightness (DC) Adjustment of B Signal Unsigned (0 (-512) to 512 (0) to 1023 (+511) [LSB], 512 [LSB] with offset)
OUT_BRIGHT2	PBRT_R[9:0]	512	Brightness (DC) Adjustment of R Signal Unsigned (0 (-512) to 512 (0) to 1023 (+511) [LSB], 512 [LSB] with offset)

34.1.5 Contrast Adjustment

Contrast is calculated for RGB signals obtained after brightness calculation.

(If an overflow or underflow occurs, contrast is clipped to the maximum or minimum value.)

(1) Calculation formulas for contrast (gain) adjustment

GOUT = BRT_GOUT × CONT_G/128

BOUT = BRT_BOUT × CONT_B/128

ROUT = BRT_ROUT × CONT_R/128

Table 34.4 Contrast (Gain) Adjustment

Register Name	Bit Name	Initial Value	Description
OUT_CONTRAST	CONT_G [7:0]	128	Contrast (Gain) Adjustment of G Signal 0/128 to 255/128 (approx.2 times)
OUT_CONTRAST	CONT_B [7:0]	128	Contrast (Gain) Adjustment of B Signal 0/128 to 255/128 (approx.2 times)
OUT_CONTRAST	CONT_R [7:0]	128	Contrast (Gain) Adjustment of R Signal 0/128 to 255/128 (approx.2 times)

chanhvinh.com

34.1.6 Gamma Correction

Gamma correction is carried out by dividing an input signal having 256 gradation levels into 32 and controlling the gain of each area. Gain coefficient of each area can be set as 0 to approx. 2.0 [times]

(1) Gamma correction formula for each area

$$DOUT = ((DIN - TH_{(n)}) \times GAIN_{(n)} + OFFSET_{(n)})/256$$

DIN: Input signal (8-bit)

DOUT: Output signal (10-bit)

TH(n): Threshold (8-bit)

OFFSET(n): Offset value (19-bit)

GAIN(n): Gain coefficient (11-bit)

(2) Offset calculation formulas for each area

$$OFFSET_{(n)} = OFFSET_{(n-1)} + DEF_O_{(n)} \text{ (When n = 0, } OFFSET_{(0)} = 0.)$$

$$DEF_O_{(n)} = (TH_{(n)} - TH_{(n-1)}) \times GAIN_{(n-1)} \text{ (When n = 0, } OFFSET_{(0)} = 0.)$$

OFFSET(n): Offset value of current area (19-bit)

OFFSET(n-1): Offset value of previous area (19-bit)

DEF_O(n): Difference in offset value of Current and previous area (19-bit)

TH(n): Threshold of current area (8-bit)

TH(n-1): Threshold of previous area (8-bit)

GAIN(n-1): Gain coefficient of previous area (11-bit)

chanhvinh.com

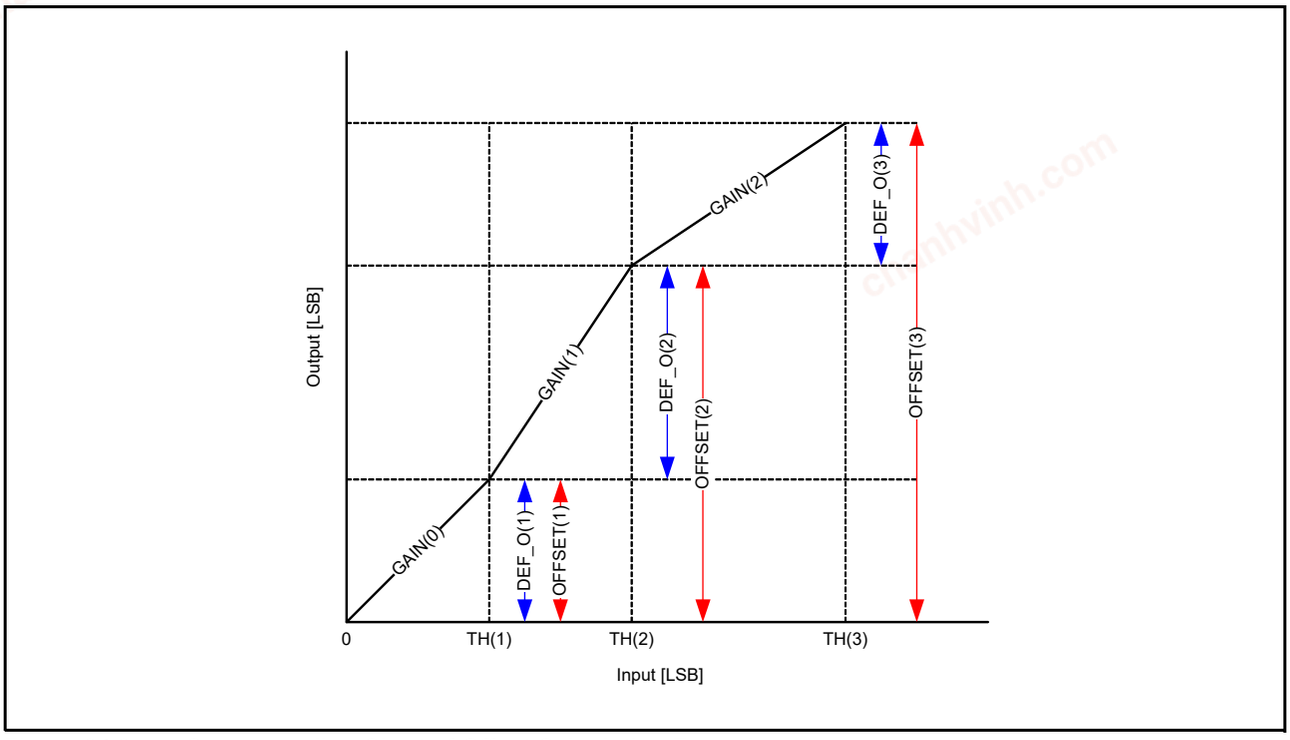


Figure 34.2 Corresponding Chart of Offset Calculation Formulas

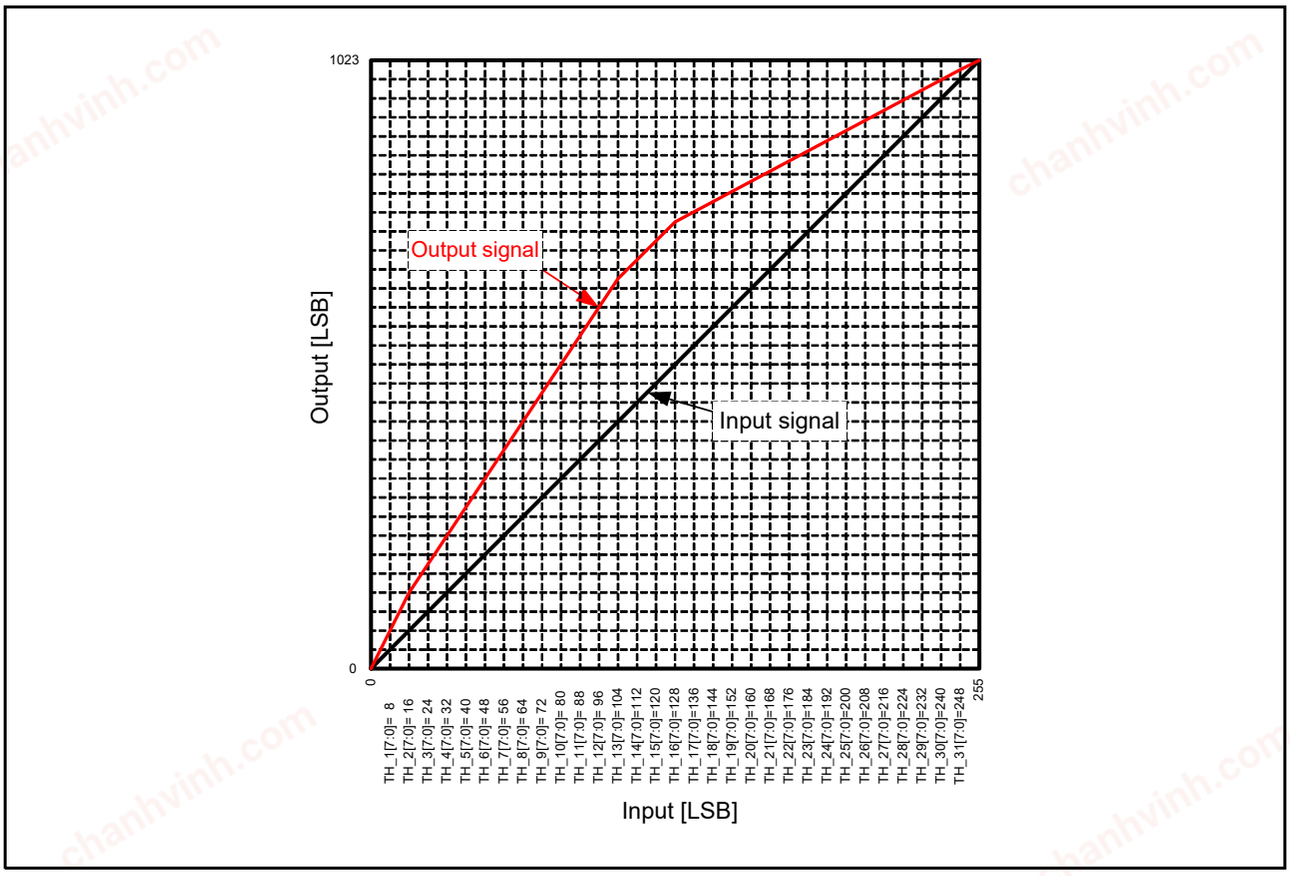


Figure 34.3 Example of Input-Output Characteristics of Gamma Correction

chanhvinh.com

Table 34.5 Gamma Correction

Register Name	Bit Name	Initial Value	Description
GAM_SW	GAM_ON	0	Gamma Correction On/Off Control 0: Off 1: On
GAM_G_AREA1 to GAM_G_AREA8	GAM_G_TH_01 to GAM_G_TH_31 [7:0]	*	Start Threshold of Area 1 to 31 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area*1 < Threshold of current area < Threshold of next area*2 *1: GAM_G_TH_01 is 0 *2: GAM_G_TH_31 is ≤ 255 *Initial Value GAM_G_TH_01:8, GAM_G_TH_02:16, GAM_G_TH_03:24, GAM_G_TH_04:32, GAM_G_TH_05:40, GAM_G_TH_06:48, GAM_G_TH_07:56, GAM_G_TH_08:64, GAM_G_TH_09:72, GAM_G_TH_10:80 GAM_G_TH_11:88, GAM_G_TH_12:96, GAM_G_TH_13:104, GAM_G_TH_14:112, GAM_G_TH_15:120, GAM_G_TH_16:128, GAM_G_TH_17:136, GAM_G_TH_18:144, GAM_G_TH_19:152, GAM_G_TH_20:160, GAM_G_TH_21:168, GAM_G_TH_22:176, GAM_G_TH_23:184, GAM_G_TH_24:192, GAM_G_TH_25:200, GAM_G_TH_26:208, GAM_G_TH_27:216, GAM_G_TH_28:224, GAM_G_TH_29:232, GAM_G_TH_30:240, GAM_G_TH_31:248
GAM_G_LUT1 to GAM_G_LUT16	GAM_G_GAIN_00 to GAM_G_GAIN_31 [10:0]	1024	Gain Adjustment of Area 0 to 31 of G Signal Unsigned (0 to 2047 [LSB], 1024 [LSB] = 1.0 [times])
GAM_B_AREA1 to GAM_B_AREA8	GAM_B_TH_01 to GAM_B_TH_31 [7:0]	*	Start Threshold of Area 1 to 31 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area*1 < Threshold of current area < Threshold of next area*2 *1: GAM_B_TH_01 is 0 *2: GAM_B_TH_31 is ≤ 255 *Initial Value GAM_B_TH_01:8, GAM_B_TH_02:16, GAM_B_TH_03:24, GAM_B_TH_04:32, GAM_B_TH_05:40, GAM_B_TH_06:48, GAM_B_TH_07:56, GAM_B_TH_08:64, GAM_B_TH_09:72, GAM_B_TH_10:80 GAM_B_TH_11:88, GAM_B_TH_12:96, GAM_B_TH_13:104, GAM_B_TH_14:112, GAM_B_TH_15:120, GAM_B_TH_16:128, GAM_B_TH_17:136, GAM_B_TH_18:144, GAM_B_TH_19:152, GAM_B_TH_20:160, GAM_B_TH_21:168, GAM_B_TH_22:176, GAM_B_TH_23:184, GAM_B_TH_24:192, GAM_B_TH_25:200, GAM_B_TH_26:208, GAM_B_TH_27:216, GAM_B_TH_28:224, GAM_B_TH_29:232, GAM_B_TH_30:240, GAM_B_TH_31:248
GAM_B_LUT1 to GAM_B_LUT16	GAM_B_GAIN_00 to GAM_B_GAIN_31 [10:0]	1024	Gain Adjustment of Area 0 to 31 of B Signal Unsigned (0 to 2047 [LSB], 1024 [LSB] = 1.0 [times])

chanhvinh.com

Table 34.5 Gamma Correction

Register Name	Bit Name	Initial Value	Description
GAM_R_AREA1 to GAM_R_AREA8	GAM_R_TH_01 to GAM_R_TH_31 [7:0]	*	Start Threshold of Area 1 to 31 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area*1 < Threshold of current area < Threshold of next area*2 *1: GAM_R_TH_01 is 0 *2: GAM_R_TH_31 is ≤ 255 *Initial Value GAM_R_TH_01:8, GAM_R_TH_02:16, GAM_R_TH_03:24, GAM_R_TH_04:32, GAM_R_TH_05:40, GAM_R_TH_06:48, GAM_R_TH_07:56, GAM_R_TH_08:64, GAM_R_TH_09:72, GAM_R_TH_10:80 GAM_R_TH_11:88, GAM_R_TH_12:96, GAM_R_TH_13:104, GAM_R_TH_14:112, GAM_R_TH_15:120, GAM_R_TH_16:128, GAM_R_TH_17:136, GAM_R_TH_18:144, GAM_R_TH_19:152, GAM_R_TH_20:160, GAM_R_TH_21:168, GAM_R_TH_22:176, GAM_R_TH_23:184, GAM_R_TH_24:192, GAM_R_TH_25:200, GAM_R_TH_26:208, GAM_R_TH_27:216, GAM_R_TH_28:224, GAM_R_TH_29:232, GAM_R_TH_30:240, GAM_R_TH_31:248
GAM_R_LUT1 to GAM_R_LUT16	GAM_R_GAIN_00 to GAM_R_GAIN_31[10:0]	1024	Gain Adjustment of Area 0 to 31 of R Signal Unsigned (0 to 2047 [LSB], 1024 [LSB] = 1.0 [times])

chanhvinh.com

34.1.7 Dither Process

Dither process is carried out by adjusting brightness/contrast or reducing 10-bit RGB signals output from the gamma correction block to 8-bit, 6-bit, or 5-bit RGB signals. The operation mode of dither process can be selected from truncate mode, round-off mode, 2 × 2 pattern dither mode and random pattern dither mode.

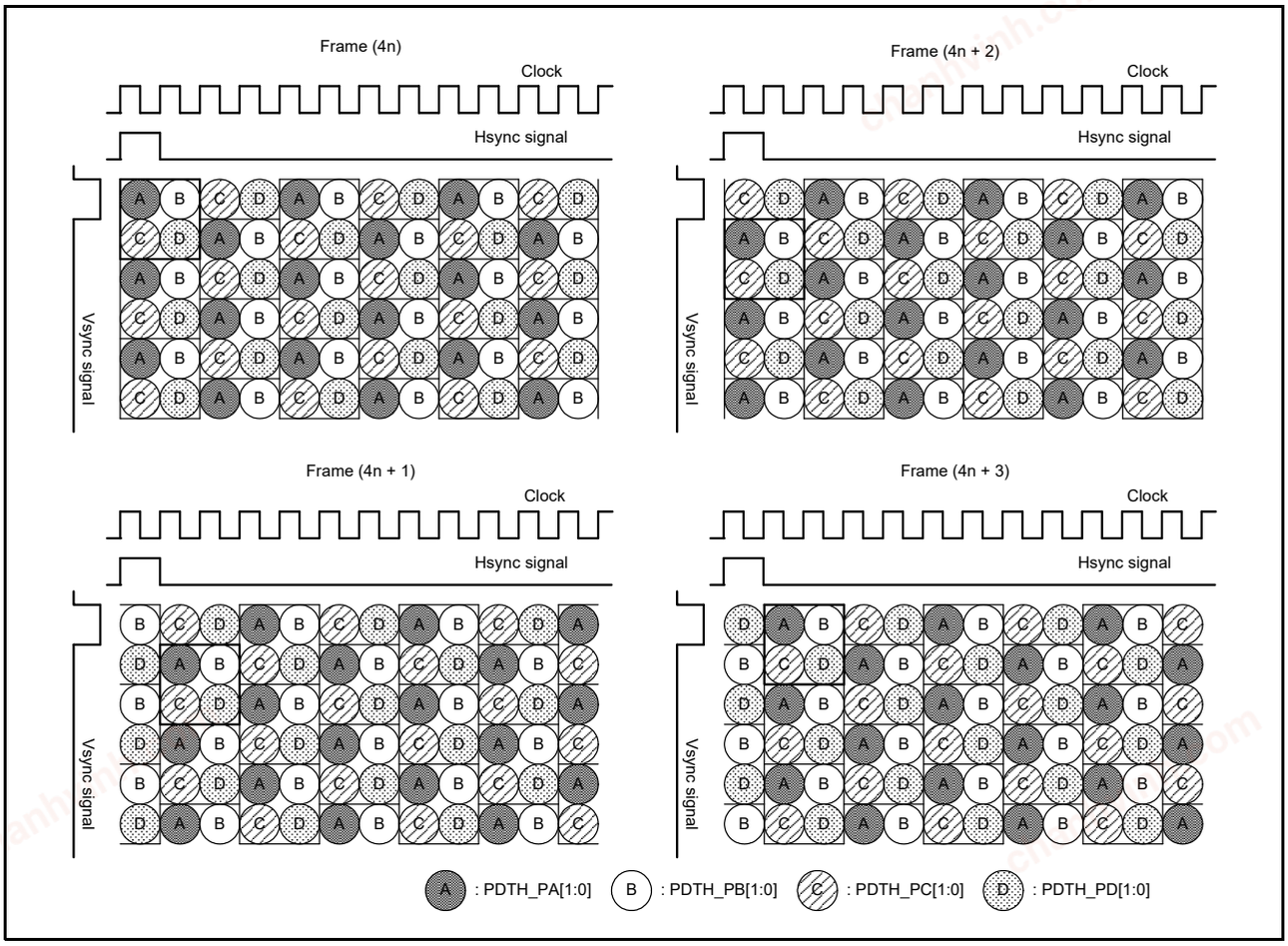


Figure 34.4 Operation Specification of 2 × 2 Pattern Dither

The conversion equations are as follows.

[Truncate mode]

(a) 10 bits to 8 bits

Output RGB data[7:0] = Input RGB data[9:0] ÷ 4 (truncate the number below the decimal point)

(b) 10 bits to 6 bits

Output RGB data[7:2] = Input RGB data[9:0] ÷ 16 (truncate the number below the decimal point)

(c) 10 bits to 5 bits

Output RGB data[7:3] = Input RGB data[9:0] ÷ 32 (truncate the number below the decimal point)

chanhvinh.com

[Round-off mode]

(a) 10 bits to 8 bits

Output RGB data[7:0] = Input RGB data[9:0] ÷ 4 (round off to an integer)

(b) 10 bits to 6 bits

Output RGB data[7:2] = Input RGB data[9:0] ÷ 16 (round off to an integer)

(c) 10 bits to 5 bits

Output RGB data[7:3] = Input RGB data[9:0] ÷ 32 (round off to an integer)

[2 × 2 pattern dither mode, random pattern dither mode]

(a) 10 bits to 8 bits

Output RGB data[7:0] = Input RGB data[9:0] ÷ 4 + pattern value at the first decimal place (truncate the number below the decimal point after addition)

(b) 10 bits to 6 bits

Output RGB data[7:2] = Input RGB data[9:0] ÷ 16 + pattern value at the first decimal place (truncate the number below the decimal point after addition)

(c) 10 bits to 5 bits

Output RGB data[7:3] = Input RGB data[9:0] ÷ 32 + pattern value at the first decimal place (truncate the number below the decimal point after addition)

Table 34.6 Panel Dither Correction

Register Name	Bit Name	Initial Value	Description
OUT_PDTHA	PDTH_SEL[1:0]	0	Panel Dither Operation Mode 0: Truncate 1: Round-off 2: 2 × 2 pattern dither 3: Random pattern dither
OUT_PDTHA	PDTH_FORMAT[1:0]	0	Panel Dither Output Format Select 0: RGB888 1: RGB666 2: RGB565 3: Setting prohibited
OUT_PDTHA	PDTH_PA[1:0]	3	Pattern Value (A) of 2 × 2 Pattern Dither Unsigned (0 to 3 [LSB])
OUT_PDTHA	PDTH_PB[1:0]	0	Pattern Value (B) of 2 × 2 Pattern Dither Unsigned (0 to 3 [LSB])
OUT_PDTHA	PDTH_PC[1:0]	2	Pattern Value (C) of 2 × 2 Pattern Dither Unsigned (0 to 3 [LSB])
OUT_PDTHA	PDTH_PD[1:0]	1	Pattern Value (D) of 2 × 2 Pattern Dither Unsigned (0 to 3 [LSB])

chanhvinh.com

34.1.8 Output Format Conversion

In output format conversion, the RGB signal after dither process is converted to LCD output signal having any of the following formats, namely, parallel RGB888, parallel RGB666, parallel RGB565, and serial RGB.

Further, converted data can be allocated to LCD output pins as selected.

(1) Bit Allocation of LCD Signals for RGB888 Output

Table 34.7 shows the RGB signal input allocated to the LCD signal output for RGB888 output.

R/G/BIN[7:0] are the RGB internal signals after dither process.

Table 34.7 Bit Allocation of RGB Signal Input for RGB888 Output

OUT_FORMAT	**0**	**0**	**0**	**0**
OUT_ENDIAN_ON	**0**	**0**	**1**	**1**
OUT_SWAP_ON	**0**	**1**	**0**	**1**
LCD_DATA23	RIN[7]	BIN[7]	RIN[0]	BIN[0]
LCD_DATA22	RIN[6]	BIN[6]	RIN[1]	BIN[1]
LCD_DATA21	RIN[5]	BIN[5]	RIN[2]	BIN[2]
LCD_DATA20	RIN[4]	BIN[4]	RIN[3]	BIN[3]
LCD_DATA19	RIN[3]	BIN[3]	RIN[4]	BIN[4]
LCD_DATA18	RIN[2]	BIN[2]	RIN[5]	BIN[5]
LCD_DATA17	RIN[1]	BIN[1]	RIN[6]	BIN[6]
LCD_DATA16	RIN[0]	BIN[0]	RIN[7]	BIN[7]
LCD_DATA15	GIN[7]	GIN[7]	GIN[0]	GIN[0]
LCD_DATA14	GIN[6]	GIN[6]	GIN[1]	GIN[1]
LCD_DATA13	GIN[5]	GIN[5]	GIN[2]	GIN[2]
LCD_DATA12	GIN[4]	GIN[4]	GIN[3]	GIN[3]
LCD_DATA11	GIN[3]	GIN[3]	GIN[4]	GIN[4]
LCD_DATA10	GIN[2]	GIN[2]	GIN[5]	GIN[5]
LCD_DATA9	GIN[1]	GIN[1]	GIN[6]	GIN[6]
LCD_DATA8	GIN[0]	GIN[0]	GIN[7]	GIN[7]
LCD_DATA7	BIN[7]	RIN[7]	BIN[0]	RIN[0]
LCD_DATA6	BIN[6]	RIN[6]	BIN[1]	RIN[1]
LCD_DATA5	BIN[5]	RIN[5]	BIN[2]	RIN[2]
LCD_DATA4	BIN[4]	RIN[4]	BIN[3]	RIN[3]
LCD_DATA3	BIN[3]	RIN[3]	BIN[4]	RIN[4]
LCD_DATA2	BIN[2]	RIN[2]	BIN[5]	RIN[5]
LCD_DATA1	BIN[1]	RIN[1]	BIN[6]	RIN[6]
LCD_DATA0	BIN[0]	RIN[0]	BIN[7]	RIN[7]

chanhvinh.com

(2) Bit Allocation of LCD Signal for RGB666 Output

Table 34.8 shows the RGB signal input allocated to the LCD signal output for RGB666 output.

R/G/BIN[7:0] are the RGB internal signals after dither process.

Table 34.8 Bit Allocation of RGB Signal Input for RGB666 Output

OUT_FORMAT	**1**	**1**	**1**	**1**
OUT_ENDIAN_ON	**0**	**0**	**1**	**1**
OUT_SWAP_ON	**0**	**1**	**0**	**1**
LCD_DATA23	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA22	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA21	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA20	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA19	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA18	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA17	RIN[7]	BIN[7]	RIN[2]	BIN[2]
LCD_DATA16	RIN[6]	BIN[6]	RIN[3]	BIN[3]
LCD_DATA15	RIN[5]	BIN[5]	RIN[4]	BIN[4]
LCD_DATA14	RIN[4]	BIN[4]	RIN[5]	BIN[5]
LCD_DATA13	RIN[3]	BIN[3]	RIN[6]	BIN[6]
LCD_DATA12	RIN[2]	BIN[2]	RIN[7]	BIN[7]
LCD_DATA11	GIN[7]	GIN[7]	GIN[2]	GIN[2]
LCD_DATA10	GIN[6]	GIN[6]	GIN[3]	GIN[3]
LCD_DATA9	GIN[5]	GIN[5]	GIN[4]	GIN[4]
LCD_DATA8	GIN[4]	GIN[4]	GIN[5]	GIN[5]
LCD_DATA7	GIN[3]	GIN[3]	GIN[6]	GIN[6]
LCD_DATA6	GIN[2]	GIN[2]	GIN[7]	GIN[7]
LCD_DATA5	BIN[7]	RIN[7]	BIN[2]	RIN[2]
LCD_DATA4	BIN[6]	RIN[6]	BIN[3]	RIN[3]
LCD_DATA3	BIN[5]	RIN[5]	BIN[4]	RIN[4]
LCD_DATA2	BIN[4]	RIN[4]	BIN[5]	RIN[5]
LCD_DATA1	BIN[3]	RIN[3]	BIN[6]	RIN[6]
LCD_DATA0	BIN[2]	RIN[2]	BIN[7]	RIN[7]

chanhvinh.com

(3) Bit Allocation of LCD Signal for RGB565 Output

Table 34.9 shows the RGB signal input allocated to the LCD signal output for RGB565 output.

R/G/BIN[7:0] are the RGB internal signals after dither process.

Table 34.9 Bit Allocation of RGB Signal Input for RGB565 Output

OUT_FORMAT	**2**	**2**	**2**	**2**
OUT_ENDIAN_ON	**0**	**0**	**1**	**1**
OUT_SWAP_ON	**0**	**1**	**0**	**1**
LCD_DATA23	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA22	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA21	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA20	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA19	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA18	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA17	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA16	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA15	RIN[7]	BIN[7]	RIN[3]	BIN[3]
LCD_DATA14	RIN[6]	BIN[6]	RIN[4]	BIN[4]
LCD_DATA13	RIN[5]	BIN[5]	RIN[5]	BIN[5]
LCD_DATA12	RIN[4]	BIN[4]	RIN[6]	BIN[6]
LCD_DATA11	RIN[3]	BIN[3]	RIN[7]	BIN[7]
LCD_DATA10	GIN[7]	GIN[7]	GIN[2]	GIN[2]
LCD_DATA9	GIN[6]	GIN[6]	GIN[3]	GIN[3]
LCD_DATA8	GIN[5]	GIN[5]	GIN[4]	GIN[4]
LCD_DATA7	GIN[4]	GIN[4]	GIN[5]	GIN[5]
LCD_DATA6	GIN[3]	GIN[3]	GIN[6]	GIN[6]
LCD_DATA5	GIN[2]	GIN[2]	GIN[7]	GIN[7]
LCD_DATA4	BIN[7]	RIN[7]	BIN[3]	RIN[3]
LCD_DATA3	BIN[6]	RIN[6]	BIN[4]	RIN[4]
LCD_DATA2	BIN[5]	RIN[5]	BIN[5]	RIN[5]
LCD_DATA1	BIN[4]	RIN[4]	BIN[6]	RIN[6]
LCD_DATA0	BIN[3]	RIN[3]	BIN[7]	RIN[7]

chanhvinh.com

(4) Bit Allocation of LCD Signal for Serial RGB Output

For serial RGB output, RGB signal input shown Table 34.10 is allocated to rgb internal signals and the signals are converted from parallel to serial format and output as LCD signals. R/G/BIN[7:0] are the RGB internal signals after dither process.

The internal signals r[7:0], g[7:0], and b[7:0] are serially output to LCD_DATA7 to LCD_DATA0.

Table 34.10 Bit Allocation of RGB Signal Input for Serial RGB Output

OUT_FORMAT	**3**	**3**	**3**	**3**
OUT_ENDIAN_ON	**0**	**0**	**1**	**1**
OUT_SWAP_ON	**0**	**1**	**0**	**1**
r[7]	RIN[7]	BIN[7]	RIN[0]	BIN[0]
r[6]	RIN[6]	BIN[6]	RIN[1]	BIN[1]
r[5]	RIN[5]	BIN[5]	RIN[2]	BIN[2]
r[4]	RIN[4]	BIN[4]	RIN[3]	BIN[3]
r[3]	RIN[3]	BIN[3]	RIN[4]	BIN[4]
r[2]	RIN[2]	BIN[2]	RIN[5]	BIN[5]
r[1]	RIN[1]	BIN[1]	RIN[6]	BIN[6]
r[0]	RIN[0]	BIN[0]	RIN[7]	BIN[7]
g[7]	GIN[7]	GIN[7]	GIN[0]	GIN[0]
g[6]	GIN[6]	GIN[6]	GIN[1]	GIN[1]
g[5]	GIN[5]	GIN[5]	GIN[2]	GIN[2]
g[4]	GIN[4]	GIN[4]	GIN[3]	GIN[3]
g[3]	GIN[3]	GIN[3]	GIN[4]	GIN[4]
g[2]	GIN[2]	GIN[2]	GIN[5]	GIN[5]
g[1]	GIN[1]	GIN[1]	GIN[6]	GIN[6]
g[0]	GIN[0]	GIN[0]	GIN[7]	GIN[7]
b[7]	BIN[7]	RIN[7]	BIN[0]	RIN[0]
b[6]	BIN[6]	RIN[6]	BIN[1]	RIN[1]
b[5]	BIN[5]	RIN[5]	BIN[2]	RIN[2]
b[4]	BIN[4]	RIN[4]	BIN[3]	RIN[3]
b[3]	BIN[3]	RIN[3]	BIN[4]	RIN[4]
b[2]	BIN[2]	RIN[2]	BIN[5]	RIN[5]
b[1]	BIN[1]	RIN[1]	BIN[6]	RIN[6]
b[0]	BIN[0]	RIN[0]	BIN[7]	RIN[7]

chanhvinh.com

(5) Parallel to Serial Conversion

As shown in Table 34.11, four types of parallel to serial conversions are possible by controlling clock speed mode and selecting the scan direction ('n' in the table are natural numbers).

Table 34.11 Specifications of Serial RGB Output

OUT_FRQ_SEL	1	1	2	2
OUT_DIR_SEL	**0**	**1**	**0**	**1**
Line (2n-1)	Repeated (r → g → b)	Repeated (b → g → r)	Repeated (r → g → b → X)	Repeated (X → b → g → r)
Line 2n	Repeated (g → b → r)	Repeated (r → b → g)	Repeated (r → g → b → X)	Repeated (X → b → g → r)

Figure 34.5 and Figure 34.6 show the timing of parallel to serial conversion in triple speed and quadruple speed modes, respectively.

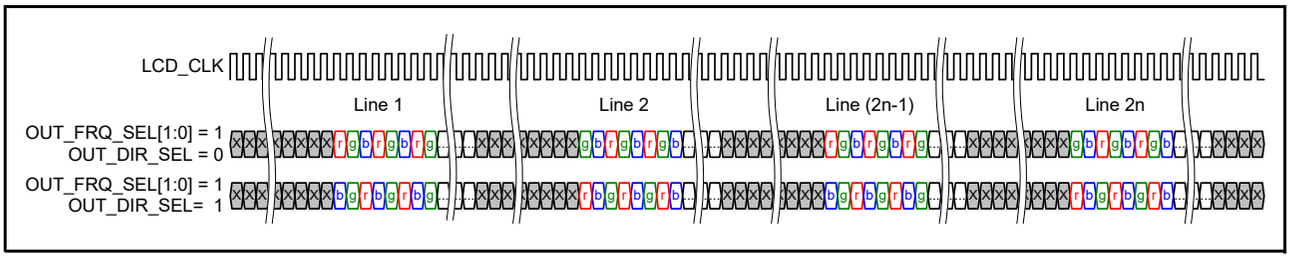


Figure 34.5 Timing of Parallel to Serial Conversion in Triple Speed Mode

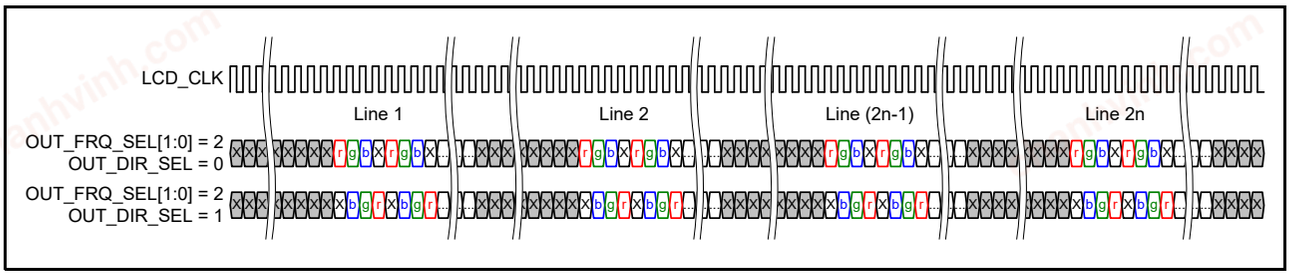


Figure 34.6 Timing of Parallel to Serial Conversion in Quadruple Speed Mode

During serial output, the phase timing with the HE signal can be adjusted by OUT_PHASE[0:1].

Figure 34.7 shows the timing of the clock phases of the serial RGB output (triple speed mode).

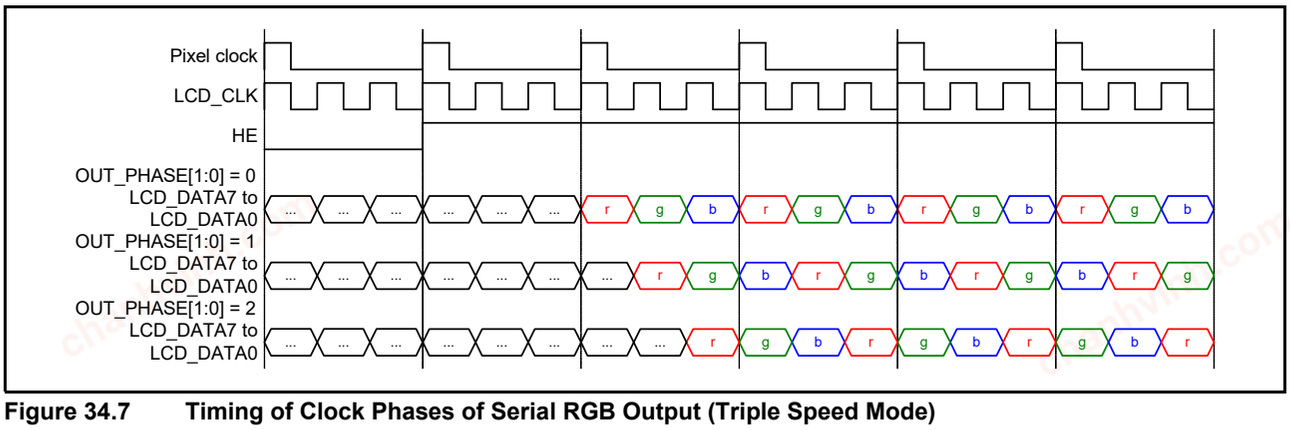


Figure 34.7 Timing of Clock Phases of Serial RGB Output (Triple Speed Mode)

chanhvinh.com

Figure 34.8 shows the timing of the clock phases of the serial RGB output (quadruple speed mode).

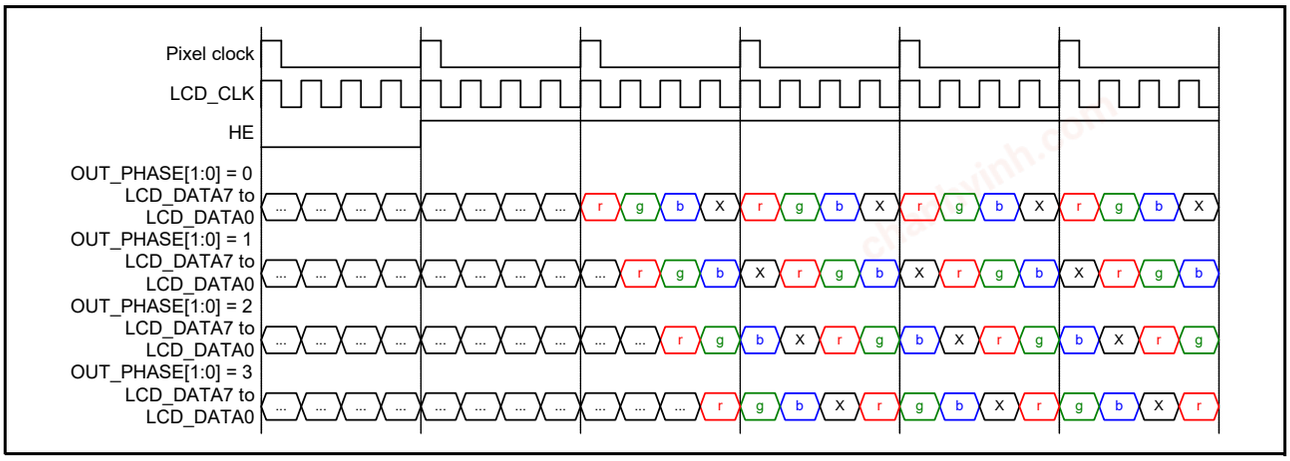


Figure 34.8 Timing of Clock Phases of Serial RGB Output (Quadruple Speed Mode)

Table 34.12 Output Format Conversion

Register Name	Bit Name	Initial Value	Description
OUT_SET	OUT_FORMAT[1:0]	0	Output Format Select 0: RBG888 1: RGB666 2: RGB565 3: Serial RGB
OUT_SET	OUT_ENDIAN_ON	0	Bit Endian Change On/Off Control 0: Off 1: On
OUT_SET	OUT_SWAP_ON	0	B/R Signal Swap On/Off Control 0: Off 1: On
OUT_SET	OUT_FRQ_SEL[1:0]	0	Clock Frequency Control 0: 100% speed — (parallel RGB) 1: Triple speed — (serial RGB) 2: Quadruple speed — (serial RGB) 3: Setting prohibited
OUT_SET	OUT_DIR_SEL	0	Scan Direction Select 0: Forward scan 1: Reverse scan
OUT_SET	OUT_PHASE[1:0]	0	Clock Phase Adjustment for Serial RGB Output Triple speed mode 0: 0 (clk) 1: 1 (clk) 2: 2 (clk) 3: Setting prohibited Quadruple speed mode 0: 0 (clk) 1: 1 (clk) 2: 2 (clk) 3: 3 (clk)

chanhvinh.com

34.1.9 LCD TCON

The LCD TCON generates various timing signals for driving the LCD panel.

Specifically, the timing include two vertical panel driver signals, five horizontal panel driver signals, and one composite signal of the vertical and horizontal panel driver signals. Table 34.13 lists the timing signals that are generated by LCD TCON

Table 34.13 Signals Generated by LCD TCON

Signal Name	Type	Description
STVA/VS	Vertical	• Gate start signal The pulse width, pulse position, and pulse polarity of the signal can be controlled. • Vsync signal The width, position, and polarity of the sync signal can be controlled.
STVB/VE	Vertical	• Gate start signal The pulse width, pulse position, and pulse polarity of the signal can be controlled. • Vertical enable signal The width, position, and polarity of the sync signal can be controlled.
STH/SP/HS	Horizontal	• Source start signal The pulse width, pulse position, and pulse polarity of the signal can be controlled. • Hsync signal The width, position, and polarity of the sync signal can be controlled.
STB/LP/HE	Horizontal	• Source strobe signal The pulse width, pulse position, and pulse polarity of the signal can be controlled. • Horizontal enable signal The width, position, and polarity of the enable signal can be controlled.
CPV/GCK	Horizontal	• Gate clock signal The pulse width, pulse position, and pulse polarity of the signal can be controlled.
POLA	Horizontal	• VCOM voltage polarity control signal The polarity inversion position, and polarity inversion operation (1 × 1, 1 × 2, 2 × 2) can be controlled.
POLB	Horizontal	• VCOM voltage polarity control signal The polarity inversion position, and polarity inversion operation (1 × 1, 1 × 2, 2 × 2) can be controlled.
DE	Horizontal/Vertical	• Data enable signal The width, position, and polarity of the enable signal can be controlled.

(1) Horizontal Reference Offset Control

The horizontal reference offset control enables generation of a reference signal with a clock delay equivalent to the value of TCON_OFFSET[10:0] from the rising edge of the Hsync signal. If a signal that spans across the Hsync signal needs to be generated, such a signal is generated with reference to the offset reference signal.

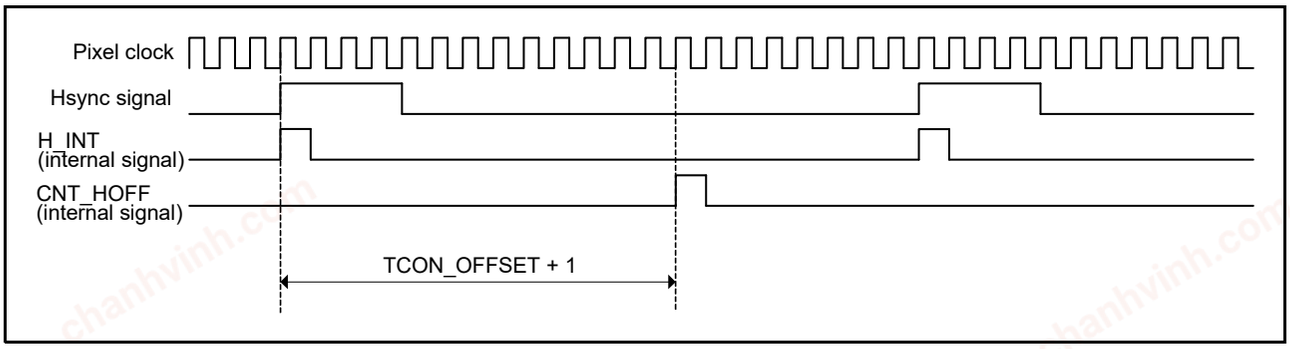


Figure 34.9 Generation of Offset Horizontal Reference (H_OFF) Signal

chanhvinh.com

Table 34.14 Horizontal Reference Signal Selection

Signal Name	Bit Name	Initial Value	Description
TCON_TIM	TCON_OFFSET[10:0]	0	Offset Hsync Signal Timing Sets the clock cycle count from the rising edge of the Hsync signal.
TCON_TIM_STH2	TCON_STH_HS_SEL	0	STH Signal Operating Reference Select 0: Hsync signal reference 1: Offset Hsync signal reference
TCON_TIM_STB2	TCON_STB_HS_SEL	0	STB Signal Operating Reference Select 0: Hsync signal reference 1: Offset Hsync signal reference
TCON_TIM_CPV2	TCON_CPV_HS_SEL	0	CPV Signal Operating Reference Select 0: Hsync signal reference 1: Offset Hsync signal reference
TCON_TIM_POLA2	TCON_POLA_HS_SEL	0	POLA Signal Operating Reference Select 0: Hsync signal reference 1: Offset Hsync signal reference
TCON_TIM_POLB2	TCON_POLB_HS_SEL	0	POLB Signal Operating Reference Select 0: Hsync signal reference 1: Offset Hsync signal reference

Note: When generating the POLA and POLB signals in reverse mode, the bits TCON_POLA_HS_SEL and TCON_POLB_HS_SEL should be set to 0.

(2) Horizontal Panel Driver Signal Generation (A)

Horizontal synchronous panel driver signal generation (A) involves generation of a timing signal that changes twice in a horizontal period according to the values of TCON_xxxx_HS[10:0] and TCON_xxxx_HW[10:0] bits, which set the first changing timing and the second changing timing, respectively.

The internal counter performs the following operations.

1. Resets the counter value at the rising edge of the Hsync signal as the reference.
2. Increments the counter value at the rising edge of the panel clock.

A fixed output value of 0 can be obtained by setting 0 in TCON_xxxx_HW[10:0], which set the second changing timing.

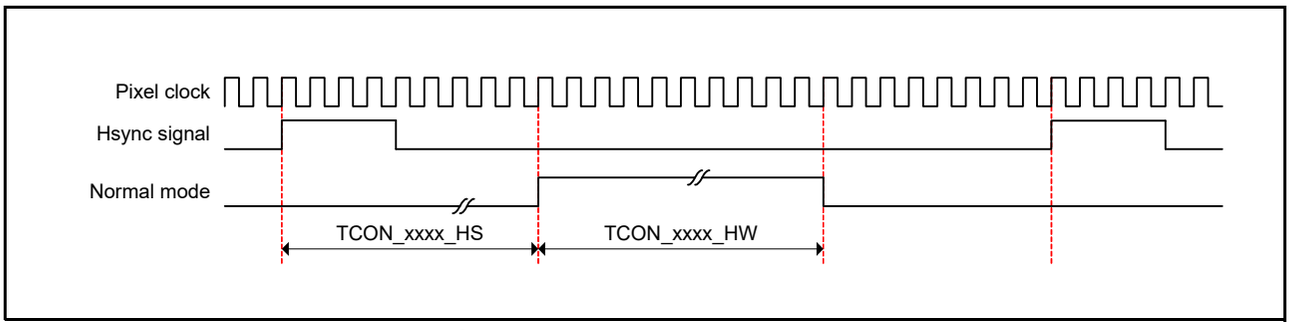


Figure 34.10 Horizontal Panel Driver Signal (in Normal Mode)

chanhvinh.com

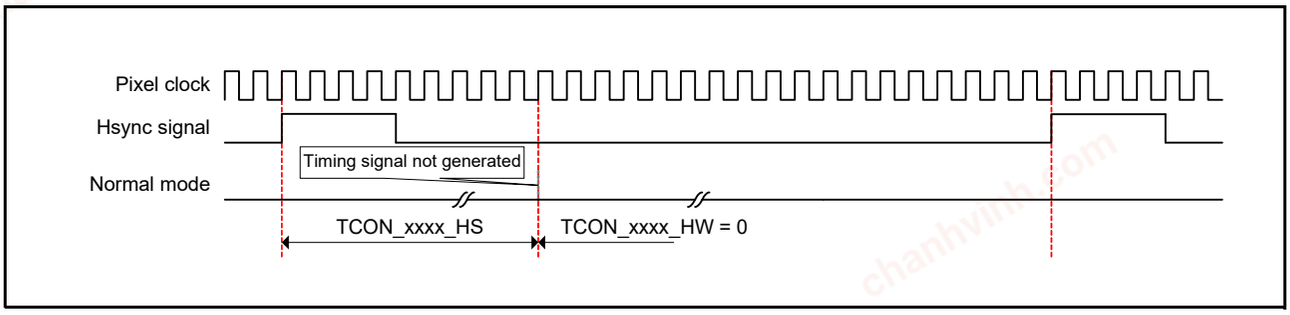


Figure 34.11 Horizontal Panel Driver Signal (in Normal Mode and When TCON_xxxx_HW_ = 0)

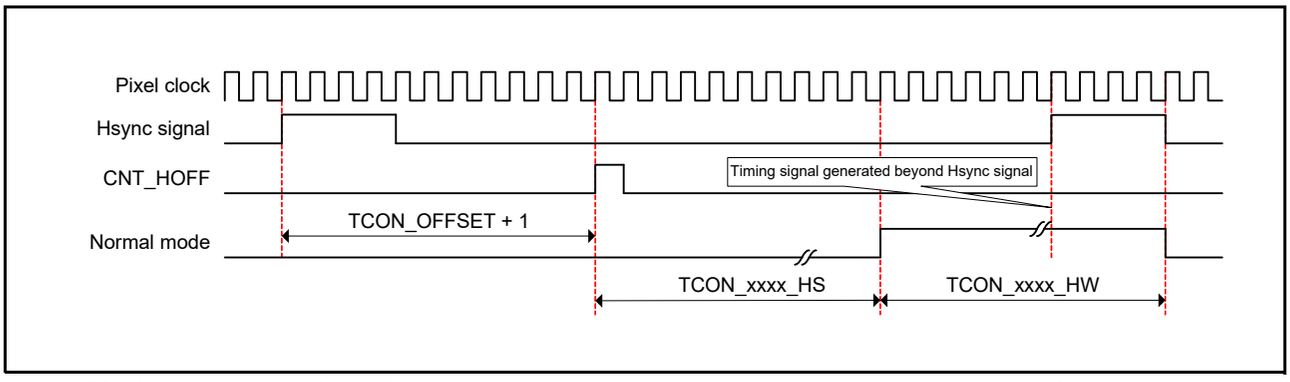


Figure 34.12 Horizontal Panel Driver Signal (in Normal Mode and When Offset Horizontal Reference is Used)

Table 34.15 Settings for Horizontal Panel Driver Signal Generation (A)

Register Name	Bit Name	Initial Value	Description
TCON_TIM_STH1	TCON_STH_HS[10:0]	0	STH Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_STH_HS from the rising edge of the Hsync signal (clock cycles)
TCON_TIM_STH1	TCON_STH_HW[10:0]	96	STH Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_STH_HW (clock cycles)
TCON_TIM_STB1	TCON_STB_HS[10:0]	144	STB Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_STB_HS from the rising edge of the Hsync signal (clock cycles)
TCON_TIM_STB1	TCON_STB_HW[10:0]	640	STB Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_STB_HW (clock cycles)
TCON_TIM_CPV1	TCON_CPV_HS[10:0]	0	CPV Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_CPV_HS from the rising edge of the Hsync signal (clock cycles)
TCON_TIM_CPV1	TCON_CPV_HW[10:0]	0	CPV Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_CPV_HW (clock cycles)

chanhvinh.com

(3) Horizontal Panel Driver Signal Generation (B)

In addition to the normal mode operation described in (2), reverse mode operation, that is, horizontal panel driver signal generation (B) is provided. In reverse mode, operation starts at the rising edge of the Vsync signal as the reference and a signal is generated such that its polarity is inverted every horizontal period in the timing set by the TCON_xxxx_HS[10:0] bits, which set the first changing timing.

In reverse mode, regardless of whether the number of lines in the vertical direction is odd or even, the polarity of the signals generated is inverted every horizontal period. The following three reverse modes are selectable for polarity inversion operation.

Table 34.16 Horizontal Panel Driver Signal Generation Modes

Register Name	Bit Name	Initial Value	Description
TCON_TIM_POLA2	TCON_POLA_MD [1:0]	1	POLA Signal Generation Mode Select 0: Normal mode Generates the signal that changes twice a horizontal period. 1: 1 × 1 reverse mode Generates the signal whose polarity is inverted every horizontal period. 2: 1 × 2 reverse mode Generates the signal whose polarity is inverted in the first horizontal period and is subsequently inverted every two horizontal periods. 3: 2 × 2 reverse mode Generates the signal whose polarity is inverted every two horizontal periods.
TCON_TIM_POLB2	TCON_POLB_MD [1:0]	1	POLB Signal Generation Mode Select 0: Normal mode Generates the signal that changes twice a horizontal period. 1: 1 × 1 reverse mode Generates the signal whose polarity is inverted every horizontal period. 2: 1 × 2 reverse mode Generates the signal whose polarity is inverted in the first horizontal period and is subsequently inverted every two horizontal periods. 3: 2 × 2 reverse mode Generates the signal whose polarity is inverted every two horizontal periods.

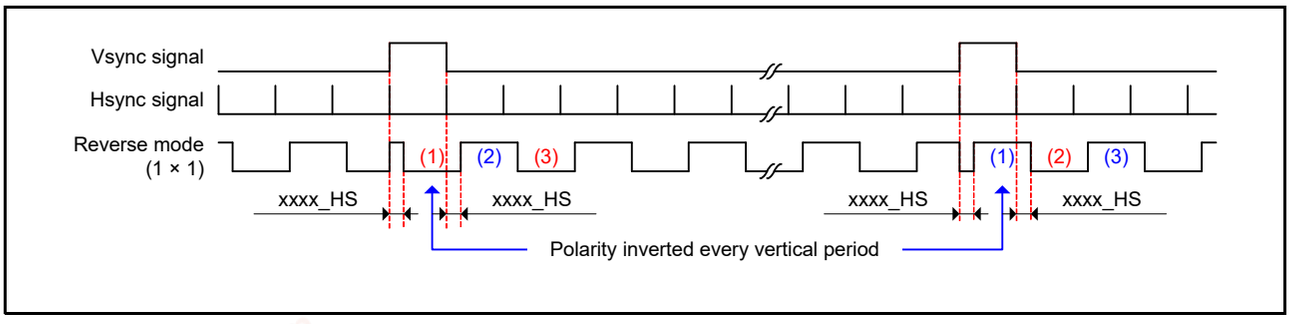


Figure 34.13 Horizontal Panel Driver Signal (in 1 × 1 Reverse Mode)

chanhvinh.com

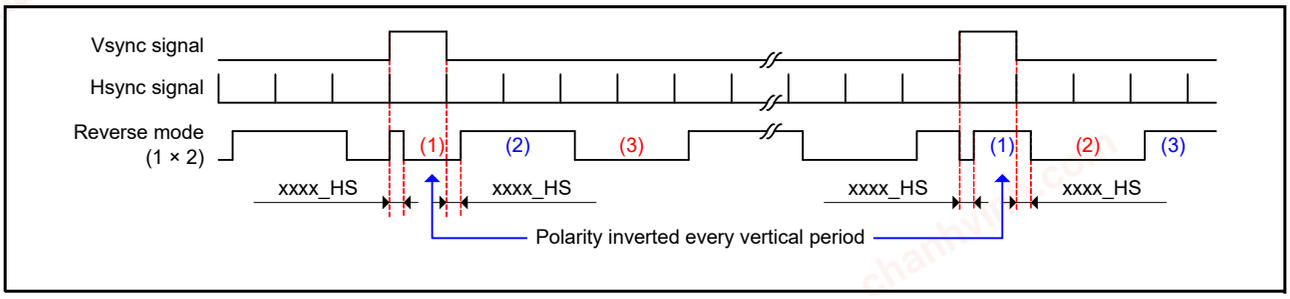


Figure 34.14 Horizontal Panel Driver Signal (in 1 × 2 Reverse Mode)

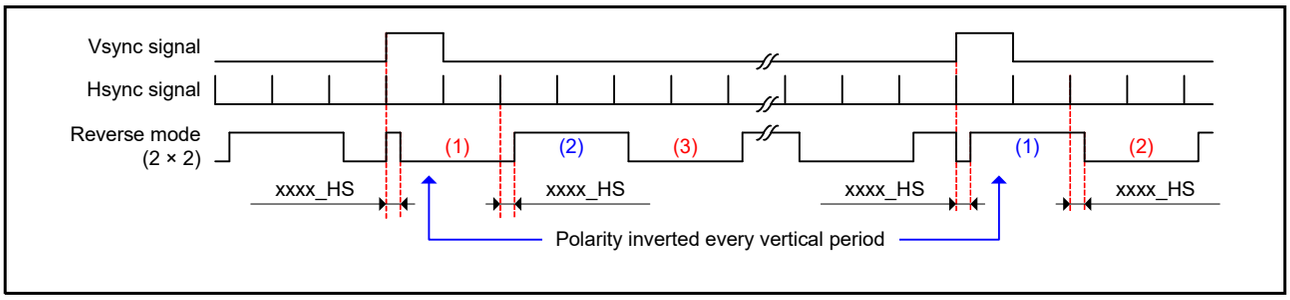


Figure 34.15 Horizontal Panel Driver Signal (in 2 × 2 Reverse Mode)

Table 34.17 Settings of Horizontal Panel Driver Signal Generation (B)

Register Name	Bit Name	Initial Value	Description
TCON_TIM_POLA1	TCON_POLA_HS [10:0]	0	POLA Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_POLA_HS + 1 from the rising edge of the Hsync signal (clock cycles) Note: When 1 × 1, 1 × 2, or 2 × 2 reverse mode is selected, these bits should be set to 1 or greater.
TCON_TIM_POLA1	TCON_POLA_HW [10:0]	0	POLA Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_POLA_HW (clock cycles)
TCON_TIM_POLB1	TCON_POLB_HS [10:0]	0	POLBA Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_POLB_HS + 1 from the rising edge of the Hsync signal (clock cycles) Note: When 1 × 1, 1 × 2, or 2 × 2 reverse mode is selected, these bits should be set to1 or greater.
TCON_TIM_POLB1	TCON_POLB_HW [10:0]	0	POLB Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_POLB_HW (clock cycles)

chanhvinh.com

(4) Vertical Panel Driver Signal Generation

The vertical synchronous panel driver signal generation involves the following operations.

1. Initialization at the rising edge of the Vsync signal
2. Generation of a timing signal that changes twice in a vertical period according to the values of the internal counter, and TCON_xxxx_VS[10:0] and TCON_xxxx_VW[10:0] bits, which set the first changing timing and the second changing timing, respectively.

The internal counter increments the counter value in the following two cases.

1. At the rising edge of the Hsync signal
2. At the point reached after a clock delay specified by the value of TCON_HALF[10:0] from the rising edge of the Hsync signal (normally, 1/2fH is set).

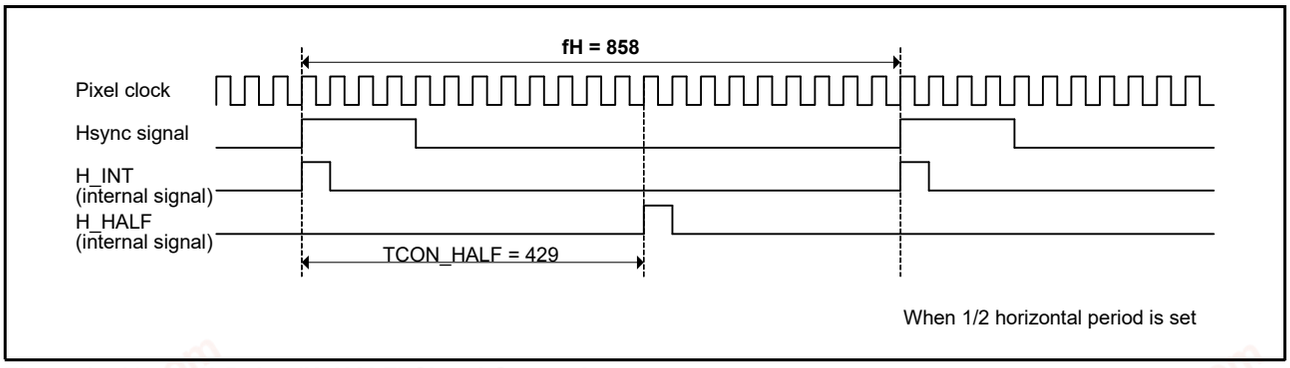


Figure 34.16 1/2 Pulse (H_HALF) Signal Generation

Table 34.18 Settings of 1/2 Pulse (H_HALF) Signal Generation

Register Name	Bit Name	Initial Value	Description
TCON_TIM	TCON_HALF[10:0]	400	1/2fH Timing Specifies the clock count from the rising edge of the Hsync signal as the counting timing of horizontal counter

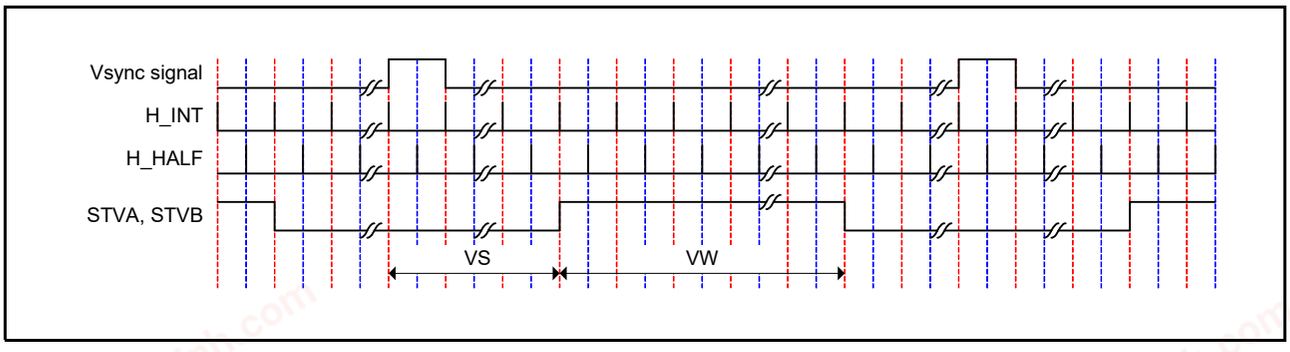


Figure 34.17 Vertical Panel Driver Signal (H_INT Reference Operation)

chanhvinh.com

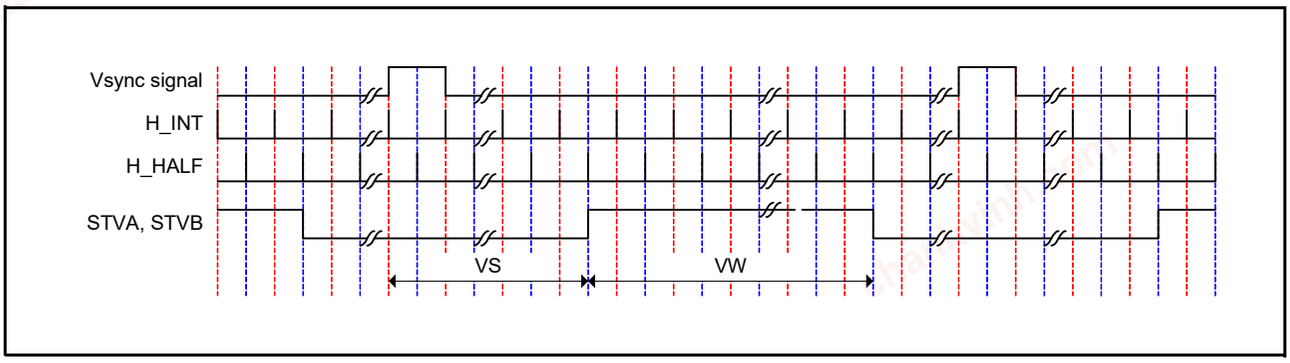


Figure 34.18 Vertical Panel Driver Signal (H_HALF Reference Operation)

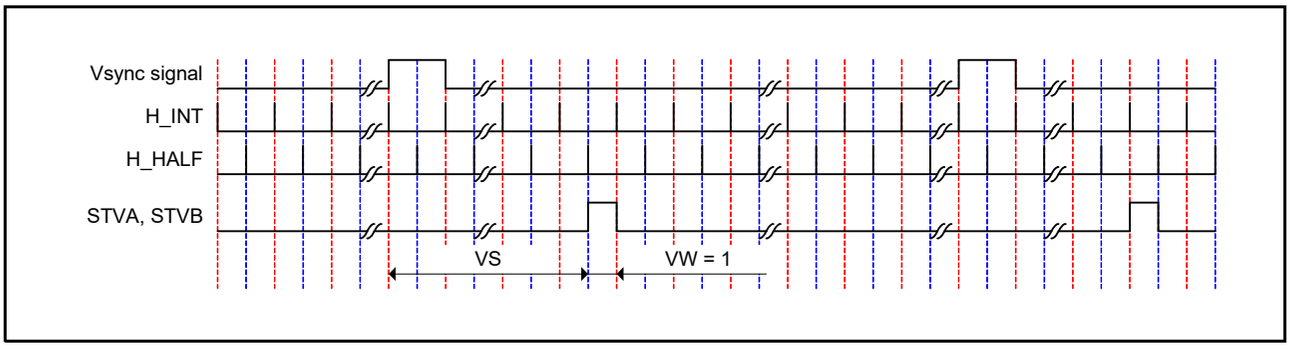


Figure 34.19 Vertical Panel Driver Signal (H_INT and H_HALF Reference Operation)

Table 34.19 Vertical Panel Driver Signal Generation

Register Name	Bit Name	Initial Value	Description
TCON_TIM_STVA1	TCON_STVA_VS[10:0]	0	STVA Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_STVA_HS from the rising edge of the Vsync signal (1/2fH cycles)
TCON_TIM_STVA1	TCON_STVA_VW[10:0]	4	STVA Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_STVA_HW (1/2fH cycles)
TCON_TIM_STVB1	TCON_STVB_VS[10:0]	70	STVB Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_STVB_HS from the rising edge of the Vsync signal (1/2fH cycles)
TCON_TIM_STVB1	TCON_STVB_VW[10:0]	960	STVB Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_STVB_HW (1/2fH cycles)

chanhvinh.com

(5) DE Timing Signal Generation

DE timing signal generation involves generation of data enable signal (DE) that indicates the valid period of the video signal by synthesizing the horizontal panel driver (HE) signal and the vertical panel driver (VE) signal (AND).

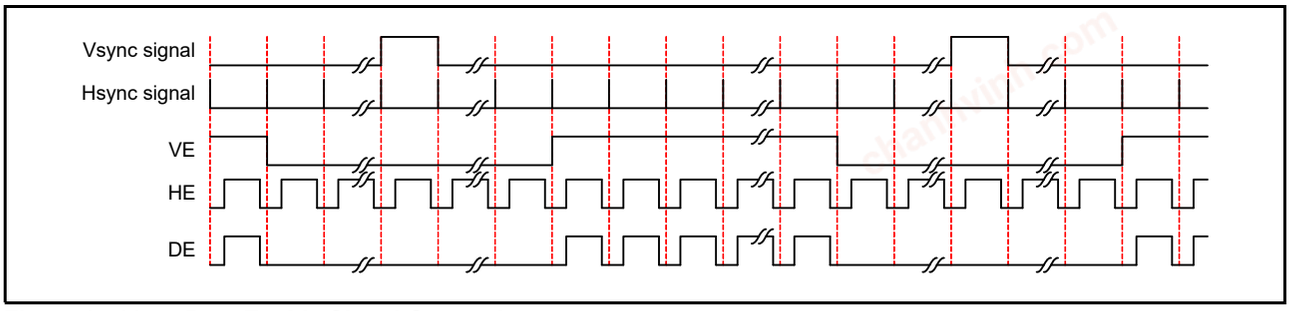


Figure 34.20 Data Enable Signal Generation

(6) Polarity Inversion

Polarity inversion enables inversion of polarity of each signal generated by the signal generating circuit.

Table 34.20 Panel Driver Signal Polarity Inversion Control

Register Name	Bit Name	Initial Value	Description
TCON_TIM_STVA2	TCON_STVA_INV	1	Polarity Inversion Control of STVA Signal 0: Not inverted 1: Inverted
TCON_TIM_STVB2	TCON_STVB_INV	0	Polarity Inversion Control of STVB Signal 0: Not inverted 1: Inverted
TCON_TIM_STH2	TCON_STH_INV	1	Polarity Inversion Control of STH Signal 0: Not inverted 1: Inverted
TCON_TIM_STB2	TCON_STB_INV	0	Polarity Inversion Control of STB Signal 0: Not inverted 1: Inverted
TCON_TIM_CPV2	TCON_CPV_INV	0	Polarity Inversion Control of CPV Signal 0: Not inverted 1: Inverted
TCON_TIM_POLA2	TCON_POLA_INV	0	Polarity Inversion Control of POLA Signal 0: Not inverted 1: Inverted
TCON_TIM_POLB2	TCON_POLB_INV	0	Polarity Inversion Control of POLB Signal 0: Not inverted 1: Inverted
TCON_TIM_DE	TCON_DE_INV	0	Polarity Inversion Control of DE Signal 0: Not inverted 1: Inverted

chanhvinh.com

(7) Output Selection

An output pin is selected for every signal subjected to polarity inversion control.

Table 34.21 Panel Driver Signal Output Selection

Register Name	Bit Name	Initial Value	Description
TCON_TIM_STVA2	TCON_STVA_SEL [2:0]	0	Output Signal Select for LCD_TCON0 Pin 0: STVA/VS 1: STVB/VE 2: STH/SP/HS 3: STB/LP/HE 4: CPV/GCK 5: POLA 6: POLB 7: DE
TCON_TIM_STVB2	TCON_STVB_SEL [2:0]	1	Output Signal Select for LCD_TCON1 Pin 0: STVA/VS 1: STVB/VE 2: STH/SP/HS 3: STB/LP/HE 4: CPV/GCK 5: POLA 6: POLB 7: DE
TCON_TIM_STH2	TCON_STH_SEL [2:0]	2	Output Signal Select for LCD_TCON2 Pin 0: STVA/VS 1: STVB/VE 2: STH/SP/HS 3: STB/LP/HE 4: CPV/GCK 5: POLA 6: POLB 7: DE
TCON_TIM_STB2	TCON_STB_SEL [2:0]	7	Output Signal Select for LCD_TCON3 Pin 0: STVA/VS 1: STVB/VE 2: STH/SP/HS 3: STB/LP/HE 4: CPV/GCK 5: POLA 6: POLB 7: DE
TCON_TIM_CPV2	TCON_CPV_SEL [2:0]	4	Output Signal Select for LCD_TCON4 Pin 0: STVA/VS 1: STVB/VE 2: STH/SP/HS 3: STB/LP/HE 4: CPV/GCK 5: POLA 6: POLB 7: DE
TCON_TIM_POLA2	TCON_POLA_SEL [2:0]	5	Output Signal Select for LCD_TCON5 Pin 0: STVA/VS 1: STVB/VE 2: STH/SP/HS 3: STB/LP/HE 4: CPV/GCK 5: POLA 6: POLB 7: DE

chanhvinh.com

Table 34.21 Panel Driver Signal Output Selection

Register Name	Bit Name	Initial Value	Description
TCON_TIM_POLB2	TCON_POLB_SEL [2:0]	6	Output Signal Select for LCD_TCON6 Pin 0: STVA/VS 1: STVB/VE 2: STH/SP/HS 3: STB/LP/HE 4: CPV/GCK 5: POLA 6: POLB 7: DE

(8) Output Phase Selection

The output phase can be individually selected for the video output signal and the various timing output signals based on the LCD_CLK (panel clock).

Table 34.22 Panel Output Signal Phase Selection

Register Name	Bit Name	Initial Value	Description
OUT_CLK_PHASE	OUTCNT_LCD_EDGE	0	Output Phase Control of LCD_DATA23 to LCD_DATA0 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
OUT_CLK_PHASE	OUTCNT_STVA_EDGE	0	Output Phase Control of LCD_TCON0 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
OUT_CLK_PHASE	OUTCNT_STVB_EDGE	0	Output Phase Control of LCD_TCON1 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
OUT_CLK_PHASE	OUTCNT_STH_EDGE	0	Output Phase Control of LCD_TCON2 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
OUT_CLK_PHASE	OUTCNT_STB_EDGE	0	Output Phase Control of LCD_TCON3 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
OUT_CLK_PHASE	OUTCNT_CPV_EDGE	0	Output Phase Control of LCD_TCON4 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
OUT_CLK_PHASE	OUTCNT_POLA_EDGE	0	Output Phase Control of LCD_TCON5 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
OUT_CLK_PHASE	OUTCNT_POLB_EDGE	0	Output Phase Control of LCD_TCON6 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin

chanhvinh.com

34.2 Register Descriptions

Table 34.23 to Table 34.25 show the register configuration.

- Symbols used in Register Description:

  Initial value: Register value after a reset
  —: Undefined value
  R/W: Readable/writable. The written value can be read.
  R/WC0: Readable/writable. Writing 0 initializes the bit. Writing 1 is ignored.
  R/WC1: Readable/writable. Writing 1 initializes the bit. Writing 0 is ignored.
  R: Read-only. The write value should always be 0.
  —/W: Write-only. The read value is undefined.

Table 34.23 Gamma Correction Block Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Register update control register G in gamma correction block	GAM_G_UPDATE	R/WC1	H'0000 0000	H'FCFF 7800	32
Function switch register in gamma correction block	GAM_SW	R/W	H'0000 0000	H'FCFF 7804	32
Table setting register G1 in gamma correction block	GAM_G_LUT1	R/W	H'0400 0400	H'FCFF 7808	32
Table setting register G2 in gamma correction block	GAM_G_LUT2	R/W	H'0400 0400	H'FCFF 780C	32
Table setting register G3 in gamma correction block	GAM_G_LUT3	R/W	H'0400 0400	H'FCFF 7810	32
Table setting register G4 in gamma correction block	GAM_G_LUT4	R/W	H'0400 0400	H'FCFF 7814	32
Table setting register G5 in gamma correction block	GAM_G_LUT5	R/W	H'0400 0400	H'FCFF 7818	32
Table setting register G6 in gamma correction block	GAM_G_LUT6	R/W	H'0400 0400	H'FCFF 781C	32
Table setting register G7 in gamma correction block	GAM_G_LUT7	R/W	H'0400 0400	H'FCFF 7820	32
Table setting register G8 in gamma correction block	GAM_G_LUT8	R/W	H'0400 0400	H'FCFF 7824	32
Table setting register G9 in gamma correction block	GAM_G_LUT9	R/W	H'0400 0400	H'FCFF 7828	32
Table setting register G10 in gamma correction block	GAM_G_LUT10	R/W	H'0400 0400	H'FCFF 782C	32
Table setting register G11 in gamma correction block	GAM_G_LUT11	R/W	H'0400 0400	H'FCFF 7830	32
Table setting register G12 in gamma correction block	GAM_G_LUT12	R/W	H'0400 0400	H'FCFF 7834	32
Table setting register G13 in gamma correction block	GAM_G_LUT13	R/W	H'0400 0400	H'FCFF 7838	32
Table setting register G14 in gamma correction block	GAM_G_LUT14	R/W	H'0400 0400	H'FCFF 783C	32
Table setting register G15 in gamma correction block	GAM_G_LUT15	R/W	H'0400 0400	H'FCFF 7840	32
Table setting register G16 in gamma correction block	GAM_G_LUT16	R/W	H'0400 0400	H'FCFF 7844	32
Area setting register G1 in gamma correction block	GAM_G_AREA1	R/W	H'0008 1018	H'FCFF 7848	32

chanhvinh.com

Table 34.23 Gamma Correction Block Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Area setting register G2 in gamma correction block	GAM_G_AREA2	R/W	H'2028 3038	H'FCFF 784C	32
Area setting register G3 in gamma correction block	GAM_G_AREA3	R/W	H'4048 5058	H'FCFF 7850	32
Area setting register G4 in gamma correction block	GAM_G_AREA4	R/W	H'6068 7078	H'FCFF 7854	32
Area setting register G5 in gamma correction block	GAM_G_AREA5	R/W	H'8088 9098	H'FCFF 7858	32
Area setting register G6 in gamma correction block	GAM_G_AREA6	R/W	H'A0A8 B0B8	H'FCFF 785C	32
Area setting register G7 in gamma correction block	GAM_G_AREA7	R/W	H'C0C8 D0D8	H'FCFF 7860	32
Area setting register G8 in gamma correction block	GAM_G_AREA8	R/W	H'E0E8 F0F8	H'FCFF 7864	32
Register update control register B in gamma correction block	GAM_B_UPDATE	R/WC1	H'0000 0000	H'FCFF 7880	32
Table setting register B1 in gamma correction block	GAM_B_LUT1	R/W	H'0400 0400	H'FCFF 7888	32
Table setting register B2 in gamma correction block	GAM_B_LUT2	R/W	H'0400 0400	H'FCFF 788C	32
Table setting register B3 in gamma correction block	GAM_B_LUT3	R/W	H'0400 0400	H'FCFF 7890	32
Table setting register B4 in gamma correction block	GAM_B_LUT4	R/W	H'0400 0400	H'FCFF 7894	32
Table setting register B5 in gamma correction block	GAM_B_LUT5	R/W	H'0400 0400	H'FCFF 7898	32
Table setting register B6 in gamma correction block	GAM_B_LUT6	R/W	H'0400 0400	H'FCFF 789C	32
Table setting register B7 in gamma correction block	GAM_B_LUT7	R/W	H'0400 0400	H'FCFF 78A0	32
Table setting register B8 in gamma correction block	GAM_B_LUT8	R/W	H'0400 0400	H'FCFF 78A4	32
Table setting register B9 in gamma correction block	GAM_B_LUT9	R/W	H'0400 0400	H'FCFF 78A8	32
Table setting register B10 in gamma correction block	GAM_B_LUT10	R/W	H'0400 0400	H'FCFF 78AC	32
Table setting register B11 in gamma correction block	GAM_B_LUT11	R/W	H'0400 0400	H'FCFF 78B0	32
Table setting register B12 in gamma correction block	GAM_B_LUT12	R/W	H'0400 0400	H'FCFF 78B4	32
Table setting register B13 in gamma correction block	GAM_B_LUT13	R/W	H'0400 0400	H'FCFF 78B8	32
Table setting register B14 in gamma correction block	GAM_B_LUT14	R/W	H'0400 0400	H'FCFF 78BC	32
Table setting register B15 in gamma correction block	GAM_B_LUT15	R/W	H'0400 0400	H'FCFF 78C0	32
Table setting register B16 in gamma correction block	GAM_B_LUT16	R/W	H'0400 0400	H'FCFF 78C4	32
Area setting register B1 in gamma correction block	GAM_B_AREA1	R/W	H'0008 1018	H'FCFF 78C8	32
Area setting register B2 in gamma correction block	GAM_B_AREA2	R/W	H'2028 3038	H'FCFF 78CC	32

chanhvinh.com

Table 34.23 Gamma Correction Block Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Area setting register B3 in gamma correction block	GAM_B_AREA3	R/W	H'4048 5058	H'FCFF 78D0	32
Area setting register B4 in gamma correction block	GAM_B_AREA4	R/W	H'6068 7078	H'FCFF 78D4	32
Area setting register B5 in gamma correction block	GAM_B_AREA5	R/W	H'8088 9098	H'FCFF 78D8	32
Area setting register B6 in gamma correction block	GAM_B_AREA6	R/W	H'A0A8 B0B8	H'FCFF 78DC	32
Area setting register B7 in gamma correction block	GAM_B_AREA7	R/W	H'C0C8 D0D8	H'FCFF 78E0	32
Area setting register B8 in gamma correction block	GAM_B_AREA8	R/W	H'E0E8 F0F8	H'FCFF 78E4	32
Register update control register R in gamma correction block	GAM_R_UPDATE	R/WC1	H'0000 0000	H'FCFF 7900	32
Table setting register R1 in gamma correction block	GAM_R_LUT1	R/W	H'0400 0400	H'FCFF 7908	32
Table setting register R2 in gamma correction block	GAM_R_LUT2	R/W	H'0400 0400	H'FCFF 790C	32
Table setting register R3 in gamma correction block	GAM_R_LUT3	R/W	H'0400 0400	H'FCFF 7910	32
Table setting register R4 in gamma correction block	GAM_R_LUT4	R/W	H'0400 0400	H'FCFF 7914	32
Table setting register R5 in gamma correction block	GAM_R_LUT5	R/W	H'0400 0400	H'FCFF 7918	32
Table setting register R6 in gamma correction block	GAM_R_LUT6	R/W	H'0400 0400	H'FCFF 791C	32
Table setting register R7 in gamma correction block	GAM_R_LUT7	R/W	H'0400 0400	H'FCFF 7920	32
Table setting register R8 in gamma correction block	GAM_R_LUT8	R/W	H'0400 0400	H'FCFF 7924	32
Table setting register R9 in gamma correction block	GAM_R_LUT9	R/W	H'0400 0400	H'FCFF 7928	32
Table setting register R10 in gamma correction block	GAM_R_LUT10	R/W	H'0400 0400	H'FCFF 792C	32
Table setting register R11 in gamma correction block	GAM_R_LUT11	R/W	H'0400 0400	H'FCFF 7930	32
Table setting register R12 in gamma correction block	GAM_R_LUT12	R/W	H'0400 0400	H'FCFF 7934	32
Table setting register R13 in gamma correction block	GAM_R_LUT13	R/W	H'0400 0400	H'FCFF 7938	32
Table setting register R14 in gamma correction block	GAM_R_LUT14	R/W	H'0400 0400	H'FCFF 793C	32
Table setting register R15 in gamma correction block	GAM_R_LUT15	R/W	H'0400 0400	H'FCFF 7940	32
Table setting register R16 in gamma correction block	GAM_R_LUT16	R/W	H'0400 0400	H'FCFF 7944	32
Area setting register R1 in gamma correction block	GAM_R_AREA1	R/W	H'0008 1018	H'FCFF 7948	32
Area setting register R2 in gamma correction block	GAM_R_AREA2	R/W	H'2028 3038	H'FCFF 794C	32
Area setting register R3 in gamma correction block	GAM_R_AREA3	R/W	H'4048 5058	H'FCFF 7950	32

chanhvinh.com

Table 34.23 Gamma Correction Block Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Area setting register R4 in gamma correction block	GAM_R_AREA4	R/W	H'6068 7078	H'FCFF 7954	32
Area setting register R5 in gamma correction block	GAM_R_AREA5	R/W	H'8088 9098	H'FCFF 7958	32
Area setting register R6 in gamma correction block	GAM_R_AREA6	R/W	H'A0A8 B0B8	H'FCFF 795C	32
Area setting register R7 in gamma correction block	GAM_R_AREA7	R/W	H'C0C8 D0D8	H'FCFF 7960	32
Area setting register R8 in gamma correction block	GAM_R_AREA8	R/W	H'E0E8 F0F8	H'FCFF 7964	32

Table 34.24 TCON Block Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
TCON register update control register	TCON_UPDATE	R/WC1	H'0000 0000	H'FCFF 7980	32
TCON reference timing setting register	TCON_TIM	R/W	H'0190 0000	H'FCFF 7984	32
TCON vertical timing setting register A1	TCON_TIM_STVA1	R/W	H'0000 0004	H'FCFF 7988	32
TCON vertical timing setting register A2	TCON_TIM_STVA2	R/W	H'0000 0010	H'FCFF 798C	32
TCON vertical timing setting register B1	TCON_TIM_STVB1	R/W	H'0046 03C0	H'FCFF 7990	32
TCON vertical timing setting register B2	TCON_TIM_STVB2	R/W	H'0000 0001	H'FCFF 7994	32
TCON horizontal timing setting register STH1	TCON_TIM_STH1	R/W	H'0000 0060	H'FCFF 7998	32
TCON horizontal timing setting register STH2	TCON_TIM_STH2	R/W	H'0000 0012	H'FCFF 799C	32
TCON horizontal timing setting register STB1	TCON_TIM_STB1	R/W	H'0090 0280	H'FCFF 79A0	32
TCON horizontal timing setting register STB2	TCON_TIM_STB2	R/W	H'0000 0007	H'FCFF 79A4	32
TCON horizontal timing setting register CPV1	TCON_TIM_CPV1	R/W	H'0000 0000	H'FCFF 79A8	32
TCON horizontal timing setting register CPV2	TCON_TIM_CPV2	R/W	H'0000 0004	H'FCFF 79AC	32
TCON horizontal timing setting register POLA1	TCON_TIM_POLA1	R/W	H'0000 0000	H'FCFF 79B0	32
TCON horizontal timing setting register POLA2	TCON_TIM_POLA2	R/W	H'0000 1005	H'FCFF 79B4	32
TCON horizontal timing setting register POLB1	TCON_TIM_POLB1	R/W	H'0000 0000	H'FCFF 79B8	32
TCON horizontal timing setting register POLB2	TCON_TIM_POLB2	R/W	H'0000 1006	H'FCFF 79BC	32
TCON data enable polarity setting register	TCON_TIM_DE	R/W	H'0000 0000	H'FCFF 79C0	32

chanhvinh.com

Table 34.25 Output Controller Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Register update control register in output controller	OUT_UPDATE	R/WC1	H'0000 0000	H'FCFF 7A00	32
Output interface register	OUT_SET	R/W	H'001F 0000	H'FCFF 7A04	32
Brightness (DC) correction register 1	OUT_BRIGHT1	R/W	H'0000 0200	H'FCFF 7A08	32
Brightness (DC) correction register 2	OUT_BRIGHT2	R/W	H'0200 0200	H'FCFF 7A0C	32
Contrast (gain) correction register	OUT_CONTRAST	R/W	H'0080 8080	H'FCFF 7A10	32
Panel dither register	OUT_PDTHA	R/W	H'0000 3021	H'FCFF 7A14	32
Output phase control register	OUT_CLK_PHASE	R/W	H'0000 0000	H'FCFF 7A24	32

34.2.1 Register Update Control Register G in Gamma Correction Block (GAM_G_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GAM_G_VEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GAM_G_VEN	0	R/WC1	Gamma Correction (G) Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

chanhvinh.com

34.2.2 Function Switch Register in Gamma Correction Block (GAM_SW)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GAM_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GAM_ON	0	R/W	Gamma Correction On/Off Control 0: Off 1: On

Note: This register is updated when GAM_G_VEN in GAM_G_UPDATE is 1.

chanhvinh.com

34.2.3 Table Setting Register G1 to G16 in Gamma Correction Block (GAM_G_LUT1 to GAM_G_LUT16)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GAM_G_GAIN_xx[10:0]										
Initial value:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GAM_G_GAIN_yy[10:0]										
Initial value:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	*	1024	R/W	GAM_G_LUT1: Gain Adjustment of Area 0 of G Signal GAM_G_LUT2: Gain Adjustment of Area 2 of G Signal GAM_G_LUT3: Gain Adjustment of Area 4 of G Signal GAM_G_LUT4: Gain Adjustment of Area 6 of G Signal GAM_G_LUT5: Gain Adjustment of Area 8 of G Signal GAM_G_LUT6: Gain Adjustment of Area 10 of G Signal GAM_G_LUT7: Gain Adjustment of Area 12 of G Signal GAM_G_LUT8: Gain Adjustment of Area 14 of G Signal GAM_G_LUT9: Gain Adjustment of Area 16 of G Signal GAM_G_LUT10: Gain Adjustment of Area 18 of G Signal GAM_G_LUT11: Gain Adjustment of Area 20 of G Signal GAM_G_LUT12: Gain Adjustment of Area 22 of G Signal GAM_G_LUT13: Gain Adjustment of Area 24 of G Signal GAM_G_LUT14: Gain Adjustment of Area 26 of G Signal GAM_G_LUT15: Gain Adjustment of Area 28 of G Signal GAM_G_LUT16: Gain Adjustment of Area 30 of G Signal Unsigned (0 to 2047 [LSB], 1024 [LSB] = 1.0 [times])
26 to 16	*	1024	R/W	*: Bit Name GAM_G_LUT1: GAM_G_GAIN_00[10:0] GAM_G_LUT2: GAM_G_GAIN_02[10:0] GAM_G_LUT3: GAM_G_GAIN_04[10:0] GAM_G_LUT4: GAM_G_GAIN_06[10:0] GAM_G_LUT5: GAM_G_GAIN_08[10:0] GAM_G_LUT6: GAM_G_GAIN_10[10:0] GAM_G_LUT7: GAM_G_GAIN_12[10:0] GAM_G_LUT8: GAM_G_GAIN_14[10:0] GAM_G_LUT9: GAM_G_GAIN_16[10:0] GAM_G_LUT10: GAM_G_GAIN_18[10:0] GAM_G_LUT11: GAM_G_GAIN_20[10:0] GAM_G_LUT12: GAM_G_GAIN_22[10:0] GAM_G_LUT13: GAM_G_GAIN_24[10:0] GAM_G_LUT14: GAM_G_GAIN_26[10:0] GAM_G_LUT15: GAM_G_GAIN_28[10:0] GAM_G_LUT16: GAM_G_GAIN_30[10:0]
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
10 to 0	*	1024	R/W	GAM_G_LUT1: Gain Adjustment of Area 1 of G Signal GAM_G_LUT2: Gain Adjustment of Area 3 of G Signal GAM_G_LUT3: Gain Adjustment of Area 5 of G Signal GAM_G_LUT4: Gain Adjustment of Area 7 of G Signal GAM_G_LUT5: Gain Adjustment of Area 9 of G Signal GAM_G_LUT6: Gain Adjustment of Area 11 of G Signal GAM_G_LUT7: Gain Adjustment of Area 13 of G Signal GAM_G_LUT8: Gain Adjustment of Area 15 of G Signal GAM_G_LUT9: Gain Adjustment of Area 17 of G Signal GAM_G_LUT10: Gain Adjustment of Area 19 of G Signal
10 to 0	*	1024	R/W	GAM_G_LUT11: Gain Adjustment of Area 21 of G Signal GAM_G_LUT12: Gain Adjustment of Area 23 of G Signal GAM_G_LUT13: Gain Adjustment of Area 25 of G Signal GAM_G_LUT14: Gain Adjustment of Area 27 of G Signal GAM_G_LUT15: Gain Adjustment of Area 29 of G Signal GAM_G_LUT16: Gain Adjustment of Area 31 of G Signal Unsigned (0 to 2047 [LSB], 1024 [LSB] = 1.0 [times]) *: Bit Name GAM_G_LUT1: GAM_G_GAIN_01[10:0] GAM_G_LUT2: GAM_G_GAIN_03[10:0] GAM_G_LUT3: GAM_G_GAIN_05[10:0] GAM_G_LUT4: GAM_G_GAIN_07[10:0] GAM_G_LUT5: GAM_G_GAIN_09[10:0] GAM_G_LUT6: GAM_G_GAIN_11[10:0] GAM_G_LUT7: GAM_G_GAIN_13[10:0] GAM_G_LUT8: GAM_G_GAIN_15[10:0] GAM_G_LUT9: GAM_G_GAIN_17[10:0] GAM_G_LUT10: GAM_G_GAIN_19[10:0] GAM_G_LUT11: GAM_G_GAIN_21[10:0] GAM_G_LUT12: GAM_G_GAIN_23[10:0] GAM_G_LUT13: GAM_G_GAIN_25[10:0] GAM_G_LUT14: GAM_G_GAIN_27[10:0] GAM_G_LUT15: GAM_G_GAIN_29[10:0] GAM_G_LUT16: GAM_G_GAIN_31[10:0]

Note: This register is updated when GAM_G_VEN in GAM_G_UPDATE is 1.

chanhvinh.com

34.2.4 Area Setting Register G1 in Gamma Correction Block (GAM_G_AREA1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	GAM_G_TH_01[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_G_TH_02[7:0]								GAM_G_TH_03[7:0]							
Initial value:	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	GAM_G_TH_01[7:0]	8	R/W	Start Threshold of Area 1 of G Signal Unsigned (0 to 255 [LSB]) 0 < Threshold of current area < Threshold of next area
15 to 8	GAM_G_TH_02[7:0]	16	R/W	Start Threshold of Area 2 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_G_TH_03[7:0]	24	R/W	Start Threshold of Area 3 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_G_VEN in GAM_G_UPDATE is 1.

chanhvinh.com

34.2.5 Area Setting Register G2 in Gamma Correction Block (GAM_G_AREA2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_G_TH_04[7:0]								GAM_G_TH_05[7:0]							
Initial value:	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_G_TH_06[7:0]								GAM_G_TH_07[7:0]							
Initial value:	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_G_TH_04[7:0]	32	R/W	Start Threshold of Area 4 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_G_TH_05[7:0]	40	R/W	Start Threshold of Area 5 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_G_TH_06[7:0]	48	R/W	Start Threshold of Area 6 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_G_TH_07[7:0]	56	R/W	Start Threshold of Area 7 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_G_VEN in GAM_G_UPDATE is 1.

chanhvinh.com

34.2.6 Area Setting Register G3 in Gamma Correction Block (GAM_G_AREA3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_G_TH_08[7:0]								GAM_G_TH_09[7:0]							
Initial value:	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_G_TH_10[7:0]								GAM_G_TH_11[7:0]							
Initial value:	0	1	0	1	0	0	0	0	0	1	0	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_G_TH_08[7:0]	64	R/W	Start Threshold of Area 8 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_G_TH_09[7:0]	72	R/W	Start Threshold of Area 9 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_G_TH_10[7:0]	80	R/W	Start Threshold of Area 10 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_G_TH_11[7:0]	88	R/W	Start Threshold of Area 11 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_G_VEN in GAM_G_UPDATE is 1.

chanhvinh.com

34.2.7 Area Setting Register G4 in Gamma Correction Block (GAM_G_AREA4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_G_TH_12[7:0]								GAM_G_TH_13[7:0]							
Initial value:	0	1	1	0	0	0	0	0	0	1	1	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_G_TH_14[7:0]								GAM_G_TH_15[7:0]							
Initial value:	0	1	1	1	0	0	0	0	0	1	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_G_TH_12[7:0]	96	R/W	Start Threshold of Area 12 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_G_TH_13[7:0]	104	R/W	Start Threshold of Area 13 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_G_TH_14[7:0]	112	R/W	Start Threshold of Area 14 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_G_TH_15[7:0]	120	R/W	Start Threshold of Area 15 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_G_VEN in GAM_G_UPDATE is 1.

chanhvinh.com

34.2.8 Area Setting Register G5 in Gamma Correction Block (GAM_G_AREA5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_G_TH_16[7:0]								GAM_G_TH_17[7:0]							
Initial value:	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_G_TH_18[7:0]								GAM_G_TH_19[7:0]							
Initial value:	1	0	0	1	0	0	0	0	1	0	0	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_G_TH_16 [7:0]	128	R/W	Start Threshold of Area 16 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_G_TH_17 [7:0]	136	R/W	Start Threshold of Area 17 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_G_TH_18 [7:0]	144	R/W	Start Threshold of Area 18 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_G_TH_19 [7:0]	152	R/W	Start Threshold of Area 19 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_G_VEN in GAM_G_UPDATE is 1.

chanhvinh.com

34.2.9 Area Setting Register G6 in Gamma Correction Block (GAM_G_AREA6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_G_TH_20[7:0]								GAM_G_TH_21[7:0]							
Initial value:	1	0	1	0	0	0	0	0	1	0	1	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_G_TH_22[7:0]								GAM_G_TH_23[7:0]							
Initial value:	1	0	1	1	0	0	0	0	1	0	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_G_TH_20 [7:0]	160	R/W	Start Threshold of Area 20 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_G_TH_21 [7:0]	168	R/W	Start Threshold of Area 21 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_G_TH_22 [7:0]	176	R/W	Start Threshold of Area 22 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_G_TH_23 [7:0]	184	R/W	Start Threshold of Area 23 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_G_VEN in GAM_G_UPDATE is 1.

chanhvinh.com

34.2.10 Area Setting Register G7 in Gamma Correction Block (GAM_G_AREA7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_G_TH_24[7:0]								GAM_G_TH_25[7:0]							
Initial value:	1	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_G_TH_26[7:0]								GAM_G_TH_27[7:0]							
Initial value:	1	1	0	1	0	0	0	0	1	1	0	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_G_TH_24 [7:0]	192	R/W	Start Threshold of Area 24 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_G_TH_25 [7:0]	200	R/W	Start Threshold of Area 25 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_G_TH_26 [7:0]	208	R/W	Start Threshold of Area 26 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_G_TH_27 [7:0]	216	R/W	Start Threshold of Area 27 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_G_VEN in GAM_G_UPDATE is 1.

chanhvinh.com

34.2.11 Area Setting Register G8 in Gamma Correction Block (GAM_G_AREA8)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_G_TH_28[7:0]								GAM_G_TH_29[7:0]							
Initial value:	1	1	1	0	0	0	0	0	1	1	1	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_G_TH_30[7:0]								GAM_G_TH_31[7:0]							
Initial value:	1	1	1	1	0	0	0	0	1	1	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_G_TH_28 [7:0]	224	R/W	Start Threshold of Area 28 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_G_TH_29 [7:0]	232	R/W	Start Threshold of Area 29 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_G_TH_30 [7:0]	240	R/W	Start Threshold of Area 30 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_G_TH_31 [7:0]	248	R/W	Start Threshold of Area 31 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area ≤ 255

Note: This register is updated when GAM_G_VEN in GAM_G_UPDATE is 1.

chanhvinh.com

34.2.12 Register Update Control Register B in Gamma Correction Block (GAM_B_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GAM_B_VEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GAM_B_VEN	0	R/WC1	Gamma Correction (B) Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

chanhvinh.com

34.2.13 Table Setting Register B1 to B16 in Gamma Correction Block (GAM_B_LUT1 to GAM_B_LUT16)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GAM_B_GAIN_xx[10:0]										
Initial value:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GAM_B_GAIN_yy[10:0]										
Initial value:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	*	1024	R/W	GAM_B_LUT1: Gain Adjustment of Area 0 of B Signal GAM_B_LUT2: Gain Adjustment of Area 2 of B Signal GAM_B_LUT3: Gain Adjustment of Area 4 of B Signal GAM_B_LUT4: Gain Adjustment of Area 6 of B Signal GAM_B_LUT5: Gain Adjustment of Area 8 of B Signal GAM_B_LUT6: Gain Adjustment of Area 10 of B Signal GAM_B_LUT7: Gain Adjustment of Area 12 of B Signal GAM_B_LUT8: Gain Adjustment of Area 14 of B Signal GAM_B_LUT9: Gain Adjustment of Area 16 of B Signal GAM_B_LUT10: Gain Adjustment of Area 18 of B Signal GAM_B_LUT11: Gain Adjustment of Area 20 of B Signal GAM_B_LUT12: Gain Adjustment of Area 22 of B Signal GAM_B_LUT13: Gain Adjustment of Area 24 of B Signal GAM_B_LUT14: Gain Adjustment of Area 26 of B Signal GAM_B_LUT15: Gain Adjustment of Area 28 of B Signal GAM_B_LUT16: Gain Adjustment of Area 30 of B Signal Unsigned (0 to 2047 [LSB], 1024 [LSB] = 1.0 [times])
26 to 16	*	1024	R/W	*: Bit Name GAM_B_LUT1: GAM_B_GAIN_00[10:0] GAM_B_LUT2: GAM_B_GAIN_02[10:0] GAM_B_LUT3: GAM_B_GAIN_04[10:0] GAM_B_LUT4: GAM_B_GAIN_06[10:0] GAM_B_LUT5: GAM_B_GAIN_08[10:0] GAM_B_LUT6: GAM_B_GAIN_10[10:0] GAM_B_LUT7: GAM_B_GAIN_12[10:0] GAM_B_LUT8: GAM_B_GAIN_14[10:0] GAM_B_LUT9: GAM_B_GAIN_16[10:0] GAM_B_LUT10: GAM_B_GAIN_18[10:0] GAM_B_LUT11: GAM_B_GAIN_20[10:0] GAM_B_LUT12: GAM_B_GAIN_22[10:0] GAM_B_LUT13: GAM_B_GAIN_24[10:0] GAM_B_LUT14: GAM_B_GAIN_26[10:0] GAM_B_LUT15: GAM_B_GAIN_28[10:0] GAM_B_LUT16: GAM_B_GAIN_30[10:0]
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
10 to 0	*	1024	R/W	GAM_B_LUT1: Gain Adjustment of Area 1 of B Signal GAM_B_LUT2: Gain Adjustment of Area 3 of B Signal GAM_B_LUT3: Gain Adjustment of Area 5 of B Signal GAM_B_LUT4: Gain Adjustment of Area 7 of B Signal GAM_B_LUT5: Gain Adjustment of Area 9 of B Signal GAM_B_LUT6: Gain Adjustment of Area 11 of B Signal GAM_B_LUT7: Gain Adjustment of Area 13 of B Signal GAM_B_LUT8: Gain Adjustment of Area 15 of B Signal GAM_B_LUT9: Gain Adjustment of Area 17 of B Signal GAM_B_LUT10: Gain Adjustment of Area 19 of B Signal GAM_B_LUT11: Gain Adjustment of Area 21 of B Signal GAM_B_LUT12: Gain Adjustment of Area 23 of B Signal GAM_B_LUT13: Gain Adjustment of Area 25 of B Signal GAM_B_LUT14: Gain Adjustment of Area 27 of B Signal GAM_B_LUT15: Gain Adjustment of Area 29 of B Signal GAM_B_LUT16: Gain Adjustment of Area 31 of B Signal Unsigned (0 to 2047 [LSB], 1024 [LSB] = 1.0 [times]) *: Bit Name GAM_B_LUT1: GAM_B_GAIN_01[10:0] GAM_B_LUT2: GAM_B_GAIN_03[10:0] GAM_B_LUT3: GAM_B_GAIN_05[10:0] GAM_B_LUT4: GAM_B_GAIN_07[10:0] GAM_B_LUT5: GAM_B_GAIN_09[10:0] GAM_B_LUT6: GAM_B_GAIN_11[10:0] GAM_B_LUT7: GAM_B_GAIN_13[10:0] GAM_B_LUT8: GAM_B_GAIN_15[10:0] GAM_B_LUT9: GAM_B_GAIN_17[10:0] GAM_B_LUT10: GAM_B_GAIN_19[10:0] GAM_B_LUT11: GAM_B_GAIN_21[10:0] GAM_B_LUT12: GAM_B_GAIN_23[10:0] GAM_B_LUT13: GAM_B_GAIN_25[10:0] GAM_B_LUT14: GAM_B_GAIN_27[10:0] GAM_B_LUT15: GAM_B_GAIN_29[10:0] GAM_B_LUT16: GAM_B_GAIN_31[10:0]

Note: This register is updated when GAM_B_VEN in GAM_B_UPDATE is 1.

chanhvinh.com

34.2.14 Area Setting Register B1 in Gamma Correction Block (GAM_B_AREA1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	GAM_B_TH_01[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_B_TH_02[7:0]								GAM_B_TH_03[7:0]							
Initial value:	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	GAM_B_TH_01[7:0]	8	R/W	Start Threshold of Area 1 of B Signal Unsigned (0 to 255 [LSB]) 0 < Threshold of current area < Threshold of next area
15 to 8	GAM_B_TH_02[7:0]	16	R/W	Start Threshold of Area 2 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_B_TH_03[7:0]	24	R/W	Start Threshold of Area 3 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_B_VEN in GAM_B_UPDATE is 1.

chanhvinh.com

34.2.15 Area Setting Register B2 in Gamma Correction Block (GAM_B_AREA2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_B_TH_04[7:0]								GAM_B_TH_05[7:0]							
Initial value:	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_B_TH_06[7:0]								GAM_B_TH_07[7:0]							
Initial value:	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_B_TH_04[7:0]	32	R/W	Start Threshold of Area 4 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_B_TH_05[7:0]	40	R/W	Start Threshold of Area 5 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_B_TH_06[7:0]	48	R/W	Start Threshold of Area 6 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_B_TH_07[7:0]	56	R/W	Start Threshold of Area 7 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_B_VEN in GAM_B_UPDATE is 1.

chanhvinh.com

34.2.16 Area Setting Register B3 in Gamma Correction Block (GAM_B_AREA3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_B_TH_08[7:0]								GAM_B_TH_09[7:0]							
Initial value:	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_B_TH_10[7:0]								GAM_B_TH_11[7:0]							
Initial value:	0	1	0	1	0	0	0	0	0	1	0	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_B_TH_08[7:0]	64	R/W	Start Threshold of Area 8 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_B_TH_09[7:0]	72	R/W	Start Threshold of Area 9 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_B_TH_10[7:0]	80	R/W	Start Threshold of Area 10 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_B_TH_11[7:0]	88	R/W	Start Threshold of Area 11 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_B_VEN in GAM_B_UPDATE is 1.

chanhvinh.com

34.2.17 Area Setting Register B4 in Gamma Correction Block (GAM_B_AREA4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_B_TH_12[7:0]								GAM_B_TH_13[7:0]							
Initial value:	0	1	1	0	0	0	0	0	0	1	1	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_B_TH_14[7:0]								GAM_B_TH_15[7:0]							
Initial value:	0	1	1	1	0	0	0	0	0	1	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_B_TH_12[7:0]	96	R/W	Start Threshold of Area 12 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_B_TH_13[7:0]	104	R/W	Start Threshold of Area 13 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_B_TH_14[7:0]	112	R/W	Start Threshold of Area 14 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_B_TH_15[7:0]	120	R/W	Start Threshold of Area 15 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_B_VEN in GAM_B_UPDATE is 1.

chanhvinh.com

34.2.18 Area Setting Register B5 in Gamma Correction Block (GAM_B_AREA5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_B_TH_16[7:0]								GAM_B_TH_17[7:0]							
Initial value:	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_B_TH_18[7:0]								GAM_B_TH_19[7:0]							
Initial value:	1	0	0	1	0	0	0	0	1	0	0	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_B_TH_16[7:0]	128	R/W	Start Threshold of Area 16 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_B_TH_17[7:0]	136	R/W	Start Threshold of Area 17 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_B_TH_18[7:0]	144	R/W	Start Threshold of Area 18 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_B_TH_19[7:0]	152	R/W	Start Threshold of Area 19 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_B_VEN in GAM_B_UPDATE is 1.

chanhvinh.com

34.2.19 Area Setting Register B6 in Gamma Correction Block (GAM_B_AREA6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_B_TH_20[7:0]								GAM_B_TH_21[7:0]							
Initial value:	1	0	1	0	0	0	0	0	1	0	1	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_B_TH_22[7:0]								GAM_B_TH_23[7:0]							
Initial value:	1	0	1	1	0	0	0	0	1	0	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_B_TH_20[7:0]	160	R/W	Start Threshold of Area 20 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_B_TH_21[7:0]	168	R/W	Start Threshold of Area 21 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_B_TH_22[7:0]	176	R/W	Start Threshold of Area 22 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_B_TH_23[7:0]	184	R/W	Start Threshold of Area 23 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_B_VEN in GAM_B_UPDATE is 1.

chanhvinh.com

34.2.20 Area Setting Register B7 in Gamma Correction Block (GAM_B_AREA7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_B_TH_24[7:0]								GAM_B_TH_25[7:0]							
Initial value:	1	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_B_TH_26[7:0]								GAM_B_TH_27[7:0]							
Initial value:	1	1	0	1	0	0	0	0	1	1	0	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_B_TH_24[7:0]	192	R/W	Start Threshold of Area 24 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_B_TH_25[7:0]	200	R/W	Start Threshold of Area 25 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_B_TH_26[7:0]	208	R/W	Start Threshold of Area 26 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_B_TH_27[7:0]	216	R/W	Start Threshold of Area 27 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_B_VEN in GAM_B_UPDATE is 1.

chanhvinh.com

34.2.21 Area Setting Register B8 in Gamma Correction Block (GAM_B_AREA8)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_B_TH_28[7:0]								GAM_B_TH_29[7:0]							
Initial value:	1	1	1	0	0	0	0	0	1	1	1	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_B_TH_30[7:0]								GAM_B_TH_31[7:0]							
Initial value:	1	1	1	1	0	0	0	0	1	1	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_B_TH_28[7:0]	224	R/W	Start Threshold of Area 28 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_B_TH_29[7:0]	232	R/W	Start Threshold of Area 29 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_B_TH_30[7:0]	240	R/W	Start Threshold of Area 30 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_B_TH_31[7:0]	248	R/W	Start Threshold of Area 31 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area ≤ 255

Note: This register is updated when GAM_B_VEN in GAM_B_UPDATE is 1.

chanhvinh.com

34.2.22 Register Update Control Register R in Gamma Correction Block (GAM_R_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GAM_R_VEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GAM_R_VEN	0	R/WC1	Gamma Correction (R) Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

chanhvinh.com

34.2.23 Table Setting Register R1 to R16 in Gamma Correction Block (GAM_R_LUT1 to GAM_R_LUT16)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GAM_G_GAIN_xx[10:0]										
Initial value:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GAM_G_GAIN_yy[10:0]										
Initial value:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	*	1024	R/W	GAM_R_LUT1: Gain Adjustment of Area 0 of R Signal GAM_R_LUT2: Gain Adjustment of Area 2 of R Signal GAM_R_LUT3: Gain Adjustment of Area 4 of R Signal GAM_R_LUT4: Gain Adjustment of Area 6 of R Signal GAM_R_LUT5: Gain Adjustment of Area 8 of R Signal GAM_R_LUT6: Gain Adjustment of Area 10 of R Signal GAM_R_LUT7: Gain Adjustment of Area 12 of R Signal GAM_R_LUT8: Gain Adjustment of Area 14 of R Signal GAM_R_LUT9: Gain Adjustment of Area 16 of R Signal GAM_R_LUT10: Gain Adjustment of Area 18 of R Signal GAM_R_LUT11: Gain Adjustment of Area 20 of R Signal GAM_R_LUT12: Gain Adjustment of Area 22 of R Signal GAM_R_LUT13: Gain Adjustment of Area 24 of R Signal GAM_R_LUT14: Gain Adjustment of Area 26 of R Signal GAM_R_LUT15: Gain Adjustment of Area 28 of R Signal GAM_R_LUT16: Gain Adjustment of Area 30 of R Signal Unsigned (0 to 2047 [LSB], 1024 [LSB] = 1.0 [times]) *: Bit Name GAM_R_LUT1: GAM_R_GAIN_00[10:0] GAM_R_LUT2: GAM_R_GAIN_02[10:0] GAM_R_LUT3: GAM_R_GAIN_04[10:0] GAM_R_LUT4: GAM_R_GAIN_06[10:0] GAM_R_LUT5: GAM_R_GAIN_08[10:0] GAM_R_LUT6: GAM_R_GAIN_10[10:0] GAM_R_LUT7: GAM_R_GAIN_12[10:0] GAM_R_LUT8: GAM_R_GAIN_14[10:0] GAM_R_LUT9: GAM_R_GAIN_16[10:0] GAM_R_LUT10: GAM_R_GAIN_18[10:0] GAM_R_LUT11: GAM_R_GAIN_20[10:0] GAM_R_LUT12: GAM_R_GAIN_22[10:0] GAM_R_LUT13: GAM_R_GAIN_24[10:0] GAM_R_LUT14: GAM_R_GAIN_26[10:0] GAM_R_LUT15: GAM_R_GAIN_28[10:0] GAM_R_LUT16: GAM_R_GAIN_30[10:0]
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
10 to 0	*	1024	R/W	GAM_R_LUT1: Gain Adjustment of Area 1 of R Signal GAM_R_LUT2: Gain Adjustment of Area 3 of R Signal GAM_R_LUT3: Gain Adjustment of Area 5 of R Signal GAM_R_LUT4: Gain Adjustment of Area 7 of R Signal GAM_R_LUT5: Gain Adjustment of Area 9 of R Signal GAM_R_LUT6: Gain Adjustment of Area 11 of R Signal GAM_R_LUT7: Gain Adjustment of Area 13 of R Signal GAM_R_LUT8: Gain Adjustment of Area 15 of R Signal GAM_R_LUT9: Gain Adjustment of Area 17 of R Signal GAM_R_LUT10: Gain Adjustment of Area 19 of R Signal GAM_R_LUT11: Gain Adjustment of Area 21 of R Signal GAM_R_LUT12: Gain Adjustment of Area 23 of R Signal GAM_R_LUT13: Gain Adjustment of Area 25 of R Signal GAM_R_LUT14: Gain Adjustment of Area 27 of R Signal GAM_R_LUT15: Gain Adjustment of Area 29 of R Signal GAM_R_LUT16: Gain Adjustment of Area 31 of R Signal Unsigned (0 to 2047 [LSB], 1024 [LSB] = 1.0 [times]) *: Bit Name GAM_R_LUT1: GAM_R_GAIN_01[10:0] GAM_R_LUT2: GAM_R_GAIN_03[10:0] GAM_R_LUT3: GAM_R_GAIN_05[10:0] GAM_R_LUT4: GAM_R_GAIN_07[10:0] GAM_R_LUT5: GAM_R_GAIN_09[10:0] GAM_R_LUT6: GAM_R_GAIN_11[10:0] GAM_R_LUT7: GAM_R_GAIN_13[10:0] GAM_R_LUT8: GAM_R_GAIN_15[10:0] GAM_R_LUT9: GAM_R_GAIN_17[10:0] GAM_R_LUT10: GAM_R_GAIN_19[10:0] GAM_R_LUT11: GAM_R_GAIN_21[10:0] GAM_R_LUT12: GAM_R_GAIN_23[10:0] GAM_R_LUT13: GAM_R_GAIN_25[10:0] GAM_R_LUT14: GAM_R_GAIN_27[10:0] GAM_R_LUT15: GAM_R_GAIN_29[10:0] GAM_R_LUT16: GAM_R_GAIN_31[10:0]

Note: This register is updated when GAM_R_VEN in GAM_R_UPDATE is 1.

chanhvinh.com

34.2.24 Area Setting Register R1 in Gamma Correction Block (GAM_R_AREA1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	GAM_R_TH_01[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_R_TH_02[7:0]								GAM_R_TH_03[7:0]							
Initial value:	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	GAM_R_TH_01[7:0]	8	R/W	Start Threshold of Area 1 of R Signal Unsigned (0 to 255 [LSB]) 0 < Threshold of current area < Threshold of next area
15 to 8	GAM_R_TH_02[7:0]	16	R/W	Start Threshold of Area 2 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_R_TH_03[7:0]	24	R/W	Start Threshold of Area 3 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_R_VEN in GAM_R_UPDATE is 1.

chanhvinh.com

34.2.25 Area Setting Register R2 in Gamma Correction Block (GAM_R_AREA2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_R_TH_04[7:0]								GAM_R_TH_05[7:0]							
Initial value:	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_R_TH_06[7:0]								GAM_R_TH_07[7:0]							
Initial value:	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_R_TH_04[7:0]	32	R/W	Start Threshold of Area 4 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_R_TH_05[7:0]	40	R/W	Start Threshold of Area 5 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_R_TH_06[7:0]	48	R/W	Start Threshold of Area 6 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_R_TH_07[7:0]	56	R/W	Start Threshold of Area 7 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_R_VEN in GAM_R_UPDATE is 1.

chanhvinh.com

34.2.26 Area Setting Register R3 in Gamma Correction Block (GAM_R_AREA3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_R_TH_08[7:0]								GAM_R_TH_09[7:0]							
Initial value:	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_R_TH_10[7:0]								GAM_R_TH_11[7:0]							
Initial value:	0	1	0	1	0	0	0	0	0	1	0	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_R_TH_08[7:0]	64	R/W	Start Threshold of Area 8 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_R_TH_09[7:0]	72	R/W	Start Threshold of Area 9 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_R_TH_10[7:0]	80	R/W	Start Threshold of Area 10 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_R_TH_11[7:0]	88	R/W	Start Threshold of Area 11 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_R_VEN in GAM_R_UPDATE is 1.

chanhvinh.com

34.2.27 Area Setting Register R4 in Gamma Correction Block (GAM_R_AREA4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_R_TH_12[7:0]								GAM_R_TH_13[7:0]							
Initial value:	0	1	1	0	0	0	0	0	0	1	1	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_R_TH_14[7:0]								GAM_R_TH_15[7:0]							
Initial value:	0	1	1	1	0	0	0	0	0	1	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_R_TH_12[7:0]	96	R/W	Start Threshold of Area 12 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_R_TH_13[7:0]	104	R/W	Start Threshold of Area 13 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_R_TH_14[7:0]	112	R/W	Start Threshold of Area 14 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_R_TH_15[7:0]	120	R/W	Start Threshold of Area 15 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_R_VEN in GAM_R_UPDATE is 1.

chanhvinh.com

34.2.28 Area Setting Register R5 in Gamma Correction Block (GAM_R_AREA5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_R_TH_16[7:0]								GAM_R_TH_17[7:0]							
Initial value:	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_R_TH_18[7:0]								GAM_R_TH_19[7:0]							
Initial value:	1	0	0	1	0	0	0	0	1	0	0	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_R_TH_16[7:0]	128	R/W	Start Threshold of Area 16 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_R_TH_17[7:0]	136	R/W	Start Threshold of Area 17 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_R_TH_18[7:0]	144	R/W	Start Threshold of Area 18 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_R_TH_19[7:0]	152	R/W	Start Threshold of Area 19 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_R_VEN in GAM_R_UPDATE is 1.

chanhvinh.com

34.2.29 Area Setting Register R6 in Gamma Correction Block (GAM_R_AREA6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_R_TH_20[7:0]								GAM_R_TH_21[7:0]							
Initial value:	1	0	1	0	0	0	0	0	1	0	1	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_R_TH_22[7:0]								GAM_R_TH_23[7:0]							
Initial value:	1	0	1	1	0	0	0	0	1	0	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_R_TH_20[7:0]	160	R/W	Start Threshold of Area 20 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_R_TH_21[7:0]	168	R/W	Start Threshold of Area 21 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_R_TH_22[7:0]	176	R/W	Start Threshold of Area 22 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_R_TH_23[7:0]	184	R/W	Start Threshold of Area 23 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_R_VEN in GAM_R_UPDATE is 1.

chanhvinh.com

34.2.30 Area Setting Register R7 in Gamma Correction Block (GAM_R_AREA7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_R_TH_24[7:0]								GAM_R_TH_25[7:0]							
Initial value:	1	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_R_TH_26[7:0]								GAM_R_TH_27[7:0]							
Initial value:	1	1	0	1	0	0	0	0	1	1	0	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_R_TH_24[7:0]	192	R/W	Start Threshold of Area 24 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_R_TH_25[7:0]	200	R/W	Start Threshold of Area 25 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_R_TH_26[7:0]	208	R/W	Start Threshold of Area 26 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_R_TH_27[7:0]	216	R/W	Start Threshold of Area 27 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_R_VEN in GAM_R_UPDATE is 1.

chanhvinh.com

34.2.31 Area Setting Register R8 in Gamma Correction Block (GAM_R_AREA8)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_R_TH_28[7:0]								GAM_R_TH_29[7:0]							
Initial value:	1	1	1	0	0	0	0	0	1	1	1	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_R_TH_30[7:0]								GAM_R_TH_31[7:0]							
Initial value:	1	1	1	1	0	0	0	0	1	1	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_R_TH_28[7:0]	224	R/W	Start Threshold of Area 28 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_R_TH_29[7:0]	232	R/W	Start Threshold of Area 29 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_R_TH_30[7:0]	240	R/W	Start Threshold of Area 30 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_R_TH_31[7:0]	248	R/W	Start Threshold of Area 31 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area ≤ 255

Note: This register is updated when GAM_R_VEN in GAM_R_UPDATE is 1.

chanhvinh.com

34.2.32 TCON Register Update Control Register (TCON_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TCON_VEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TCON_VEN	0	R/WC1	LCD TCON Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

34.2.33 TCON Reference Timing Setting Register (TCON_TIM)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TCON_HALF[10:0]										
Initial value:	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TCON_OFFSET[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	TCON_HALF[10:0]	400	R/W	1/2fH Timing Specifies the clock count from the rising edge of the Hsync signal as the counting timing of horizontal counter.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	TCON_OFFSET[10:0]	0	R/W	Offset Hsync Signal Timing Sets the clock cycle count from the rising edge of the Hsync signal.

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

chanhvinh.com

34.2.34 TCON Vertical Timing Setting Register A1 (TCON_TIM_STVA1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TCON_STVA_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TCON_STVA_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	TCON_STVA_VS [10:0]	0	R/W	STVA Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_STVA_VS from the rising edge of the Vsync signal (1/2fH cycles).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	TCON_STVA_VW [10:0]	4	R/W	STVA Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_STVA_VW (1/2fH cycles).

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

chanhvinh.com

34.2.35 TCON Vertical Timing Setting Register A2 (TCON_TIM_STVA2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	TCON_STVA_INV	—	TCON_STVA_SEL[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	TCON_STVA_INV	1	R/W	Polarity Inversion Control of STVA Signal 0: Not inverted 1: Inverted
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	TCON_STVA_SEL [2:0]	0	R/W	Output Signal Select for LCD_TCON0 Pin 0: STVA/VS 1: STVB/VE 2: STH/SP/HS 3: STB/LP/HE 4: CPV/GCK 5: POLA 6: POLB 7: DE

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

chanhvinh.com

34.2.36 TCON Vertical Timing Setting Register B1 (TCON_TIM_STVB1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TCON_STVB_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TCON_STVB_VW[10:0]										
Initial value:	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	TCON_STVB_VS [10:0]	70	R/W	STVB Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_STVB_VS from the rising edge of the Vsync signal (1/2fH cycles).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	TCON_STVB_VW [10:0]	960	R/W	STVB Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_STVB_VW (1/2fH cycles).

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

chanhvinh.com

34.2.37 TCON Vertical Timing Setting Register B2 (TCON_TIM_STVB2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	TCON_STVB_INV	—	TCON_STVB_SEL[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	TCON_STVB_INV	0	R/W	Polarity Inversion Control of STVB Signal 0: Not inverted 1: Inverted
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	TCON_STVB_SEL [2:0]	1	R/W	Output Signal Select for LCD_TCON1 Pin 0: STVA/VS 1: STVB/VE 2: STH/SP/HS 3: STB/LP/HE 4: CPV/GCK 5: POLA 6: POLB 7: DE

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

chanhvinh.com

34.2.38 TCON Horizontal Timing Setting Register STH1 (TCON_TIM_STH1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TCON_STH_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TCON_STH_HW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	TCON_STH_HS [10:0]	0	R/W	STH Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_STH_HS + 1 from the rising edge of the Hsync signal (clock cycles).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	TCON_STH_HW [10:0]	96	R/W	STH Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_STH_HW (clock cycles).

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

chanhvinh.com

34.2.39 TCON Horizontal Timing Setting Register STH2 (TCON_TIM_STH2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TCON_STH_HS_SEL	—	—	—	TCON_STH_INV	—	TCON_STH_SEL[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	TCON_STH_HS_SEL	0	R/W	STH Signal Operating Reference Select 0: Hsync signal reference 1: Offset Hsync signal reference
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	TCON_STH_INV	1	R/W	Polarity Inversion Control of STH Signal 0: Not inverted 1: Inverted
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	TCON_STH_SEL[2:0]	2	R/W	Output Signal Select for LCD_TCON2 Pin 0: STVA/VS 1: STVB/VE 2: STH/SP/HS 3: STB/LP/HE 4: CPV/GCK 5: POLA 6: POLB 7: DE

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

chanhvinh.com

34.2.40 TCON Horizontal Timing Setting Register STB1 (TCON_TIM_STB1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TCON_STB_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TCON_STB_HW[10:0]										
Initial value:	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	TCON_STB_HS [10:0]	144	R/W	STB Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_STB_HS + 1 from the rising edge of the Hsync signal (clock cycles).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	TCON_STB_HW [10:0]	640	R/W	STB Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_STB_HW (clock cycles).

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

chanhvinh.com

34.2.41 TCON Horizontal Timing Setting Register STB2 (TCON_TIM_STB2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TCON_STB_HS_SEL	—	—	—	TCON_STB_INV	—	TCON_STB_SEL[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	TCON_STB_HS_SEL	0	R/W	STB Signal Operating Reference Select 0: Hsync signal reference 1: Offset Hsync signal reference
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	TCON_STB_INV	0	R/W	Polarity Inversion Control of STB Signal 0: Not inverted 1: Inverted
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	TCON_STB_SEL[2:0]	7	R/W	Output Signal Select for LCD_TCON3 Pin 0: STVA/VS 1: STVB/VE 2: STH/SP/HS 3: STB/LP/HE 4: CPV/GCK 5: POLA 6: POLB 7: DE

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

chanhvinh.com

34.2.42 TCON Horizontal Timing Setting Register CPV1 (TCON_TIM_CPV1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TCON_CPV_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TCON_CPV_HW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	TCON_CPV_HS [10:0]	0	R/W	CPV Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_CPV_HS + 1 from the rising edge of the Hsync signal (clock cycles).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	TCON_CPV_HW [10:0]	0	R/W	CPV Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_CPV_HW (clock cycles).

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

chanhvinh.com

34.2.43 TCON Horizontal Timing Setting Register CPV2 (TCON_TIM_CPV2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TCON_CPV_HS_SEL	—	—	—	TCON_CPV_INV	—	TCON_CPV_SEL[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	TCON_CPV_HS_SEL	0	R/W	CPV Signal Operating Reference Select 0: Hsync signal reference 1: Offset Hsync signal reference
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	TCON_CPV_INV	0	R/W	Polarity Inversion Control of CPV Signal 0: Not inverted 1: Inverted
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	TCON_CPV_SEL[2:0]	4	R/W	Output Signal Select for LCD_TCON4 Pin 0: STVA/VS 1: STVB/VE 2: STH/SP/HS 3: STB/LP/HE 4: CPV/GCK 5: POLA 6: POLB 7: DE

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

chanhvinh.com

34.2.44 TCON Horizontal Timing Setting Register POLA1 (TCON_TIM_POLA1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TCON_POLA_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TCON_POLA_HW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	TCON_POLA_HS [10:0]	0	R/W	POLA Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_POLA_HS from the rising edge of the Hsync signal (clock cycles). Note: When 1 × 1, 1 × 2, or 2 × 2 reverse mode is selected, these bits should be set to1 or greater.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	TCON_POLA_HW [10:0]	0	R/W	POLA Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_POLA_HW (clock cycles).

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

chanhvinh.com

34.2.45 TCON Horizontal Timing Setting Register POLA2 (TCON_TIM_POLA2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	TCON_POLA_MD[1:0]		—	—	—	TCON_POLA_HS_SEL	—	—	—	TCON_POLA_INV	—	TCON_POLA_SEL[2:0]		
Initial value:	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1
R/W:	R	R	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	TCON_POLA_MD [1:0]	1	R/W	POLA Signal Generation Mode Select 0: Normal mode Generates the signal that changes twice a horizontal period. 1: 1 × 1 reverse mode Generates the signal whose polarity is inverted every horizontal period. 2: 1 × 2 reverse mode Generates the signal whose polarity is inverted in the first horizontal period and is subsequently inverted every two horizontal periods. 3: 2 × 2 reverse mode Generates the signal whose polarity is inverted every two horizontal periods.
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	TCON_POLA_HS _SEL	0	R/W	POLA Signal Operating Reference Select 0: Hsync signal reference 1: Offset Hsync signal reference
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	TCON_POLA_INV	0	R/W	Polarity Inversion Control of POLA Signal 0: Not inverted 1: Inverted
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	TCON_POLA_SEL [2:0]	5	R/W	Output Signal Select for LCD_TCON5 pin 0: STVA/VS 1: STVB/VE 2: STH/SP/HS 3: STB/LP/HE 4: CPV/GCK 5: POLA 6: POLB 7: DE

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

chanhvinh.com

34.2.46 TCON Horizontal Timing Setting Register POLB1 (TCON_TIM_POLB1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TCON_POLB_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TCON_POLB_HW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	TCON_POLB_HS [10:0]	0	R/W	POLB Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_POLB_HS from the rising edge of the Hsync signal (clock cycles). Note: When 1 × 1, 1 × 2, or 2 × 2 reverse mode is selected, these bits should be set to1 or greater.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	TCON_POLB_HW [10:0]	0	R/W	POLB Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_POLB_HW (clock cycles).

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

chanhvinh.com

34.2.47 TCON Horizontal Timing Setting Register POLB2 (TCON_TIM_POLB2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	TCON_POLB_MD[1:0]		—	—	—	TCON_POLB_HS_SEL	—	—	—	TCON_POLB_INV	—	TCON_POLB_SEL[2:0]		
Initial value:	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0
R/W:	R	R	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	TCON_POLB_MD [1:0]	1	R/W	POLB Signal Generation Mode Select 0: Normal mode Generates the signal that changes twice a horizontal period. 1: 1 × 1 reverse mode Generates the signal whose polarity is inverted every horizontal period. 2: 1 × 2 reverse mode Generates the signal whose polarity is inverted in the first horizontal period and is subsequently inverted every two horizontal periods. 3: 2 × 2 reverse mode Generates the signal whose polarity is inverted every two horizontal periods.
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	TCON_POLB_HS _SEL	0	R/W	POLB Signal Operating Reference Select 0: Hsync signal reference 1: Offset Hsync signal reference
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	TCON_POLB_INV	0	R/W	Polarity Inversion Control of POLB Signal 0: Not inverted 1: Inverted
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	TCON_POLB_SEL [2:0]	6	R/W	Output Signal Select for LCD_TCON6 pin 0: STVA/VS 1: STVB/VE 2: STH/SP/HS 3: STB/LP/HE 4: CPV/GCK 5: POLA 6: POLB 7: DE

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

chanhvinh.com

34.2.48 TCON Data Enable Polarity Setting Register (TCON_TIM_DE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TCON_DE_INV
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TCON_DE_INV	0	R/W	Polarity Inversion Control of DE Signal 0: Not inverted 1: Inverted

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

34.2.49 Register Update Control Register in Output Controller (OUT_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OUTCNT_VEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	OUTCNT_VEN	0	R/WC1	Brightness/Contrast, Dither Process, Output Interface Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

chanhvinh.com

34.2.50 Output Interface Register (OUT_SET)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	OUT_ENDIAN_ON	—	—	—	OUT_SWAP_ON	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	OUT_FORMAT[1:0]		—	—	OUT_FRQ_SEL[1:0]		—	—	—	OUT_DIR_SEL	—	—	OUT_PHASE[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28	OUT_ENDIAN_ON	0	R/W	Bit Endian Change On/Off Control 0: Off 1: On
27 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	OUT_SWAP_ON	0	R/W	B/R Signal Swap On/Off Control 0: Off 1: On
23 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20 to 16	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	OUT_FORMAT[1:0]	0	R/W	Output Format Select 0: RBG888 1: RGB666 2: RGB565 3: Serial RGB
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	OUT_FRQ_SEL[1:0]	0	R/W	Clock Frequency Control 0: 100% speed — (parallel RGB) 1: Triple speed — (serial RGB) 2: Quadruple speed — (serial RGB) 3: Setting prohibited
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	OUT_DIR_SEL	0	R/W	Scan Direction Select 0: Forward scan 1: Reverse scan
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
1, 0	OUT_PHASE[1:0]	0	R/W	Clock Phase Adjustment During Serial RGB Output Triple speed mode 0: 0 (clk) 1: 1 (clk) 2: 2 (clk) 3: Setting prohibited Quadruple speed mode 0: 0 (clk) 1: 1 (clk) 2: 2 (clk) 3: 3 (clk)

Note: This register is updated when OUTCNT_VEN in OUT_UPDATE is 1.

34.2.51 Brightness (DC) Correction Register 1 (OUT_BRIGHT1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PBRT_G[9:0]									
Initial value:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	PBRT_G[9:0]	512	R/W	Brightness (DC) Adjustment of G Signal Unsigned (0 (-512) to 512 (0) to 1023 (+511) [LSB], 512 [LSB] with offset)

Note: This register is updated when OUTCNT_VEN in OUT_UPDATE is 1.

chanhvinh.com

34.2.52 Brightness (DC) Correction Register 2 (OUT_BRIGHT2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	PBRT_B[9:0]									
Initial value:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PBRT_R[9:0]									
Initial value:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 16	PBRT_B[9:0]	512	R/W	Brightness (DC) Adjustment of B Signal Unsigned (0 (-512) to 512 (0) to 1023 (+511) [LSB], 512 [LSB] with offset)
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	PBRT_R[9:0]	512	R/W	Brightness (DC) Adjustment of R Signal Unsigned (0 (-512) to 512 (0) to 1023 (+511) [LSB], 512 [LSB] with offset)

Note: This register is updated when OUTCNT_VEN in OUT_UPDATE is 1.

34.2.53 Contrast (Gain) Correction Register (OUT_CONTRAST)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CONT_G[7:0]							
Initial value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CONT_B[7:0]								CONT_R[7:0]							
Initial value:	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	CONT_G[7:0]	128	R/W	Contrast (Gain) Adjustment of G Signal 0/128 to 255/128 (approx.2 times)
15 to 8	CONT_B[7:0]	128	R/W	Contrast (Gain) Adjustment of B Signal 0/128 to 255/128 (approx.2 times)
7 to 0	CONT_R[7:0]	128	R/W	Contrast (Gain) Adjustment of R Signal 0/128 to 255/128 (approx.2 times)

Note: This register is updated when OUTCNT_VEN in OUT_UPDATE is 1.

chanhvinh.com

34.2.54 Panel Dither Register (OUT_PDTHA)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	PDTH_SEL[1:0]		—	—	PDTH_FORMAT[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PDTH_PA[1:0]		—	—	PDTH_PB[1:0]		—	—	PDTH_PC[1:0]		—	—	PDTH_PD[1:0]	
Initial value:	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	1
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21, 20	PDTH_SEL[1:0]	0	R/W	Panel Dither Operation Mode 0: Truncate 1: Round-off 2: 2 × 2 pattern dither 3: Random pattern dither
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	PDTH_FORMAT [1:0]	0	R/W	Panel Dither Output Format Select 0: RGB888 1: RGB666 2: RGB565 3: Setting prohibited
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	PDTH_PA[1:0]	3	R/W	Pattern Value (A) of 2 × 2 Pattern Dither Unsigned (0 to 3 [LSB])
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PDTH_PB[1:0]	0	R/W	Pattern Value (B) of 2 × 2 Pattern Dither Unsigned (0 to 3 [LSB])
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PDTH_PC[1:0]	2	R/W	Pattern Value (C) of 2 × 2 Pattern Dither Unsigned (0 to 3 [LSB])
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PDTH_PD[1:0]	1	R/W	Pattern Value (D) of 2 × 2 Pattern Dither Unsigned (0 to 3 [LSB])

Note: This register is updated when OUTCNT_VEN in OUT_UPDATE is 1.

chanhvinh.com

34.2.55 Output Phase Control Register (OUT_CLK_PHASE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	OUTCNT_FRONT_GAM	—	—	—	OUTCNT_LCD_EDGE	—	OUTCNT_STVA_EDGE	OUTCNT_STVB_EDGE	OUTCNT_STH_EDGE	OUTCNT_STB_EDGE	OUTCNT_CPV_EDGE	OUTCNT_POLA_EDGE	OUTCNT_POLB_EDGE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	OUTCNT_FRONT_GAM	0	R/W	Correction Circuit Sequence Control 0: Brightness → contrast → gamma correction 1: Gamma correction → brightness → contrast
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	OUTCNT_LCD_EDGE	0	R/W	Output Phase Control of LCD_DATA23 to LCD_DATA0 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	OUTCNT_STVA_EDGE	0	R/W	Output Phase Control of LCD_TCON0 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
5	OUTCNT_STVB_EDGE	0	R/W	Output Phase Control of LCD_TCON1 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
4	OUTCNT_STH_EDGE	0	R/W	Output Phase Control of LCD_TCON2 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
3	OUTCNT_STB_EDGE	0	R/W	Output Phase Control of LCD_TCON3 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
2	OUTCNT_CPV_EDGE	0	R/W	Output Phase Control of LCD_TCON4 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
1	OUTCNT_POLA_EDGE	0	R/W	Output Phase Control of LCD_TCON5 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
0	OUTCNT_POLB_EDGE	0	R/W	Output Phase Control of LCD_TCON6 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin

Note: This register is updated when OUTCNT_VEN in OUT_UPDATE is 1.

chanhvinh.com

34.3 Usage Methods

34.3.1 Gamma Correction Adjustment Method

The characteristics of G, B, R of each panel to be connected should be measured and gamma correction should be made to suit the panel.

Since the gamma correction adjustment depends on to the characteristics of the panel, there are no recommended setting values.

34.3.2 Dither Usage Method

Dither is used when pseudo contour is appeared on the display screen.

Table 34.26 Dither Settings

Bit Name	Setting Value
PDTH_FORMAT[1:0]	Selects the format. For RGB888: 0 For RGB666: 1 For RGB565: 2
PDTH_SEL[1:0]	When 2 × 2 pattern dither is to be used: 2
PDTH_PA[1:0]	Normally 3 (initial value)
PDTH_PB[1:0]	Normally 0 (initial value)
PDTH_PC[1:0]	Normally 2 (initial value)
PDTH_PD[1:0]	Normally 1 (initial value)

chanhvinh.com

34.3.3 Output Format Adjustment Method

The setting example of the typical output format is shown in Table 34.27 and Table 34.28.

It is necessary to carry out the setting of synchronization system signals of each output format similarly as the setting of output after scaling.

Table 34.27 Setting Example of Synchronizing Signal

Register Name	Bit Name	VGA	SVGA	Description
TCON_TIM	TCON_HALF[10:0]	400	528	Sets the half value of 1H time period in clock units.
Vsync signal				
TCON_TIM_STVA1	TCON_STVA_VS[10:0]	0	0	Sets the pulse generation start position from the rising edge of the internal Vsync signal. A 1/2H time period is set as 1.
TCON_TIM_STVA1	TCON_STVA_VW[10:0]	4	8	Sets the changing point from the above pulse generation start position. A 1/2H time period is set as 1.
TCON_TIM_STVA2	TCON_STVA_INV	0	0	Sets the output polarity of the above pulse. For inverted output: 1
TCON_TIM_STVA2	TCON_STVA_SEL[2:0]	0	0	For STVA output selection: 0
Vertical enable signal				
TCON_TIM_STVB1	TCON_STVB_VS[10:0]	68	44	Sets the pulse generation start position from the rising edge of the internal Vsync signal. A 1/2H time period is set as 1.
TCON_TIM_STVB1	TCON_STVB_VW[10:0]	960	1200	Sets the changing point from the above pulse generation start position. A 1/2H time period is set as 1.
TCON_TIM_STVB2	TCON_STVB_INV	0	0	Sets the output polarity of the above pulse. For inverted output: 1
TCON_TIM_STVB2	TCON_STVB_SEL[2:0]	1	1	For STVB output selection: 1
Hsync signal				
TCON_TIM_STH1	TCON_STH_HS[10:0]	0	0	Sets the pulse generation start position from the rising edge of the internal Hsync signal.
TCON_TIM_STH1	TCON_STH_HW[10:0]	96	128	Sets the changing point from the above pulse generation start position.
TCON_TIM_STH2	TCON_STH_INV	0	0	Sets the output polarity of the above pulse. For inverted output: 1
TCON_TIM_STH2	TCON_STH_SEL[2:0]	2	2	For STH output selection: 2
Horizontal enable signal				
TCON_TIM_STB1	TCON_STB_HS[10:0]	128	192	Sets the pulse generation start position from the rising edge of the internal Hsync signal.
TCON_TIM_STB1	TCON_STB_HW[10:0]	640	800	Sets the changing point from the above pulse generation start position.
TCON_TIM_STB2	TCON_STB_INV	0	0	Sets the output polarity of the above pulse. For inverted output: 1
TCON_TIM_STB2	TCON_STB_SEL[2:0]	3	3	For STB output selection: 3

chanhvinh.com

Table 34.28 Setting Example of Data System

Register Name	Bit Name	RGB888	Serial RGB (Triple Speed)	Description
OUT_SET	OUT_ENDIAN_ON	0	0	When bit endian is changed: 1
OUT_SET	OUT_SWAP_ON	0	0	When B/R is to be swapped: 1
OUT_SET	OUT_PIXEL_INV_ON	0	0	When the function of reducing number of simultaneous changes is to be used: 1
OUT_SET	OUT_SUM_MOVE[4:0]	31	31	When OUT_PIXEL_INV_ON = 1, this register becomes valid. Sets the simultaneous changing threshold.
OUT_SET	OUT_FORMAT[1:0]	0	3	Sets the output format. For RGB888: 0 For RGB666: 1 For RGB565: 2 For serial RGB: 3
OUT_SET	OUT_FRQ_SEL[1:0]	0	1	Sets the output clock. For RGB888, RGB666, RGB565: 0 For triple speed serial RGB output: 1 For quadruple serial RGB output: 2
OUT_SET	OUT_DIR_SEL	0	0	When data arrangement of the serial RGB output is to be reversed: 1
OUT_SET	OUT_PHASE[1:0]	0	0	Sets when output phase of the serial RGB is shifted. Not delayed Delayed by one clock cycle: 1 Delayed by two clock cycles: 2 Delayed by three clock cycles: 3 (applicable only for quadruple speed mode)

chanhvinh.com

35. Video Display Controller 5 (8):System Controller

35.1 System Controller

35.1.1 Overview of Functions

The system controller provides interrupt control, panel clock control, CLUT table read select signal status flag output functions.

35.1.2 Interrupt Control

Ten interrupt signals are output from the scaler and image synthesizer. The system controller controls whether to output these interrupt signals.

One is written to the corresponding INT_STA* bit when an interrupt signal is to be accepted. After 1 has been written to the bit, however, its value is still read out as 0 until the interrupt signal is accepted. Once the interrupt signal has been accepted, the INT_STA* bit is read out as 1.

The bit for an accepted interrupt signal is cleared by writing 0 to the INT_STA* bit. If further interrupt signals are to be accepted after the INT_STA* bit has been cleared, 1 is again written to the bit.

Table 35.1 Interrupt Signals

Request Source Name	Bit Name	Function
S0_VI_VSYNC	INT_STA 0	Vsync signal input to scaler 0
S0_LO_VSYNC	INT_STA 1	Vsync signal output from scaler 0
S0_VSYNCERR	INT_STA 2	Missing Vsync signal for scaler 0
GR3_VLINE	INT_STA 3	Specified line signal for panel output in graphics 3
S0_VFIELD	INT_STA 4	Field end signal for recording function in scaler 0
IV1_VBUFERR	INT_STA 5	Frame buffer write overflow signal for scaler 0
IV3_VBUFERR	INT_STA 6	Frame buffer read underflow signal for graphics 0
IV5_VBUFERR	INT_STA 7	Frame buffer read underflow signal for graphics 2
IV6_VBUFERR	INT_STA 8	Frame buffer read underflow signal for graphics 3
S0_WLINE	INT_STA 9	Write specification line signal input to scaling-down control block in scaler 0 (This signal is not present in this product.)

chanhvinh.com

Table 35.2 Interrupt Clear/Hold Settings

Register Name	Bit Name	Initial Value	Description
SYSCNT_INT1	INT_STA0	0	S0_VI_VSYNC Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
SYSCNT_INT1	INT_STA1	0	S0_LO_VSYNC Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
SYSCNT_INT1	INT_STA2	0	S0_VSYNCERR Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
SYSCNT_INT1	INT_STA3	0	GR3_VLINE Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
SYSCNT_INT1	INT_STA4	0	S0_VFIELD Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
SYSCNT_INT1	INT_STA5	0	IV1_VBUFERR Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
SYSCNT_INT1	INT_STA6	0	IV3_VBUFERR Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
SYSCNT_INT1	INT_STA7	0	IV5_VBUFERR Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
SYSCNT_INT2	INT_STA8	0	IV6_VBUFERR Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
SYSCNT_INT2	INT_STA9	0	S0_WLINE Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. (Setting prohibited in this product) 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.

chanhvinh.com

Table 35.3 Interrupt Output On/Off Settings

Register Name	Bit Name	Initial Value	Description
SYSCNT_INT4	INT_OUT0_ON	0	S0_VI_VSYNC Interrupt Output On/Off 0: Off 1: On
SYSCNT_INT4	INT_OUT1_ON	0	S0_LO_VSYNC Interrupt Output On/Off 0: Off 1: On
SYSCNT_INT4	INT_OUT2_ON	0	S0_VSYNCERR Interrupt Output On/Off 0: Off 1: On
SYSCNT_INT4	INT_OUT3_ON	0	GR3_VLINE Interrupt Output On/Off 0: Off 1: On
SYSCNT_INT4	INT_OUT4_ON	0	S0_VFIELD Interrupt Output On/Off 0: Off 1: On
SYSCNT_INT4	INT_OUT5_ON	0	IV1_VBUFERR Interrupt Output On/Off 0: Off 1: On
SYSCNT_INT4	INT_OUT6_ON	0	IV3_VBUFERR Interrupt Output On/Off 0: Off 1: On
SYSCNT_INT4	INT_OUT7_ON	0	IV5_VBUFERR Interrupt Output On/Off 0: Off 1: On
SYSCNT_INT5	INT_OUT8_ON	0	IV6_VBUFERR Interrupt Output On/Off 0: Off 1: On
SYSCNT_INT5	INT_OUT9_ON	0	S0_WLINE Interrupt Output On/Off 0: Off 1: On (Setting prohibited in this product)

chanhvinh.com

35.1.3 Panel Clock Control

The video image clock, external clock, or peripheral clock 1 is selectable as the source of the panel clock for supply to this module. The module is also equipped with a divider for scaling the frequency by factors from 1/1 to 1/32. The bits listed in Table 35.4 control the panel clock.

Table 35.4 Panel Clock Control

Register Name	Bit Name	Initial Value	Description
SYSCNT_PANEL_CLK	PANEL_ICKSEL [1:0]	0	Divided Clock Source Select 0: Video image clock (this is DV_CLK if INP_SEL = 1) 1: External clock (LCD0_EXTCLK) 2: Setting prohibited 3: Peripheral clock 1 (P1φ)
SYSCNT_PANEL_CLK	PANEL_ICKEN	0	Panel Clock Operation Enable 0: Disables operation of the panel clock operation block. 1: Enables operation of the panel clock operation block. Note: Set this bit to 0 before changing the value of the PANEL_ICKSEL or PANEL_DCDR bits.
SYSCNT_PANEL_CLK	PANEL_DCDR [5:0]	1	Clock Frequency Division Ratio Refer to Table 35.5 for details on the settings. Note: Settings other than those in Table 35.5 are prohibited.

chanhvinh.com

Table 35.5 I/O Clock Frequency and Divisors

		I/O Clock Frequency (MHz)		
DCDR[5:0]	**Clock Ratio**	**27.00**	**54.00**	**66.67**
000001	1/1	27.00	54.00	66.67
000010	1/2	13.50	27.00	33.33
000011	1/3	9.00	18.00	22.22
000100	1/4	6.75	13.50	16.67
000101	1/5	5.40	10.80	13.33
000110	1/6	4.50	9.00	11.11
000111	1/7	3.86	7.71	9.52
001000	1/8	3.38	6.75	8.33
001001	1/9	3.00	6.00	7.41
001100	1/12	2.25	4.50	5.56
010000	1/16	1.69	3.38	4.17
011000	1/24	1.13	2.25	2.78
100000	1/32	0.84	1.69	2.08

35.1.4 CLUT Table Read Select Signal Status Flag

The CLUT read select signal status can be read using the flags shown in Table 35.6.

Table 35.6 CLUT Table Read Select Signal Status Flags

Register Name	Bit Name	Initial Value	Description
SYSCNT_CLUT	GR0_CLT_SEL_ST	—	Graphics 0 CLUT Table Read Select Signal Status Flag 0: CLUT table 0 is read out. 1: CLUT table 1 is read out.
SYSCNT_CLUT	GR2_CLT_SEL_ST	—	Graphics 2 CLUT Table Read Select Signal Status Flag 0: CLUT table 0 is read out. 1: CLUT table 1 is read out.
SYSCNT_CLUT	GR3_CLT_SEL_ST	—	Graphics 3 CLUT Table Read Select Signal Status Flag 0: CLUT table 0 is read out. 1: CLUT table 1 is read out.

chanhvinh.com

35.2 Register Descriptions

The following register sets are allocated in the SH register map area.

- Symbols used in Register Descriptions

  Initial value: Register value after a reset
  —: Undefined value
  R/W: Readable/writable. The written value can be read.
  R/WC0: Readable/writable. Writing 0 initializes the bit. Writing 1 is ignored.
  R/WC1: Readable/writable. Writing 1 initializes the bit. Writing 0 is ignored.
  R: Read-only. The write value should always be 0.
  —/W: Write-only. The read value is undefined.

Table 35.7 Register Configuration of System Controller (CH0)

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt control register 1	SYSCNT_INT1	R/W	H'0000 0000	H'FCFF 7A80	32
Interrupt control register 2	SYSCNT_INT2	R/W	H'0000 0000	H'FCFF 7A84	32
Interrupt control register 4	SYSCNT_INT4	R/W	H'0000 0000	H'FCFF 7A8C	32
Interrupt control register 5	SYSCNT_INT5	R/W	H'0000 0000	H'FCFF 7A90	32
Panel clock control register	SYSCNT_PANEL_CLK	R/W	H'0001	H'FCFF 7A98	16
CLUT table read select signal status register	SYSCNT_CLUT	R	H'0000	H'FCFF 7A9A	16

chanhvinh.com

35.2.1 Interrupt Control Register 1 (SYSCNT_INT1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	INT_STA7	—	—	—	INT_STA6	—	—	—	INT_STA5	—	—	—	INT_STA4
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	INT_STA3	—	—	—	INT_STA2	—	—	—	INT_STA1	—	—	—	INT_STA0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28	INT_STA7	0	R/W	IV5_VBUFERR Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
27 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	INT_STA6	0	R/W	IV3_VBUFERR Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
23 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	INT_STA5	0	R/W	IV1_VBUFERR Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
19 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	INT_STA4	0	R/W	S0_VFIELD Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	INT_STA3	0	R/W	GR3_VLINE Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	INT_STA2	0	R/W	S0_VSYNCERR Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	INT_STA1	0	R/W	S0_LO_VSYNC Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	INT_STA0	0	R/W	S0_VI_VSYNC Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.

chanhvinh.com

35.2.2 Interrupt Control Register 2 (SYSCNT_INT2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	INT_STA9	—	—	—	INT_STA8
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	INT_STA9	0	R/W	S0_WLINE Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. (Setting prohibited in this product) 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	INT_STA8	0	R/W	IV6_VBUFERR Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.

chanhvinh.com

35.2.3 Interrupt Control Register 4 (SYSCNT_INT4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	INT_OUT7_ON	—	—	—	INT_OUT6_ON	—	—	—	INT_OUT5_ON	—	—	—	INT_OUT4_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	INT_OUT3_ON	—	—	—	INT_OUT2_ON	—	—	—	INT_OUT1_ON	—	—	—	INT_OUT0_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28	INT_OUT7_ON	0	R/W	IV5_VBUFERR Interrupt Output On/Off 0: Off 1: On
27 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	INT_OUT6_ON	0	R/W	IV3_VBUFERR Interrupt Output On/Off 0: Off 1: On
23 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	INT_OUT5_ON	0	R/W	IV1_VBUFERR Interrupt Output On/Off 0: Off 1: On
19 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	INT_OUT4_ON	0	R/W	S0_VFIELD Interrupt Output On/Off 0: Off 1: On
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	INT_OUT3_ON	0	R/W	GR3_VLINE Interrupt Output On/Off 0: Off 1: On
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	INT_OUT2_ON	0	R/W	S0_VSYNCERR Interrupt Output On/Off 0: Off 1: On
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	INT_OUT1_ON	0	R/W	S0_LO_VSYNC Interrupt Output On/Off 0: Off 1: On
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	INT_OUT0_ON	0	R/W	S0_VI_VSYNC Interrupt Output On/Off 0: Off 1: On

chanhvinh.com

35.2.4 Interrupt Control Register 5 (SYSCNT_INT5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	INT_OUT9_ON	—	—	—	INT_OUT8_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	INT_OUT9_ON	0	R/W	S0_WLINE Interrupt Output On/Off 0: Off 1: On (Setting prohibited in this product)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	INT_OUT8_ON	0	R/W	IV6_VBUFERR Interrupt Output On/Off 0: Off 1: On

chanhvinh.com

35.2.5 Panel Clock Control Register (SYSCNT_PANEL_CLK)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PANEL_ICKSEL [1:0]		—	—	—	PANEL_ ICKEN	—	—	PANEL_DCDR[5:0]					
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R/W	R/W	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	PANEL_ICKSEL [1:0]	All 0	R/W	Divided Clock Source Select 0: Video image clock (this is DV_CLK if INP_SEL = 1) 1: External clock (LCD0_EXTCLK) 2: Setting prohibited 3: Peripheral clock 1 (P1φ)
11 to 9	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PANEL_ ICKEN	0	R/W	Panel Clock Operation Enable 0: Disables operation of the panel clock operation block. 1: Enables operation of the panel clock operation block. Note: Set this bit to 0 before changing the value of the PANEL_ICKSEL or PANEL_DCDR bits.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5 to 0	PANEL_ DCDR[5:0]	1	R/W	Clock Frequency Division Ratio Refer to Table 35.5 for details on the settings. Note: Settings other than those in Table 35.5 are prohibited.

chanhvinh.com

35.2.6 CLUT Table Read Select Signal Status Register (SYSCNT_CLUT)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	GR3_CLT_SEL_ST	—	—	—	GR2_CLT_SEL_ST	—	—	—	—	—	—	—	GR0_CLT_SEL_ST
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	GR3_CLT_SEL_ST	—	R	Graphics 3 CLUT Table Read Select Signal Status Flag 0: CLUT table 0 is read out. 1: CLUT table 1 is read out.
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	GR2_CLT_SEL_ST	—	R	Graphics 2 CLUT Table Read Select Signal Status Flag 0: CLUT table 0 is read out. 1: CLUT table 1 is read out.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR0_CLT_SEL_ST	—	R	Graphics 0 CLUT Table Read Select Signal Status Flag 0: CLUT table 0 is read out. 1: CLUT table 1 is read out.

chanhvinh.com

36. Capture Engine Unit

The capture engine unit (CEU) is a capture module that fetches image data externally input and transfers it to the memory. The CEU is connected to the system bus via bus bridge modules.

36.1 Features of CEU

Lists the features of CEU as follows.

(1) Image data fetch

- Captures an image output from an external module and writes YCbCr data to the memory with it separated into Y data and CbCr data.
- Fetches image data other than YCbCr data, e.g. JPEG data, RGB565, from an externally connected module, such as a camera, and sequentially writes the image data to the memory.
- Fetches an interlace source image in both-field units or one-field units and writes it to the memory. In both-field capture, an image can be stored in the memory as a frame image.

(2) Filter processing

- Performs scale-down and removal of high-frequency components (only in the horizontal direction) for an image using internal filters. Note that the scaled-down image must not exceed VGA. The filter processing can be applied to only YCbCr input data.

(3) Format conversion

- Converts image data input in the YCbCr422 format into the YCbCr420 format and writes it to the memory. Note that the conversion algorithm is simple skipping in which the chrominance component (CbCr) of the even-numbered lines is skipped.

chanhvinh.com

36.2 Functional Overview of CEU

The functional overview of the CEU is shown in Table 36.1, and the main functions and their details are shown in Table 36.1.

Table 36.1 Functional Overview of CEU

Classification	Item	Function	Description	Note
Connectable camera	Size	5 megapixels	2,560 pixels × 1,920 lines	Horizontal: 4-pixel units Vertical: 4-line units The range of the image size that can be input is as follows. Horizontal: 2,560 pixels to 128 pixels Vertical: 1,920 lines to 96 lines [Note] This depends on the AC characteristics of the device to be connected, frame rate of the connected device, and transfer speed to the destination RAM.
		3 megapixels	2,048 pixels × 1,536 lines	
		2 megapixels	1,632 pixels × 1,224 lines	
		UXGA	1,600 pixels × 1,200 lines	
		SXGA (1)	1,280 pixels × 1,024 lines	
		SXGA (2)	1,280 pixels × 960 lines	
		WXGA	1,280 pixels × 768 lines	
		XGA	1,024 pixels × 768 lines	
		SVGA	800 pixels × 600 lines	
		WVGA	800 pixels × 480 lines	
		VGA	640 pixels × 480 lines	
		CIF	352 pixels × 288 lines	
		WQVGA	480 pixels × 240 lines	
		QVGA	320 pixels × 240 lines, 240 pixels × 320 lines	
		QCIF	176 pixels × 144 lines	
		QQVGA	160 pixels × 120 lines	
		Sub-QCIF	128 pixels × 96 lines	
	Input format	YCbCr422 8 bits	Cb_0, Y_0, Cr_0, $Y_{1...}$	Supports clock ratio of 1:1
			Cr_0, Y_0, Cb_0, $Y_{1...}$	
			Y_0, Cb_0, Y_1, $Cr_{0...}$	
			Y_0, Cr_0, Y_1, $Cb_{0...}$	
		Binary data	Specified amount to be fetched on edges of the sync signal	Written sequentially
			Data is fetched with the horizontal sync signal as an enable signal (RZ/A1LU and RZ/A1LC only).	
	Horizontal and vertical sync signal polarities	Arbitrary	High-active and low-active	
	Capture start location	Arbitrary	Can be specified in camera input clock units	Horizontal: 1-cycle units Vertical: 1-HD (horizontal sync signal) units
	Number of captured pixels	Arbitrary	Can be specified in 4-pixel units horizontally and in 4-line units vertically	
	Interlace	Both-field capture	Stored as a field image	Capture: 2-VD (vertical sync signal) units
			Stored as a frame image	
		One-field capture	Top field or bottom field can be specified	Capture: 1-VD units
Memory write	Output format	YCbCr422 YCbCr420	YCbCr420 is realized by simple skipping	
Filter function	No scaling or scale-down	Scale-down of captured display	Desired scaling factor from 1/16 to 1 (scaled-down display must not exceed VGA)	
	Low-pass filter		Removal of high-frequency components	Only in the horizontal direction

chanhvinh.com

Table 36.2 Main Functions of CEU and Their Details

Main Function	Detailed Description
Image data fetch	• Captures an image output from an external module and writes YCbCr data to the memory with it separated into Y data and CbCr data. • Fetches image data other than YCbCr data, e.g. JPEG data, from an externally connected module, such as a camera, and sequentially writes the image data to the memory. • Fetches an interlace source image in both-field units or one-field units and writes it to the memory. In both-field capture, an image can be stored in the memory as a frame image.
Filter processing	Performs scale-down and removal of high-frequency components (only in the horizontal direction) for an image using internal filters. Note that the scaled-down image must not exceed VGA. The filter processing can be applied to only YCbCr input data.
Format conversion	Converts image data input in the YCbCr422 format into the YCbCr420 format and writes it to the memory. Note that the conversion algorithm is simple skipping in which the chrominance component (CbCr) of the even-numbered lines is skipped.

Figure 36.1 shows a block diagram of the CEU.

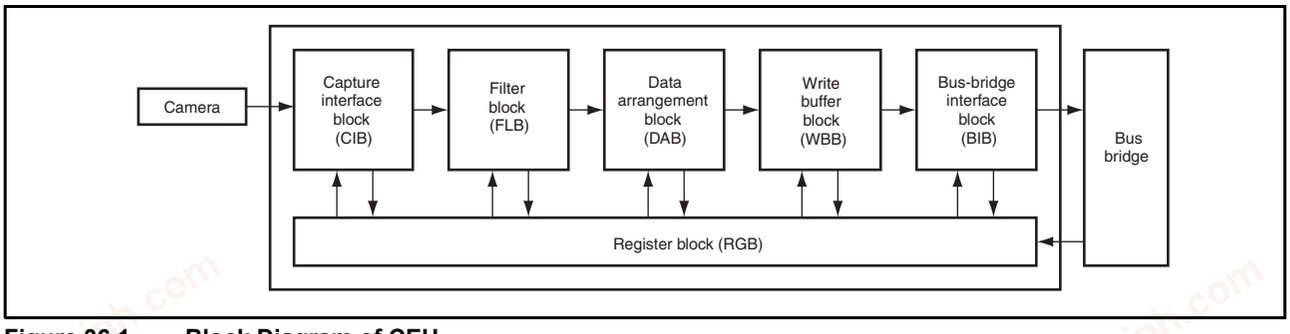


Figure 36.1 Block Diagram of CEU

36.3 Pin Configuration of CEU

The pin configuration of the CEU is shown in Table 36.3.

Table 36.3 Pin Configuration of CEU

Pin Name	Function	I/O	Description
VIO_D7 to VIO_D0*	CEU data bus	Input	Camera image data input to the CEU
VIO_CLK	CEU clock	Input	Camera clock input to the CEU
VIO_VD	CEU vertical sync	Input	Camera vertical sync signal input to the CEU
VIO_HD	CEU horizontal sync	Input	Camera horizontal sync signal input to the CEU
VIO_FLD	Field signal	Input	Field identification signal to the CEU

Note: * When the distinction according to the bus width for the data bus is not needed, VIO_D is used in this manual. Otherwise, VIO_D7 to VIO_D0 are used.

chanhvinh.com

36.4 Register Descriptions of CEU

The register configuration of the CEU is shown in Table 36.4.

Most CEU registers have a 2-plane configuration (plane A and plane B). The CEU switches the planes when using these 2-plane registers. A mirror address, which is an address that can always access the register on the unused plane, is provided for each 2-plane register. Figure 36.2 shows the timing to switch the register planes. The CEU switches the register planes at the same time a VD interrupt is asserted.

In the following register descriptions, "during operation" indicates the period that begins when the CEU is activated by the CE bit in the capture start register (CAPSR) and ends when a capture end interrupt (CPE) of the capture event flag clear register (CETCR) occurs. In the read-only bits in each register, the write value should always be 0. If a value other than 0 is written to any of these bits, correct operation cannot be guaranteed.

Table 36.4 Register Configuration of CEU

			Addresses			
Register Name	**Abbr.**	**R/W**	**Address (Plane A)**	**Address (Plane B)**	**Mirror Address**	**Access Size**
CEU Capture start register	CAPSR	R/W	H'E821 0000	—	—	32
CEU Capture control register	CAPCR	R/W	H'E821 0004	—	—	32
CEU Capture interface control register*	CAMCR	R/W	H'E821 0008	—	—	32
CEU Capture interface cycle register*	CMCYR	R/W	H'E821 000C	—	—	32
CEU Capture interface offset register	CAMOR	R/W	H'E821 0010	H'E821 1010	H'E821 2010	32
CEU Capture interface width register	CAPWR	R/W	H'E821 0014	H'E821 1014	H'E821 2014	32
CEU Capture interface input format register	CAIFR	R/W	H'E821 0018	—	—	32
CEU register control register	CRCNTR	R/W	H'E821 0028	—	—	32
CEU register forcible control register	CRCMPR	R/W	H'E821 002C	—	—	32
CEU Capture filter control register	CFLCR	R/W	H'E821 0030	H'E821 1030	H'E821 2030	32
CEU Capture filter size clip register	CFSZR	R/W	H'E821 0034	H'E821 1034	H'E821 2034	32
CEU Capture destination width register	CDWDR	R/W	H'E821 0038	H'E821 1038	H'E821 2038	32
CEU Capture data address Y register	CDAYR	R/W	H'E821 003C	H'E821 103C	H'E821 203C	32
CEU Capture data address C register	CDACR	R/W	H'E821 0040	H'E821 1040	H'E821 2040	32
CEU Capture data bottom-field address Y register	CDBYR	R/W	H'E821 0044	H'E821 1044	H'E821 2044	32
CEU Capture data bottom-field address C register	CDBCR	R/W	H'E821 0048	H'E821 1048	H'E821 2048	32
CEU Capture bundle destination size register	CBDSR	R/W	H'E821 004C	H'E821 104C	H'E821 204C	32
CEU Firewall operation control register	CFWCR	R/W	H'E821 005C	—	—	32
CEU Capture low-pass filter control register	CLFCR	R/W	H'E821 0060	H'E821 1060	H'E821 2060	32
CEU Capture data output control register	CDOCR	R/W	H'E821 0064	H'E821 1064	H'E821 2064	32
CEU Capture event interrupt enable register	CEIER	R/W	H'E821 0070	—	—	32
CEU Capture event flag clear register	CETCR	R/W	H'E821 0074	—	—	32
CEU Capture status register	CSTSR	R	H'E821 007C	—	—	32
CEU Capture data size register	CDSSR	R/W	H'E821 0084	—	—	32
CEU Capture data address Y register 2	CDAYR2	R/W	H'E821 0090	H'E821 1090	H'E821 2090	32
CEU Capture data address C register 2	CDACR2	R/W	H'E821 0094	H'E821 1094	H'E821 2094	32
CEU Capture data bottom-field address Y register 2	CDBYR2	R/W	H'E821 0098	H'E821 1098	H'E821 2098	32
CEU Capture data bottom-field address C register 2	CDBCR2	R/W	H'E821 009C	H'E821 109C	H'E821 209C	32

chanhvinh.com

Note: * After changing the setting of a register (CAMCR or CMCYR) that is determined by the external module characteristics, do not start capture until at least 10 external input clock cycles have elapsed.

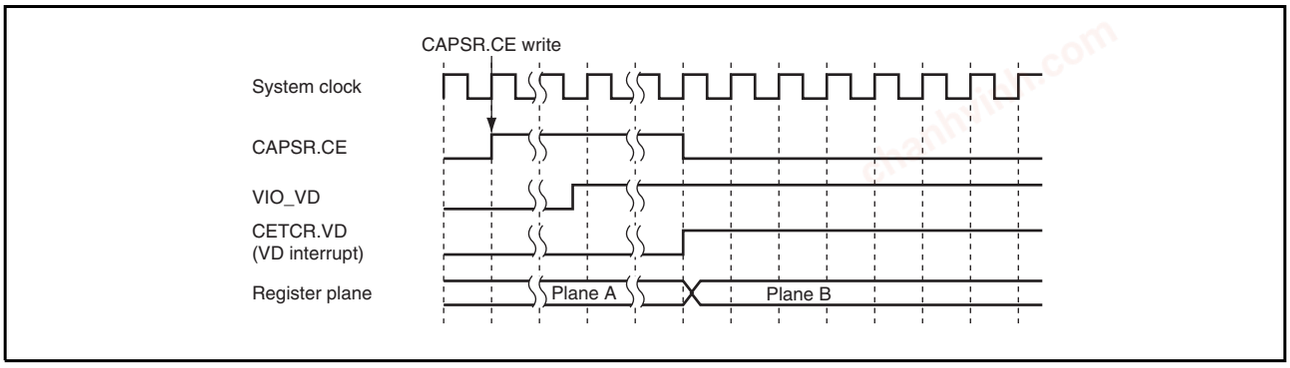


Figure 36.2 Register Plane Switching Timing (VD Polarity is High-Active in Data Enable Fetch Mode)

36.4.1 Capture Start Register (CAPSR)

CAPSR captures data input to the CEU from an external module.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPKIL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	CPKIL	0	R/W	Write 1 to this bit to perform a software reset of capturing. At a software reset, capturing ends immediately without completing capture operation until the end of a frame. Clear the CE bit to 0 when writing 1 to this bit. Processing of the capture software reset is indicated by this bit being set to 1. When this bit is 1, do not start capturing since reset processing is in progress. When restarting capture operations, after referring to the CPTON bit in CSTSR to ensure that the CEU is halted (in the idle state), wait until this bit is cleared to 0. The timing of restarting capture operations is shown in Figure 36.6. When a software reset is generated by this bit, a capture end interrupt (CPE bit in CETCR) may be output immediately after the software reset. However, such kind of interrupt should be ignored. Also, even if the capture end interrupt is not output, the interrupt source (CPE bit) must be cleared before capturing of the next frame. 0: Normal state 1: Software reset of capturing
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
0	CE	0	R/W	• Single-capture This bit reserves capturing of the next frame. When 1 is written to this bit, the capture of one frame starts from the next VD input, and stops when the one-frame capture end interrupt (CPE bit in CETCR) is asserted (Figure 36.7). To perform capture again, write 1 to this bit. After the VD or HD polarity is changed, do not write 1 to this bit until the next VD interrupt is asserted. As this bit indicates the capture reserve state, this bit is read as 1 after it is set to 1 and until VD is input. When VD is input, this bit returns to 0 and so is read as 0. The capture end is determined by the one-frame capture end interrupt (CPE bit). This is similar in data fetch mode. Registers should be set before the VD interrupt of the frame where capture starts next. The new register settings take effect at the next VD input. When registers are modified during capturing, the register settings take effect from the capture operations of the next VD input. If a setting register to which writing during capturing is prohibited is modified during capturing, an interrupt source (IGRW bit in CETCR) is generated. For details on the interrupt source, see the description on CETCR. • Continuous capture When this bit is set to 1 while the CTNCP bit in CAPCR is set to 1, continuous capture starts from the next frame (Figure 36.8). Note that this bit is not cleared to 0 but remains as 1. To stop capturing, clear this bit to 0; capturing stops after the current frame is completed. Continuous capture operations are possible in only image capture mode. The start address of the memory to which the captured data is written to must be set for each frame. 0: Stops capturing 1: Starts capturing

When both the VD (vertical sync signal) and HD (horizontal sync signal) polarities are high-active, one frame is defined as a period from a VD rising edge to the next VD rising edge, and one line as a period from an HD rising edge to the next HD rising edge. Figure 36.3 shows the timing of one frame (when both the VD and HD polarities are high-active).

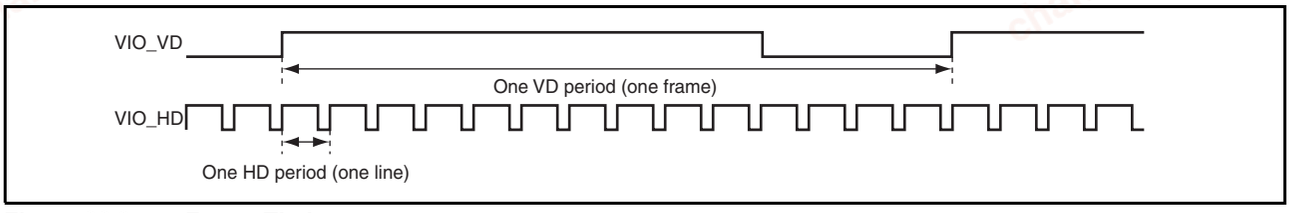


Figure 36.3 Frame Timing

When both the VD and HD polarities are high-active, similar to one frame, one field is defined as follows:

- Period from a VD rising edge to the next VD rising edge
- One line is a period from an HD rising edge to the next HD rising edge

The field identification signal FLD should be fixed for at least 1-HD period from a VD input. Figure 36.4 shows the timing of one field (when both the VD and HD polarities are high-active).

chanhvinh.com

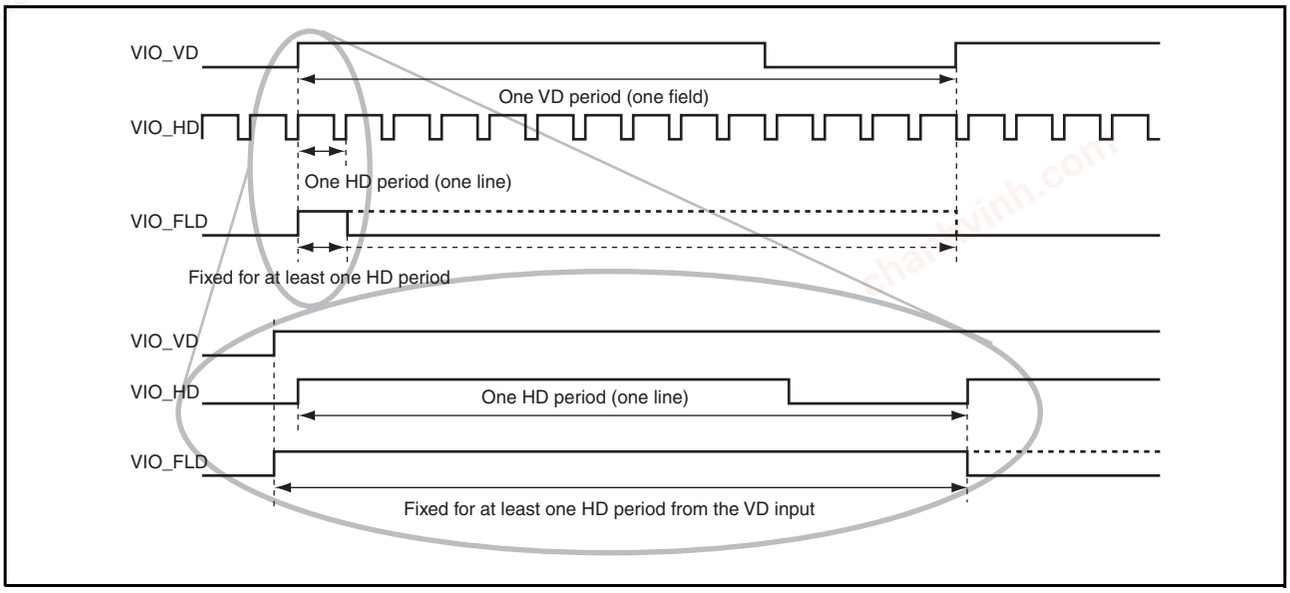


Figure 36.4 One Field Timing

In data enable fetch (RZ/A1LU and RZ/A1LC only), one frame is defined as a period from a VD rising edge to the VD falling edge. With the HD as an enable signal (positive polarity), data of a cycle in which the HD is asserted is fetched while the VD is high. Figure 36.5 shows the timing of one frame for data enable fetch.

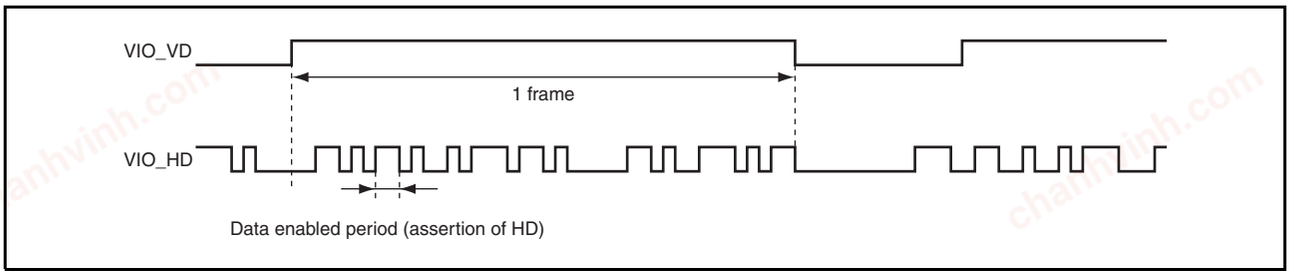


Figure 36.5 Frame Timing (Data Enable Fetch)

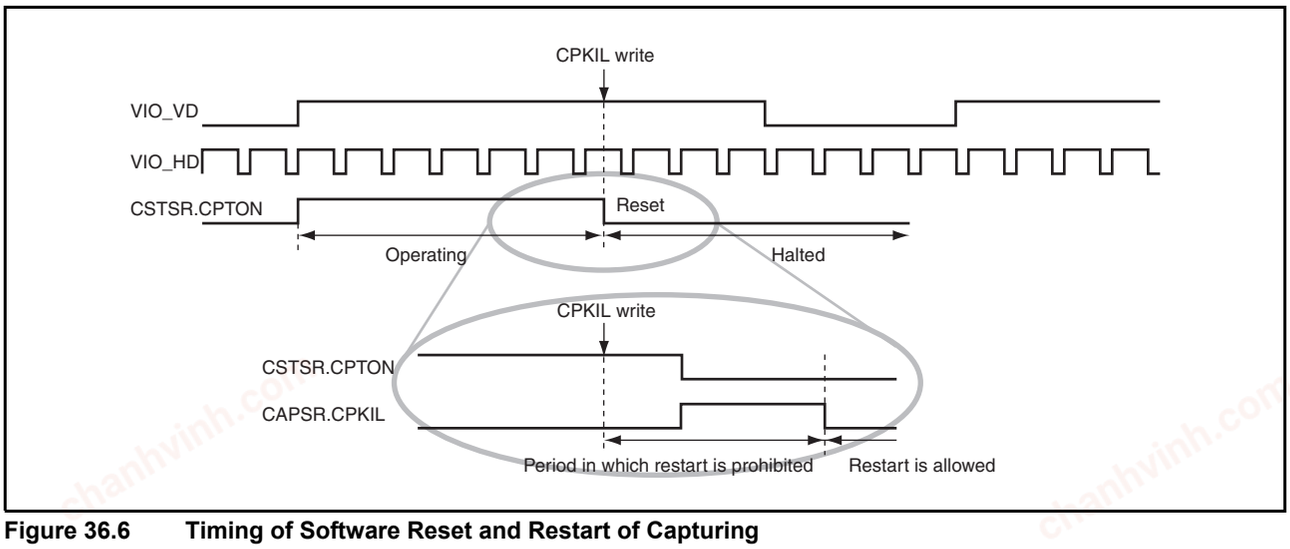


Figure 36.6 Timing of Software Reset and Restart of Capturing

chanhvinh.com

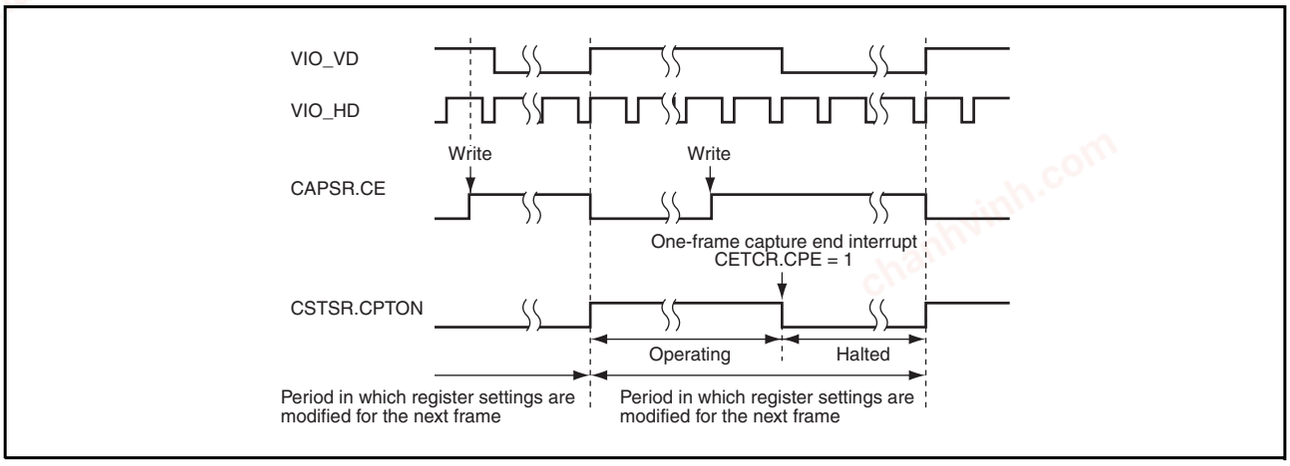


Figure 36.7 Timing of Modifying CE Bit and Register Setting in One Frame Capture

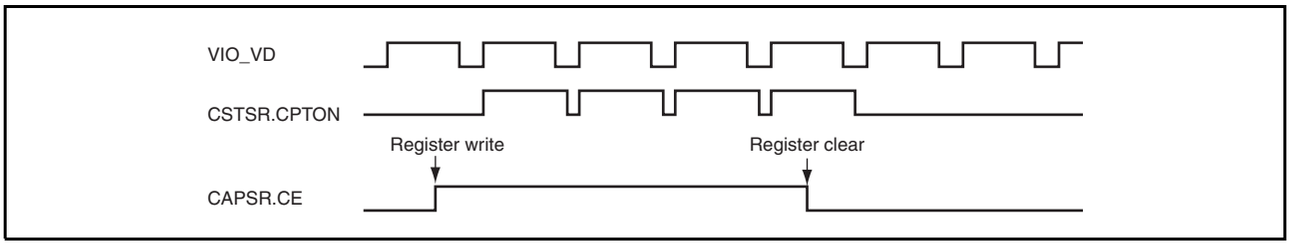


Figure 36.8 Continuous-Frame Capture

chanhvinh.com

36.4.2 Capture Control Register (CAPCR)

CAPCR sets continuous-frame capture and the frame drop intervals.

Do not modify this register during operation. If this register is modified during operation, correct operation cannot be guaranteed. In addition, the IGRW bit (interrupt source) in CETCR is set to B'1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FDRP[7:0]								—	—	MTCM[1:0]		—	—	—	CTNCP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	FDRP[7:0]	H'00	R/W	These bits set the frame drop interval in continuous-frame capture. When these bits are cleared to 0, frame drop is not performed, and all frames are captured. Figure 36.9 shows the value set in these bits and the timing of captured frames. The frame drop interval unit differs according to the capture setting. Table 36.5 shows the relationship between the capture setting and frame drop interval unit. The image of the frame drop timing for each capture setting when these bits are set to 2 is shown in Figure 36.10. In both-field capture, capturing is performed continuously for 2-VD periods, regardless of whether the second field is the top field or bottom field. In addition, in both-field capture, the frame drop counter is incremented when the first field has been identified as the top field or bottom field, regardless of whether the second field is the top field or bottom field. When 0 is written to the CE bit in CAPSR, capturing terminates after the current frame has been captured for a capture frame. However, for a drop frame, capturing is forcibly terminated in the CEU so no capture end interrupt (CPE bit in CETCR) is output. While CE bit is 1, do not change the setting of these bits. Note: Do not change the setting of these bits during continuous capture operations. To change the setting of these bits, stop continuous capture (CE bit = 0), clear the CTNCP bit (continuous capture) in CAPCR to 0, and then restart continuous capture. Continuous capture is performed during the period of CAPSR.CE = 1 shown in Figure 36.9.
23, 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
21, 20	MTCM[1:0]	00	R/W	These bits specify the unit for transferring data to a bus bridge module. The efficiency of writing image data can be improved by continuously accessing the addresses. To improve the write efficiency, set these bits to 11. The setting of these bits appear to be unchanged from the outside. 00: Transferred to the bus in 32-byte units 01: Transferred to the bus in 64-byte units 10: Transferred to the bus in 128-byte units 11: Transferred to the bus in 256-byte units (1) Image capture 00: Y data and C data are transferred in 32-byte units 01: Y data and C data are transferred in 64-byte units 10: Y data and C data are transferred in 128-byte units 11: Y data and C data are transferred in 256-byte units (2) Data fetch 00: Data is transferred in 32-byte units 01: Data is transferred in 64-byte units 10: Data is transferred in 128-byte units 11: Data is transferred in 256-byte units
19 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	CTNCP	0	R/W	When capturing is started with this bit set to 1, capturing continues until the CE bit in CAPSR is cleared to 0 or a software reset is initiated by the CPKIL bit in CAPSR (see Figure 36.8). Continuous capture must be set before capturing is started. This bit is modified only after 0 is written to the CE bit to stop capturing. If this bit is modified during capturing, correct operation cannot be guaranteed. In data fetch mode, clear this bit to 0. 0: One-frame capture when the CE bit is 1 1: Continuous capture until the CE bit is cleared to 0
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

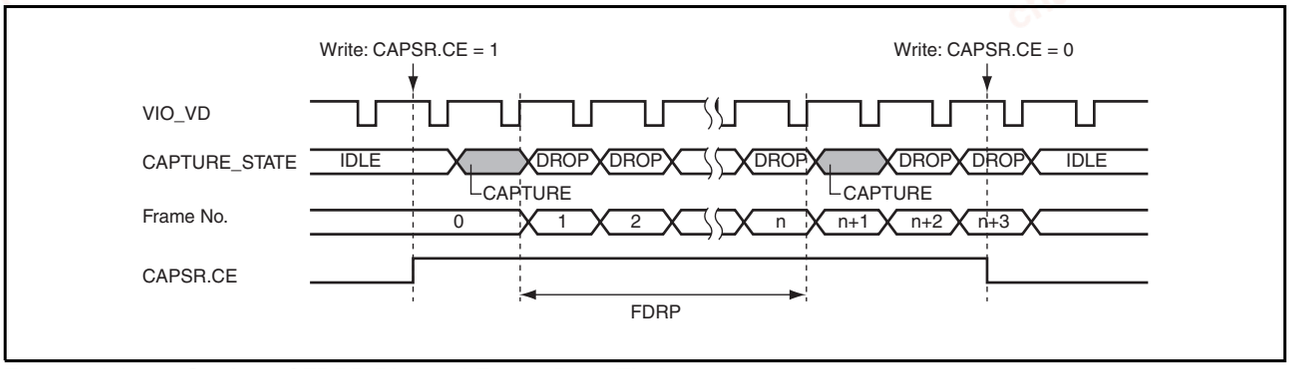


Figure 36.9 Setting of FDRP Bits and Frame Drop Timing

chanhvinh.com

Table 36.5 Relationship between Capture Setting and Frame Drop Interval Unit

Input Mode	Captured Image	First Captured Image	Frame Drop Interval Unit	Capture Setting
Progressive	Frame	Frame immediately after capture start	Frame	A
Interlace	Both-field (2-VD capture)	Field immediately after capture start	2 fields (first capture field count)	B
				C
		Top field	2 fields (top-field count)	D
		Bottom field	2 fields (bottom-field count)	E
	One-field (1-VD capture)	Field immediately after capture start	First capture field	F
				G
		Top field	Top field	H
		Bottom field	Bottom field	I

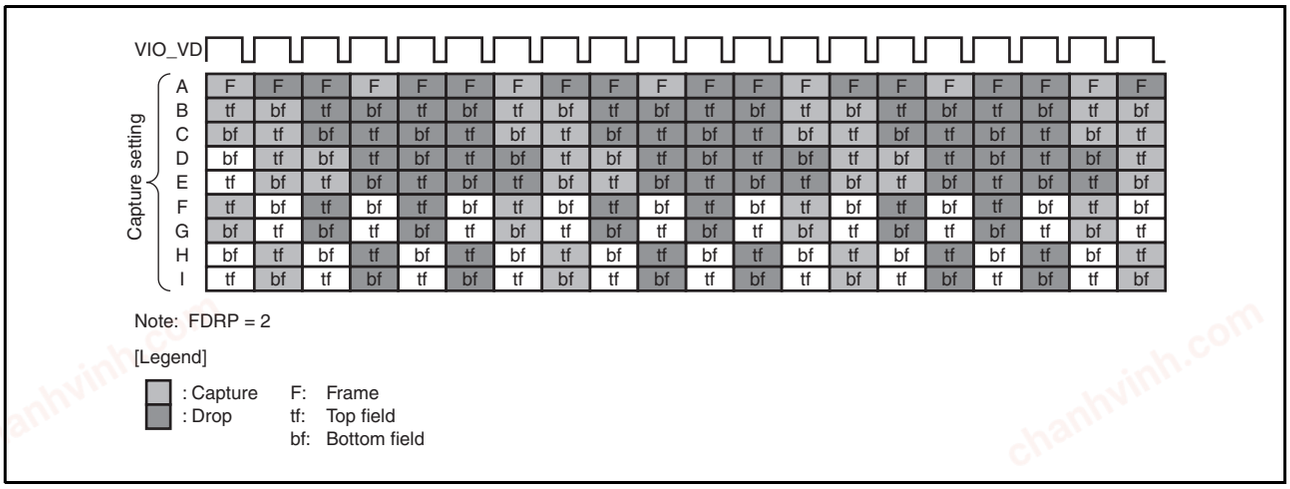


Figure 36.10 Image of Frame Drop Timing for Capture Settings (FDRP Bits = 2)

chanhvinh.com

36.4.3 Capture Interface Control Register (CAMCR)

CAMCR sets the capture interface.

The following items are set by CAMCR.

- Selection between the rising edge or falling edge of the camera clock for signal capture or data fetch operation (RZ/A1LU and RZ/A1LC only)
- Selection between image capture operation or data fetch operation
- Polarities of the vertical and horizontal sync signals
- Input order of image data components (Y, Cb, and Cr) (only for image capture mode)
- Selection of digital image input pins (8 bits only)
- Polarity of the field identification signal

CAMCR must be set according to the module connected. In data fetch mode, set the DTARY bits to B'0. Do not modify this register during operation. If this register is modified during operation, correct operation cannot be guaranteed. In addition, the IGRW bit (interrupt source) in CETCR is set to B'1.

Note: After changing the setting of this register, do not start capture until at least 10 external input clock cycles have elapsed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	VDSEL*	HDSEL*	FLDSEL*	DSEL*	—	—	—	—	—	—	—	FLDPOL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	DTIF	—	—	DTARY[1:0]		—	—	JPG[1:0]		—	—	VDPOL	HDPOL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27	VDSEL*	0	R/W	Sets the edge for capturing the vertical sync signal (VD) from an external module. 0: VD is captured at the rising edge of the camera clock. 1: VD is captured at the falling edge of the camera clock.
26	HDSEL*	0	R/W	Sets the edge for capturing the horizontal sync signal (HD) from an external module. 0: HD is captured at the rising edge of the camera clock. 1: HD is captured at the falling edge of the camera clock.
25	FLDSEL*	0	R/W	Sets the edge for capturing the field identification signal (FLD) from an external module. 0: FLD is captured at the rising edge of the camera clock. 1: FLD is captured at the falling edge of the camera clock.
24	DSEL*	0	R/W	Sets the edge for fetching the image data (D7 to D0) from an external module. 0: D7 to D0 are fetched at the rising edge of the camera clock. 1: D7 to D0 are fetched at the falling edge of the camera clock.
23 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
16	FLDPOL	0	R/W	Sets the polarity of the field identification signal (FLD) from an external module. 0: When the FLD signal is high-active, the field is detected as the top field and when low-active, the field is detected as the bottom field. 1: When the FLD signal is low-active, the field is detected as the top field and when high-active, the field is detected as the bottom field.
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	DTIF	0	R/W	Sets the digital image input pins from which data is to be captured. The write value should always be 0.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	DTARY[1:0]	00	R/W	These bits set the input order of the luminance component and chrominance component. The order in which the luminance component (Y) and chrominance component (Cb and Cr) are input from an external module differs among modules. The CEU supports the input orders shown in Figure 36.12. Set the corresponding value in these bits. In data fetch mode, set these bits to 00. 00: Image input data is fetched in the order of Cb_0, Y_0, Cr_0, and Y_1 01: Image input data is fetched in the order of Cr_0, Y_0, Cb_0, and Y_1 10: Image input data is fetched in the order of Y_0, Cb_0, Y_1, and Cr_0 11: Image input data is fetched in the order of Y_0, Cr_0, Y_1, and Cb_0
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	JPG[1:0]	00	R/W	These bits select the fetched data type. 00: Image capture mode (input data are separated into Y data and CbCr data for output to the memory) 01: Data synchronous fetch mode (specified size of input data are output to the specified memory addresses in order of input and in synchronization with the sync signal) 10: Data enable fetch mode (RZ/A1LU and RZ/A1LC only) (input data are fetched with HD as an enable signal and output to the specified addresses in memory in order of input)
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	VDPOL	0	R/W	Sets the polarity for detection of the vertical sync signal input from an external module. Figure 36.14 and Figure 36.15 show the relationship between the VIO_VD and VIO_HD signals and VD interrupt when high-active is selected. Since a VD interrupt may occur when this bit value is modified, the VD bit in CETCR must always be cleared to 0 when this bit value is changed. In data enable fetch mode, this bit is not used and the sense for detection is always active high. 0: Vertical sync signal (VD) from an external module is detected as high-active 1: Vertical sync signal (VD) from an external module is detected as low-active
0	HDPOL	0	R/W	Sets the polarity for detection of the horizontal sync signal input from an external module. Figure 36.16 shows the relationship between the HD and HD interrupt when high-active is selected. Since an HD interrupt may occur when this bit value is modified, the HD bit in CETCR must always be cleared to 0 when this bit value is changed. 0: Horizontal sync signal (HD) from an external module is detected as high-active 1: Horizontal sync signal (HD) from an external module is detected as low-active

Note: * This bit is only present in the RZ/A1LU and RZ/A1LC. This is reserved in the RZ/A1L.

chanhvinh.com

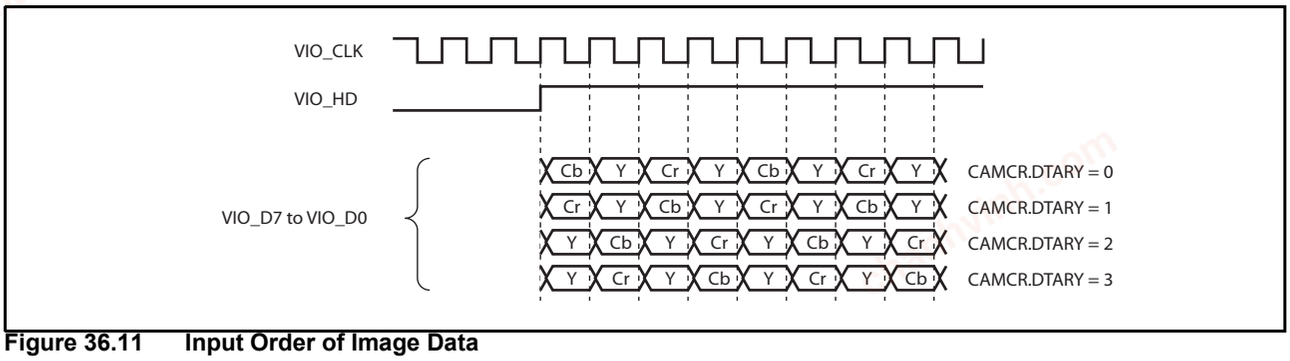


Figure 36.11 Input Order of Image Data

The JPG bit in CAMCR selects whether digital image data is fetched or data such as JPEG is fetched. In addition, when data such as JPEG is fetched, select whether the specified amount of data is continuously fetched in synchronization with the sync signal or data is fetched while the horizontal sync signal is enabled (RZ/A1LU and RZ/A1LC only).

In data enable fetch mode, one frame is defined as a period from the rising edge to the falling edge of the vertical sync signal (VD) for data fetching. The horizontal sync signal (HD) is enabled only when the VD is high and treated as an enable signal. Data input in the cycle in which the HD is asserted (high) is fetched and output to the memory continuously.

This module starts fetching data at the rising edge of the VD and stops fetching data at the falling edge of the VD in data enable fetch mode. Thus, if the VD remains high and does not go low, end processing does not start. In addition, if the VD remains high and the HD also remains asserted, data continues to be fetched.

Figure 36.12 and Figure 36.13 show the interface timing in data enable fetch mode.

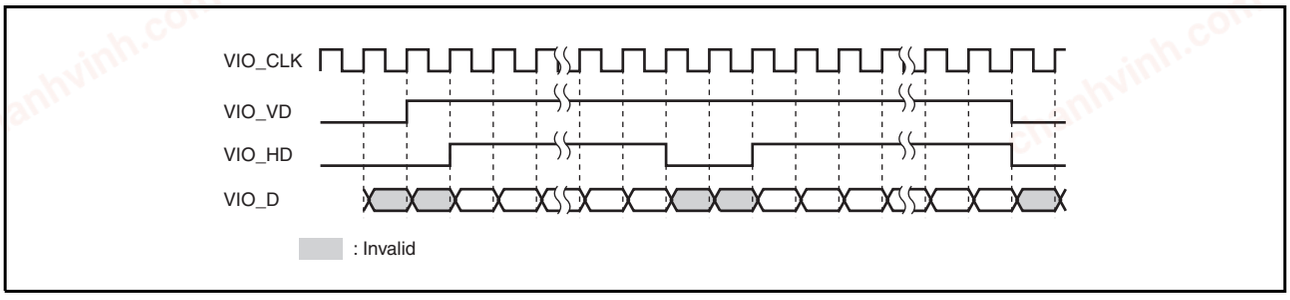


Figure 36.12 Data Enable Fetch Timing (HD Asserted (High) While VD is High)



Figure 36.13 Data Enable Fetch Timing (HD Asserted (High) When VD is Not High)

In data enable fetch mode, this module generates a VD interrupt in response to detection of the active level of VD. In image capture mode and data synchronous fetch mode, this module generates a VD interrupt in response to the first detection of the active level of HD following detection of the active level of VD. Note that, when VD and HD are asserted and detected at the same time, this module generates a VD interrupt at that time.

Figures 36.14 to 36.16 show the relationships between the VIO_VD signal and the VD interrupt, the VIO_VD and VIO_HD signals and the VD interrupt, and the VIO_HD signal and the HD interrupt.

chanhvinh.com

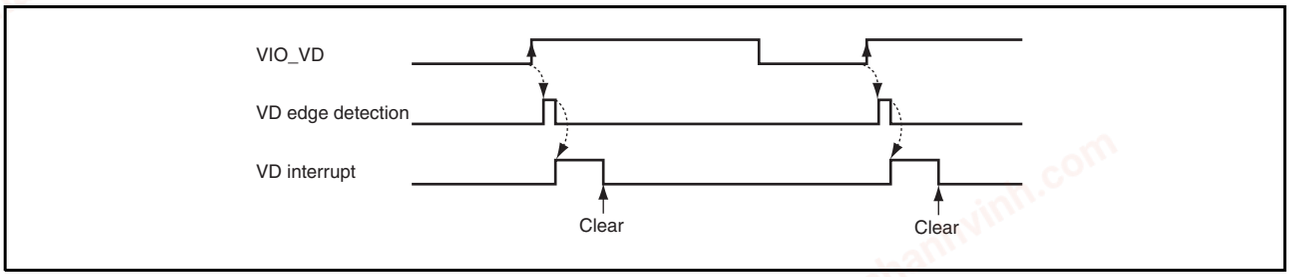


Figure 36.14 Relationship between VIO_VD and VD Interrupt when VD is High-Active (In Data Enable Fetch Mode)

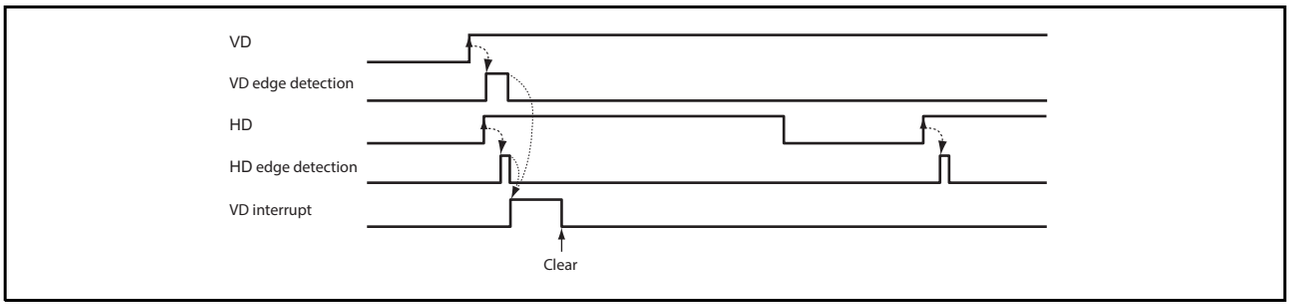


Figure 36.15 Relationship between the VIO_VD and VIO_HD signals and the VD Interrupt when VD and HD are High-Active (in Image Capture Mode or Data Synchronous Fetch Mode)

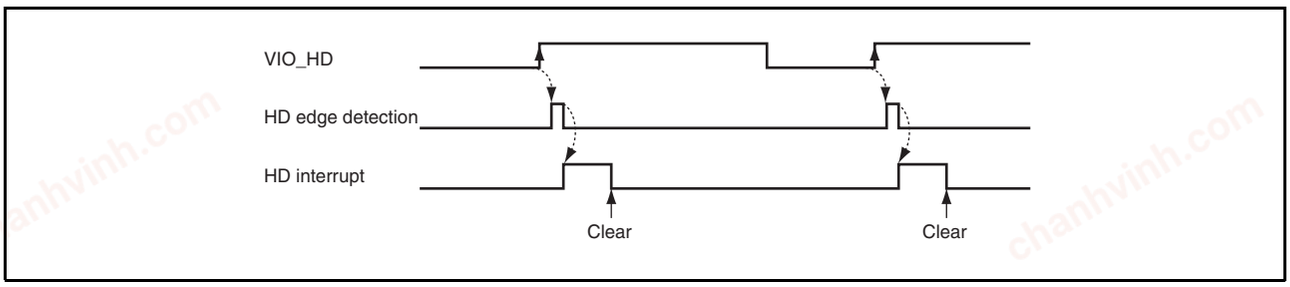


Figure 36.16 Relationship between VIO_HD and HD Interrupt when HD is High-Active

chanhvinh.com

36.4.4 Capture Interface Cycle Register (CMCYR)

CMCYR is used to detect an illegal VD and an illegal HD. For HD, the number of cycles from a rising edge of HD to the next rising edge is set (falling edges when low-active is selected for HD). For VD, the number of HD inputs from a rising edge of VD to the next rising edge is set (falling edges when low-active is selected for VD).

Do not modify this register during operation. If this register is modified during operation, correct operation cannot be guaranteed. In addition, the IGRW bit (interrupt source) in CETCR is set to B'1.

Set 0 in all bits of this register, during data enable fetch mode.

Note: After changing the setting of this register, do not start capture until at least 10 external input clock cycles have elapsed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	VCYL[13:0]													
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	HCYL[13:0]													
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
29 to 16	VCYL[13:0]	H'0000	R/W	Vertical HD Count of External Module These bits set the number of VD cycles of an external module with the number of HD inputs. The interrupt source bit IGVS in CETCR is set to 1 when the actual number of VD cycles input from the external module differs from this setting. Set these bits for detecting an illegal VD. When these bits are all cleared to 0, the interrupt source bit IGVS in CETCR is not set to 1. Though the interrupt source bit IGVS in CETCR may be set to 1 after the VDPOL bit (VD polarity) in CAMCR is changed, this interrupt should be ignored.
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13 to 0	HCYL[13:0]	H'0000	R/W	Horizontal Cycle Count of External Module These bits set the number of HD cycles of an external module. The interrupt source bit IGHS in CETCR is set to 1 when the actual number of HD cycles input from the external module differs from this setting. Set these bits for detecting an illegal HD. When these bits are all cleared to 0, the interrupt source bit IGHS in CETCR is not set to 1. Though the interrupt source bit IGHS in CETCR may be set to 1 after the HDPOL bit (HD polarity) in CAMCR is changed, this interrupt should be ignored.

chanhvinh.com

36.4.5 Capture Interface Offset Register (CAMOR)

CAMOR sets the location to start capturing when capturing images.

Since the number of HD (horizontal sync signal) inputs from a VD (vertical sync signal) input to the start of a valid image period, and the number of clock cycles from an HD input to the start of a valid image period differ among external modules, these must be set in CAMOR. By setting a value greater than the valid image area, part of the image can be clipped for capture. When fetching data, the setting of this register becomes the number of cycles (HD count) up to the start of a valid data period.

This register is not used, during data enable fetch mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	VOFST[11:0]											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	HOFST[12:0]												
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27 to 16	VOFST[11:0]	H'000	R/W	These bits specify the capture start location in terms of the HD count from a vertical sync signal (1-HD units). The blanking period from a vertical sync signal differs among external modules. Therefore, the vertical capture start location must be specified by these bits in terms of the HD count from a vertical sync signal so that an image can be captured from the valid image area (see Figure 36.17). Some external modules output a vertical sync signal as a data enable signal. In this case, there is no blanking period so these bits must be cleared to 0 (see Figure 36.18).
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12 to 0	HOFST[12:0]	H'0000	R/W	These bits specify the capture start location in terms of the number of clock cycles from a horizontal sync signal (1-cycle units). The blanking period from a horizontal sync signal differs among external modules. Therefore, the horizontal capture start location must be specified by these bits in terms of external input clock cycles from a horizontal sync signal so that an image can be captured from the valid image area. This is similar in data synchronous fetch mode (see Figure 36.19). Some external modules output a horizontal sync signal as a data enable signal. In this case, there is no blanking period so these bits must be cleared to 0 (see Figure 36.20). Note: The first HD (horizontal sync signal) being input simultaneously or after the first VD (vertical sync signal) is the operating condition of the CEU. These inputs are affected by the polarities (set by the VDPOL and HDPOL bits in CAMCR).

chanhvinh.com

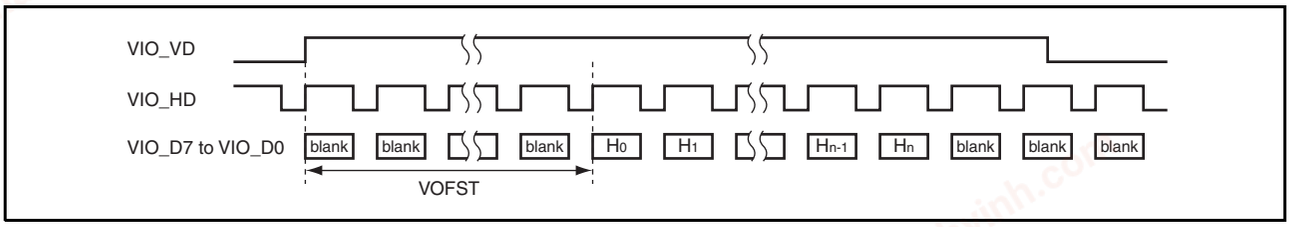


Figure 36.17 Vertical Offset

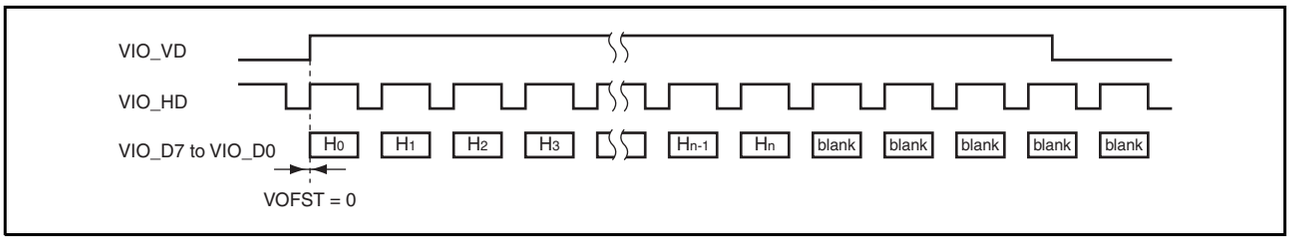


Figure 36.18 Timing when VD is Data Enable Signal

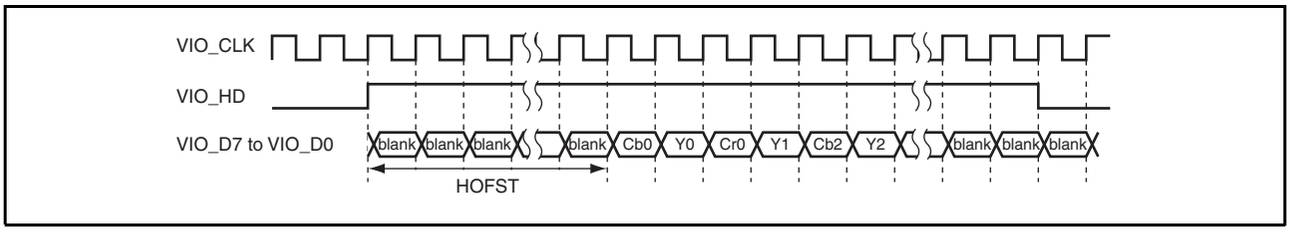


Figure 36.19 Horizontal Offset

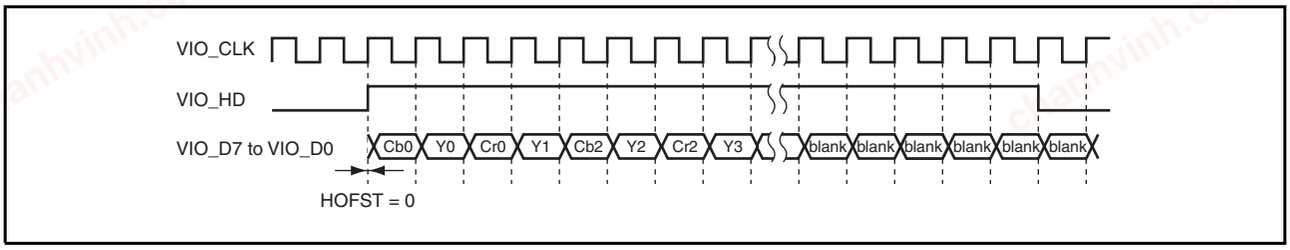


Figure 36.20 Timing when HD is Data Enable Signal (8-bit Interface)

chanhvinh.com

36.4.6 Capture Interface Width Register (CAPWR)

CAPWR sets the fetch (capture) cycle width when capturing images.

The cycle width unit differs according to the interface and the data type to be captured. For each setting unit, see Table 36.6.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	VWDTH[11:0]											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	HWDTH[12:0]												
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27 to 18 17, 16	VWDTH[11:2] VWDTH[1:0]	H'000	R/W R	These bits specify the vertical capture period (4-HD units). These bits specify the number of lines (HD count) to be captured from the location specified by the VOFST bits in CAMOR. Figure 36.21 shows the timing when the vertical blanking period is 0. The CEU captures only the number of lines (HD count) specified by these bits in the vertical direction. Make a setting in the same way to obtain data synchronization. The maximum value to be set is 1,920 HD (5 megapixels).
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12 to 1 0	HWDTH[12:1] HWDTH[0]	H'0000	R/W R	These bits specify the horizontal capture period. These bits specify the number of cycles to be captured from the location specified by the HOFST bits. Figure 36.22 shows the timing when the horizontal blanking period is 0. The CEU captures for only the number of cycles specified by these bits in the horizontal direction. Make a similar setting for data synchronous fetch. The maximum value to be set is as follows: Image capture (8-cycle units) : 5,120 cycles (2,560 pixels) Data synchronous fetch (4-cycle units) : 2,560 cycles (2,560 bytes) Note: In data synchronous fetch mode, set CFSZR and CDWDR according to the values set in this register. For details, see the descriptions on CFSZR and CDWDR.

chanhvinh.com

Table 36.6 Unit for Setting Fetch (Capture) Cycle Width

	Vertical Direction		Horizontal Direction	
Interface	Image Capture	Data Synchronous Fetch	Image Capture	Data Synchronous Fetch
8-bit interface	4 HD	4 HD	8 cycles	4 cycles

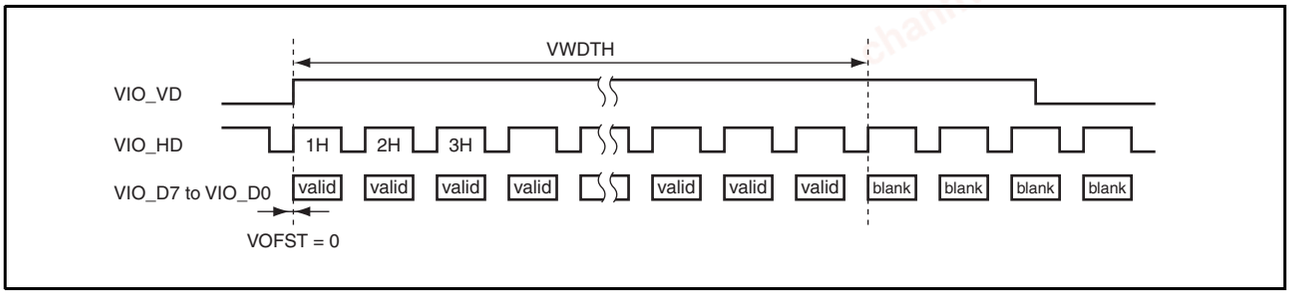


Figure 36.21 Vertical Capture Timing

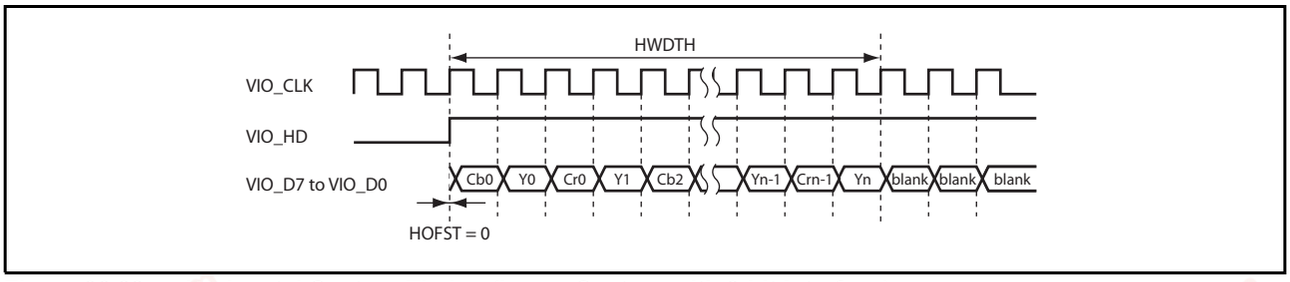


Figure 36.22 Horizontal Capture Timing (Image Capture with 8-bit Interface)

chanhvinh.com

36.4.7 Capture Interface Input Format Register (CAIFR)

CAIFR sets the input mode (progressive or interlace) for capturing images, the images to be captured (frame, both-field, or one-field), and the image from which capturing starts (top field, bottom field, etc.). CAIFR is not used in data fetch mode.

Do not modify this register during operation. If this register is modified during operation, correct operation cannot be guaranteed. In addition, the IGRW bit (interrupt source) in CETCR is set to B'1.

The items set by CAIFR are listed in Table 36.7.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	IFS	—	—	—	CIM	—	—	FCI[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	IFS	0	R/W	Sets the input mode for capturing images. 0: Progressive 1: Interlace
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	CIM	0	R/W	Sets the images to be captured. Clear this bit to 0 when the input mode for image capture is progressive (frame image) or when the input mode for image capture is interlace for continuous capture of both the top and bottom fields. Set this bit to 1 when the input mode for image capture is interlace for capture of only a one-field image. 0: Capture of frame image (1 VD) or both-field image (2 VD) 1: Capture of one-field image (1 VD)
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	FCI[1:0]	00	R/W	These bits set the timing to start capturing. The timing to start capturing is set by specifying the image to be captured first. Set these bits to 00 when the input mode is progressive. 00: Capture starts from the VD input immediately after the CEU activation regardless of it being a top or bottom field 01: After the CEU activation, input of a top-field image is waited, and then capture starts from the top field 10: After the CEU activation, input of a bottom-field image is waited, and then capture starts from the bottom field 11: Setting prohibited

chanhvinh.com

Table 36.7 CAIFR Setting Items

Input Mode	IFS Bit	Captured Image	CIM Bit	Image to Start Capture	FCI Bits
Progressive	0	Frame	0	Frame immediately after activation	00
Interlace	1	Both-field (2-VD capture)	0	Field immediately after activation	00
				Top field	01
				Bottom field	10
				Setting prohibited	11
		One-field (1-VD capture)	1	Field immediately after activation	00
				Top field	01
				Bottom field	10
				Setting prohibited	11

In frame image capture and one-field image capture, a one-frame capture end interrupt occurs when capture for 1 VD finishes. In both-field image capture, a one-field capture end interrupt occurs when capture for 1 VD finishes and a one-frame capture end interrupt occurs when capture for 2 VD finishes. At this time, a one-field capture end interrupt occurs simultaneously with a one-frame capture end interrupt. Figure 36.24 shows the timing of a one-frame capture end interrupt and one-field capture end interrupt in both-field image capture.

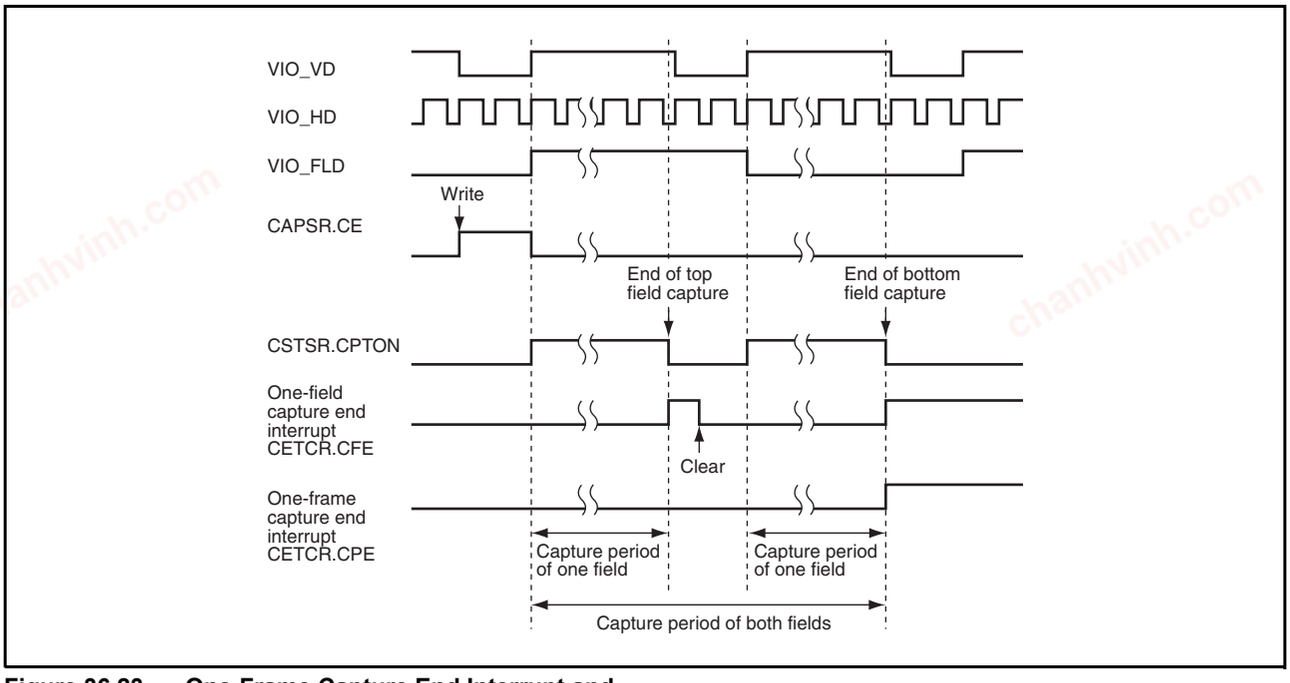


Figure 36.23 One-Frame Capture End Interrupt and One-Field Capture End Interrupt in Both-Field Image Capture

A captured frame image or captured one-field image is stored in the memory from the addresses set in CDAYR and CDACR (Figure 36.24). Captured both-field images are stored in different memory areas depending on whether it is a top-field or bottom-field image. A top-field image is stored in the memory from the addresses set in CDAYR and CDACR whereas a bottom-field image is stored in the memory from the addresses set in CDBYR and CDBCR (Figure 36.25).

chanhvinh.com

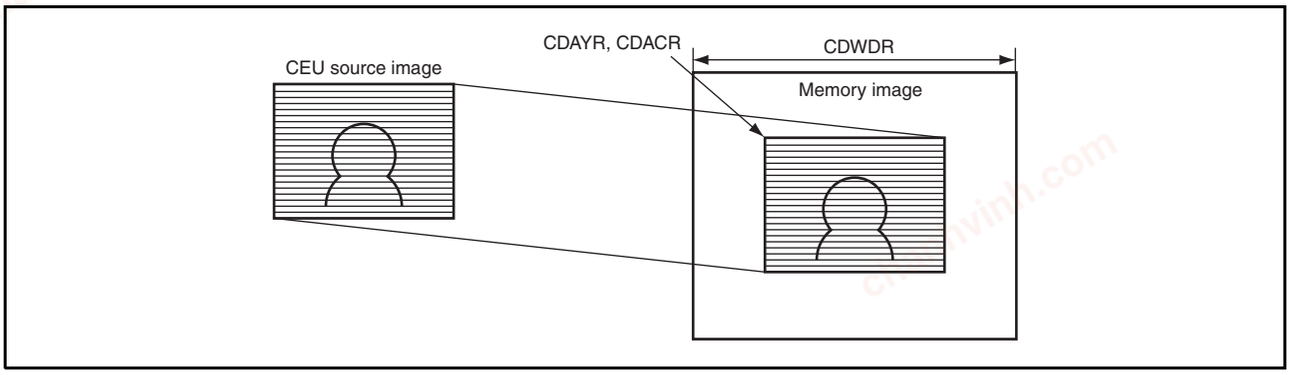


Figure 36.24 Image of Storing Captured Frame Image or Captured One-Field Image in Memory

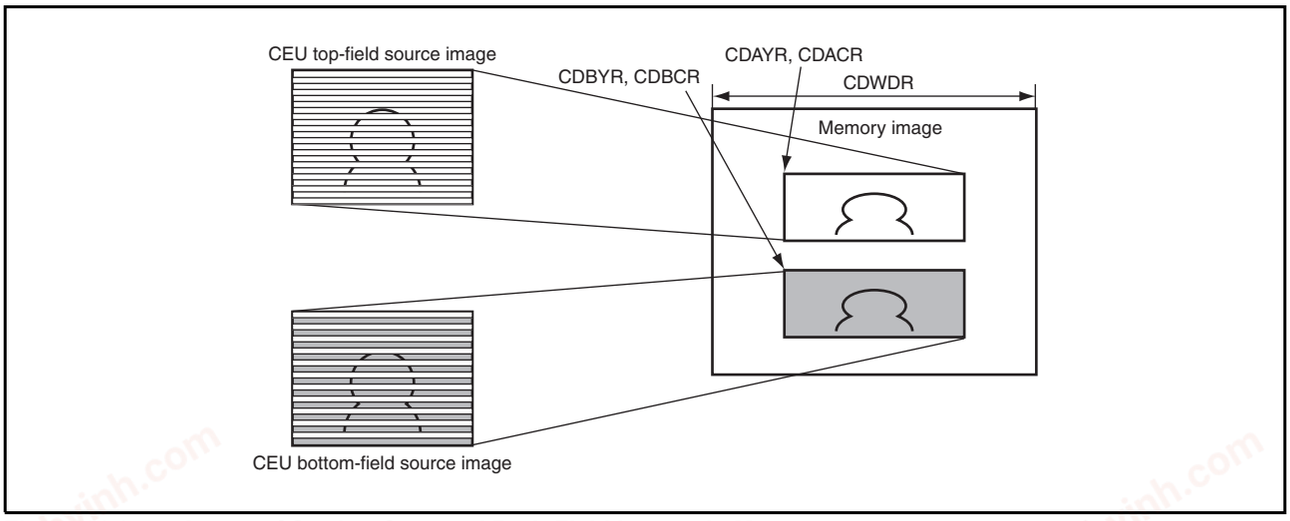


Figure 36.25 Image of Storing Captured Both-Field Images in Memory

If the FCI bits are set to B'00 for continuous capture in interlace input mode, images are continuously captured for 2 VD with the first captured field as the reference in both-field image capture (Figure 36.26). In one-field image capture, only the first captured field is continuously captured for 1 VD (Figure 36.27).

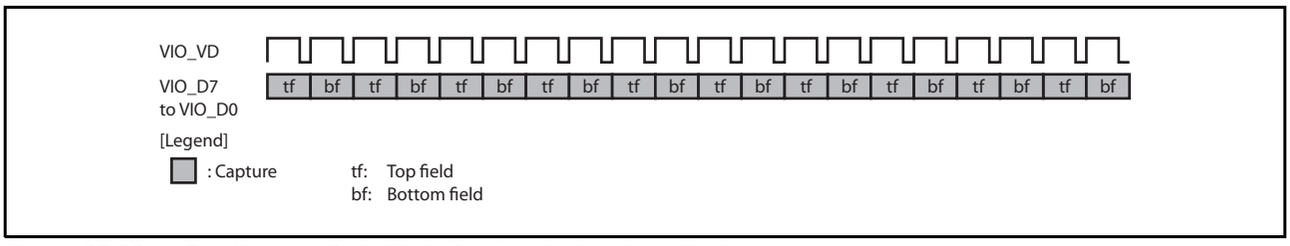


Figure 36.26 Continuous Both-Field Capture in Interlace Mode (Image Immediately after Activation is Top Field (FCI Bits = B'00))

chanhvinh.com

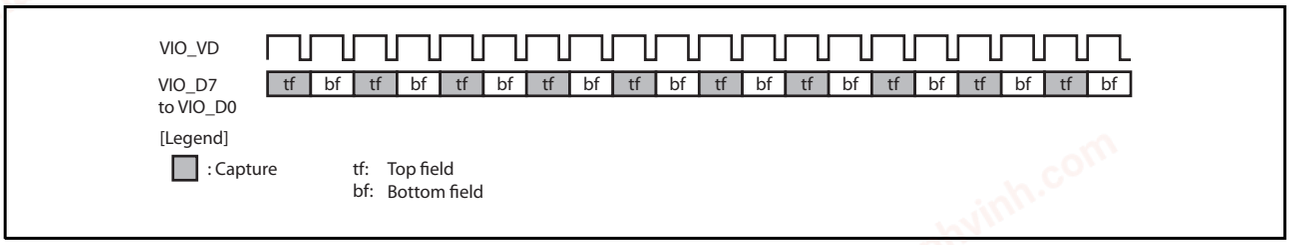


Figure 36.27 Continuous One-Field Capture in Interlace Mode (Image Immediately after Activation is Top Field (FCI Bits = B'00))

(1) Storage of Interlace Input as Frame Image

The CEU can store an interlace source image in the memory as a frame image. To store an interlace source image as a frame image, make the following register settings:

Input mode: Interlace (IFS bit = B'1)
Capture image: Both-field (CIM bit = B'0)
Image to start capture: Any setting other than the prohibited setting (FCI bits = as desired)

Figure 36.28 shows a memory image of capturing both fields of an interlace input and storing it as a frame image in the memory. Set the start addresses of the memory destination for the captured top-field image in CDAYR and CDACR, and the start addresses of the memory destination for the captured bottom-field image in CDBYR and CDBCR. When storing an interlace image as a frame image in the memory, set the horizontal image size of the memory area in CDWDR with the top-field image and bottom-field image placed next to each other as shown in Figure 36.28. In addition, set the number of captured lines of the field image in the VWDTH bits in CAPWR.

A memory image of folding the horizontal image size of the memory area in Figure 36.28 at CDWDR/2 is shown in Figure 36.29. Setting the registers to form the image in Figure 36.28 enables an interlace image to be stored as a frame image in the memory as shown in Figure 36.29.

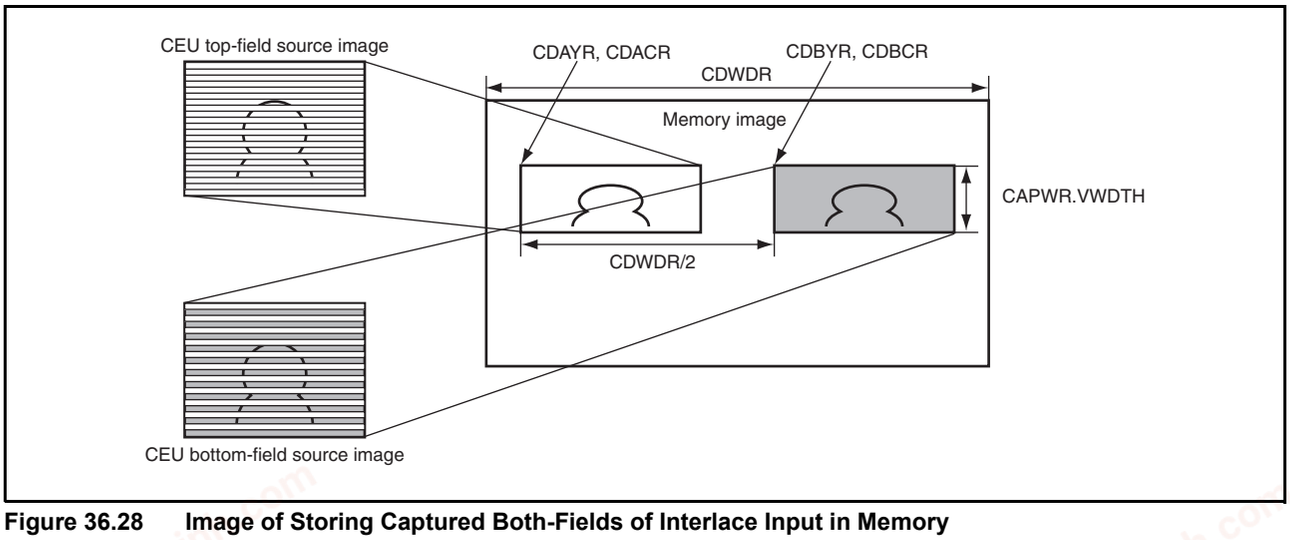


Figure 36.28 Image of Storing Captured Both-Fields of Interlace Input in Memory

chanhvinh.com

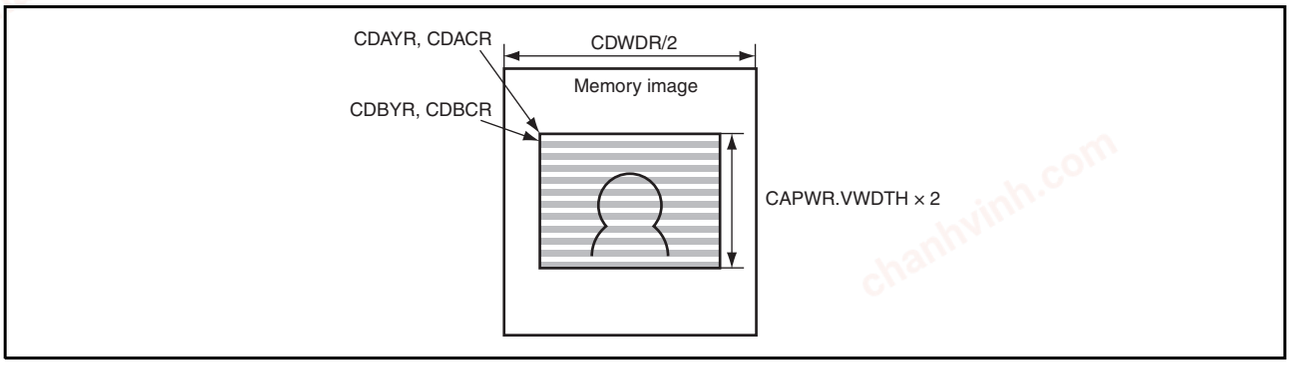


Figure 36.29 Image of Storing Interlace Input as Frame Image in Memory

36.4.8 CEU Register Control Register (CRCNTR)

CRCNTR controls switching of the planes of registers with a 2-plane configuration.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	RVS	—	—	RS	RC
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	RVS	0	R/W	Sets the timing to switch the register plane in both-field capture. The setting of this bit is valid only when the RC bit is 1 in both-field capture. 0: Switches the register plane every 2 VD 1: Switches the register plane every 1 VD
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	RS	0	R/W	Specifies which register plane is used by the CEU in synchronization with VD. The setting of this bit is valid only when the RC bit is 0. 0: Uses plane A of the register 1: Uses plane B of the register
0	RC	0	R/W	Specifies switching of the register plane used by the CEU in synchronization with VD. If the register plane is not switched, the register plane specified by the RS bit is used. 0: Uses the specified register plane in synchronization with VD 1: Switches the register plane in synchronization with VD

chanhvinh.com

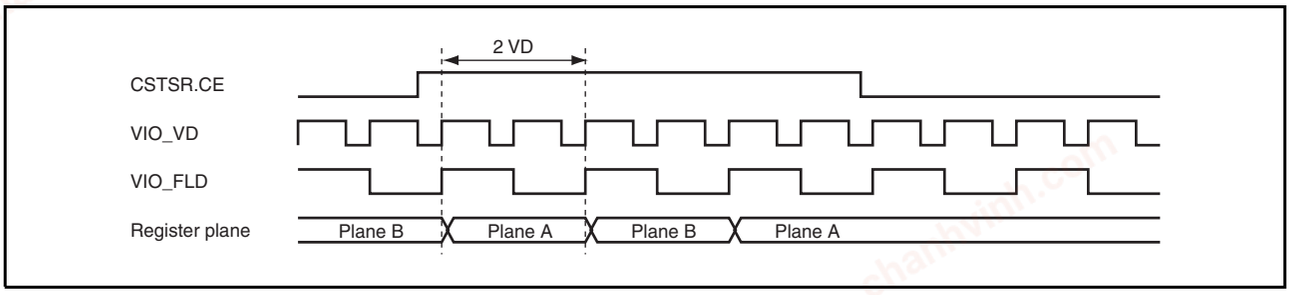


Figure 36.30 Timing for Register Plane Switching when RVS Bit is B'0

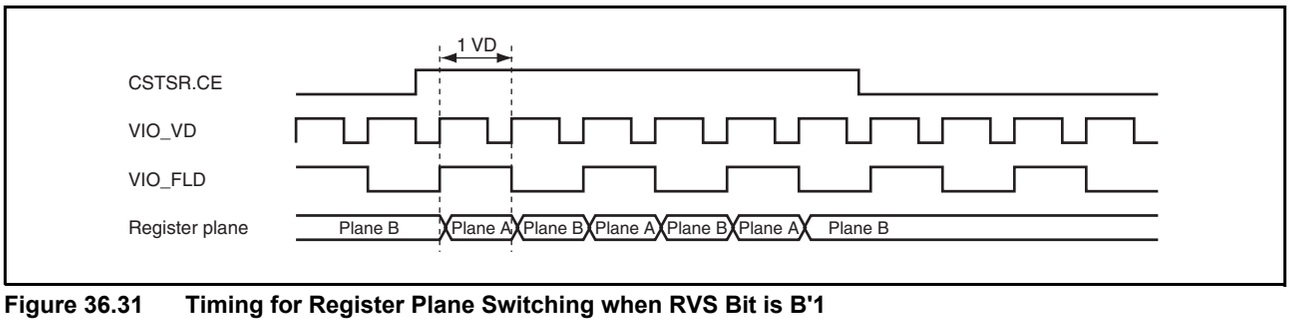


Figure 36.31 Timing for Register Plane Switching when RVS Bit is B'1

36.4.9 CEU Register Forcible Control Register (CRCMPR)

CRCMPR forcibly controls switching of the planes of registers with a 2-plane configuration. Setting this register enables direct control of register plane switching.

Do not modify this register during operation. If this register is modified during operation, correct operation cannot be guaranteed. In addition, the IGRW bit (interrupt source) in CETCR is set to B'1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RA
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	RA	0	R/W	Indicates the register plane currently specified. This register value automatically changes in synchronization with VD for starting capture. To start capture with plane A of the register when a setting to switch the register plane in synchronization with VD has been made (RC bit in CRCNTR is 1), specify plane B of the register using this bit. 0: Specifies plane A of the register 1: Specifies plane B of the register

chanhvinh.com

36.4.10 Capture Filter Control Register (CFLCR)

CFLCR sets the scale-down factor for the filter to scale images down.

The CEU has an image scale-down filter which can be used to scale down the captured images before storing them in the memory. Set CFLCR to 0 when not performing scale-down (same size output). If a value other than 0 is set in CFLCR, scale-down is performed. In data fetch mode, set CFLCR to 0.

When handling an interlace source image as a frame image, set CFLCR to 0 not to use the filter.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VMANT[3:0]				VFRAC[11:0]											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HMANT[3:0]				HFRAC[11:0]											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	VMANT[3:0]	H'0	R/W	Mantissa Part of Vertical Scale-Down Factor The specifiable range is H'0 to H'F. When H'0 is set for the VMANT bits and H'000 is set for the VFRAC bits, the scale-down filter is not used.
27 to 19 18 to 16	VFRAC[11:3] VFRAC[2:0]	H'000	R/W R	Fraction Part of Vertical Scale-Down Factor The specifiable range is H'000 to H'FF8. The fraction of the scale-down factor that cannot be set with only the VMANT bits must be set with these bits.
15 to 12	HMANT[3:0]	H'0	R/W	Mantissa Part of Horizontal Scale-Down Factor The specifiable range is H'0 to H'F. When H'0 is set for the HMANT bits and H'000 is set for the HFRAC bits, the scale-down filter is not used.
11 to 3 2 to 0	HFRAC[11:3] HFRAC[2:0]	H'000	R/W R	Fraction Part of Horizontal Scale-Down Factor The specifiable range is H'000 to H'FF8. The fraction of the scale-down factor that cannot be set with only the HMANT bits must be specified with these bits.

An image scale-down filter is installed in the CEU, and the captured images can be scaled down and stored in the memory.

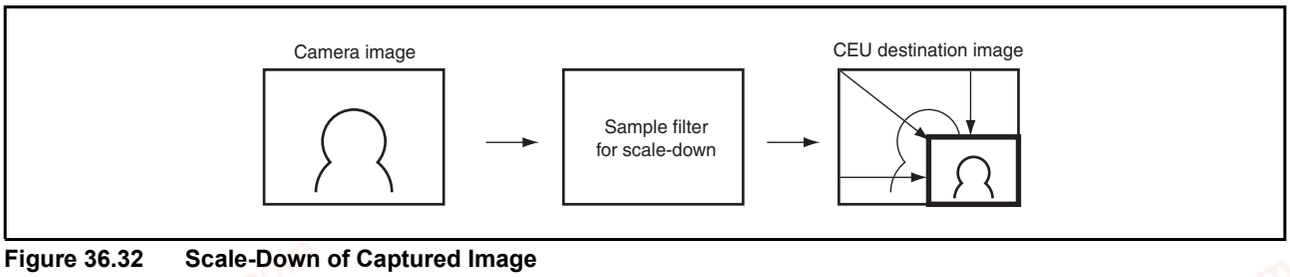


Figure 36.32 Scale-Down of Captured Image

The formulas for obtaining the MANT (VMANT or HMANT) and FRAC (VFRAC or HFRAC) values from the input pixel count and output pixel count of the filter are shown below. Set the MANT and FRAC bits in order to obtain the desired output pixel count from the number of pixels input to the CEU.

First, calculate preliminary MANT and FRAC values. The parameters needed for calculation are defined as follows:

chanhvinh.com

$$\alpha = MANT \times 4096 + FRAC \quad \text{. . . Formula 1}$$

$$SCL \text{ (scaling factor)} = \frac{4096}{\alpha} \quad \text{. . . Formula 2}$$

Assuming an operator $\lfloor x \rfloor$ which discards fractions of an integer x, the MANT and FRAC values can be temporarily set as follows, according to formula 1 and formula 2.

$$MANT = \left\lfloor \frac{1}{SCL} \right\rfloor, \quad FRAC = \left\lfloor 512 \times \left(\frac{1}{SCL} - MANT \right) \right\rfloor \times 8$$

Here, the scaled-down filter output size ($SIZE_D$) can be calculated using the input image size S_{in} (half of the CAPWR setting) in the following formula.

$$SIZE_D = \left\lfloor 1 + \left(\left\lfloor \frac{1}{2} + \frac{S_{in} - 1}{MANT_{pre}} \right\rfloor - 1 \right) \times \frac{MANT_{pre} \times 4096}{\alpha} \right\rfloor \quad \text{. . . Formula 3}$$

$$\begin{bmatrix} MANT_{pre} = 1 \longrightarrow (0 \le MANT < 2) \\ MANT_{pre} = 2 \longrightarrow (2 \le MANT < 4) \\ MANT_{pre} = 4 \longrightarrow (4 \le MANT < 8) \\ MANT_{pre} = 8 \longrightarrow (8 \le MANT\ ) \end{bmatrix}$$

The number of output pixels can be obtained by substituting the temporarily calculated MANT, FRAC, and input image size into these formulas. If the calculated number of output pixels is smaller than the number of output pixels used to obtain the preliminary MANT and FRAC values, recalculate with a smaller FRAC (α) value, and set the MANT and FRAC values in this register so that a pixel value greater than the desired number of output pixels can be obtained.

Example: Scale down 640 pixels to 480 pixels

SCL = 480/640 = 3/4, and the preliminary settings of MANT = 1, $MANT_{pre}$ = 1, and FRAC = H'550 are made. Substituting these in the following formula results in an output pixel count of 479.

$$SIZE_D = \left\lfloor 1 + \left(\left\lfloor \frac{1}{2} + \frac{S_{in} - 1}{MANT_{pre}} \right\rfloor - 1 \right) \times \frac{MANT_{pre} \times 4096}{\alpha} \right\rfloor \quad \text{. . . Formula 3}$$

Since this output pixel count is smaller than the desired output pixel count of 480, the formula is recalculated with a FRAC value of H'548, a value eight less than the previous time. The obtained result of output pixel count = 480 is equal to the desired output pixel count of 480, so this register is set as MANT = 1 and FRAC = H'548.

chanhvinh.com

Table 36.8 Setting Examples for Each Scale-Down Filter Factor

Scale-Down Factor	FRAC		MANT	Input Pixel Count	Output Pixel Count	Clipping Size (CFSZR)
	Decimal	Hexadecimal				
7/8	576	H'240	1	640	560	560
3/4	1352	H'548	1	640	480	480
5/8	2448	H'990	1	640	400	400
1/2	0	H'0	2	640	320	320
3/8	2728	H'AA8	2	640	240	240
1/3	0.0	H'0	3	640	213	212
1/4	0.0	H'0	4	640	160	160
1/5	0.0	H'0	5	640	128	128
1/6	0.0	H'0	6	640	107	104
1/7	0.0	H'0	7	640	91	88
1/8	0.0	H'0	8	640	80	80
1/16	4088	H'FF8	15	640	40	40

Note: This scale-down filter uses a VGA-size line memory for scale-down. Therefore, when an image larger than the VGA size is input for scale-down, settings must be made so that the output image size is equal to or larger than the SubQCIF size and equal to or smaller than the VGA size. When an image is not scaled down (same size output), this restriction does not apply.

36.4.11 Capture Filter Size Clip Register (CFSZR)

CFSZR sets the clipping size for fine adjustment of the image size output from the filter, and must be set in combination with CFLCR. When clipping the output size of the filter, set the clipping size as a number of pixels, and the setting unit should be four pixels. CFSZR must be set even when scale-down is not performed (same size output).

In data synchronous fetch mode, set CFSZR according to the setting of CAPWR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	VFCLP[11:0]											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	HFCLP[11:0]											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27 to 18 17, 16	VFCLP[11:2] VFCLP[1:0]	H'000	R/W R	These bits set the vertical clipping value of the filter output size (4-pixel units).
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 2 1, 0	HFCLP[11:2] HFCLP[1:0]	H'000	R/W R	These bits specify the horizontal clipping value of the filter output size (4-pixel units).

chanhvinh.com

The scale-down filter in the CEU may output an odd number of pixels or lines depending on the settings. To adjust the output size of the filter, the CEU clips the destination image by using the number of pixels specified in CFSZR, as shown in Figure 36.33. The clipping size must be specified vertically and horizontally in 4-pixel units.

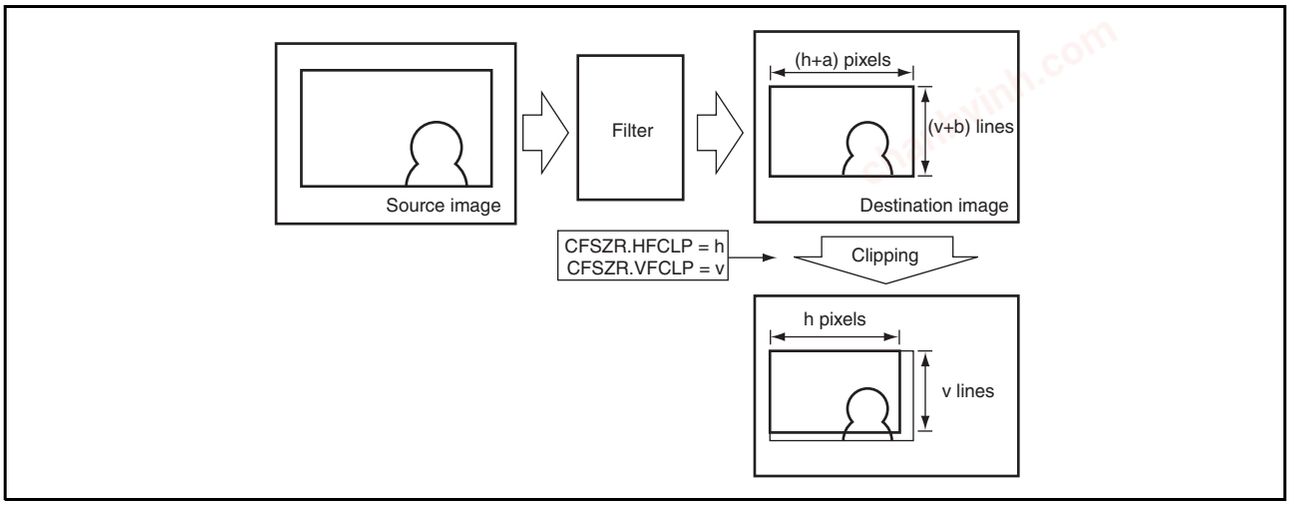


Figure 36.33 Clipping of Image Output from Filter

The pixels to be clipped are counted from the top-left corner of a display. The pixels located to the right of the specified number of pixels or below the specified number of lines are discarded by the clipping function. If the number of pixels specified in CFSZR is larger than that output from the filter, correct operation cannot be guaranteed. To avoid this, the clipping size specified in CFSZR must be equal to or smaller than the number of pixels output from the filter.

Note: In data synchronous fetch mode, the following settings are required. Data cannot be fetched correctly unless the following settings are made.
VFCLP = CAPWR.VWDTH
HFCLP = CAPWR.HWDTH/2

chanhvinh.com

36.4.12 Capture Destination Width Register (CDWDR)

CDWDR sets the horizontal image size in the memory area where the captured image is to be output in 4-byte units (4-pixel units).

In data synchronous fetch mode, set CDWDR according to the setting of CAPWR.

This register is not used, during data enable fetch mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	CHDW[12:0]												
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12 to 2 1, 0	CHDW[12:2] CHDW[1:0]	H'0000	R/W R	These bits specify the horizontal image size in the memory area where the captured image is to be stored (4-byte units). The image data captured by the CEU is stored in the memory. If the right end of the captured image does not match the horizontal image size in the memory area as shown in Figure 36.34, some addresses must be skipped at the right end of the image when storing the captured image. Therefore, the horizontal image size in the memory area where the captured image is to be stored must be set in these bits. The maximum value to be set is 8188 bytes (8188 pixels). In data synchronous fetch mode, set as follows: CHDW = CAPWR.HWDTH

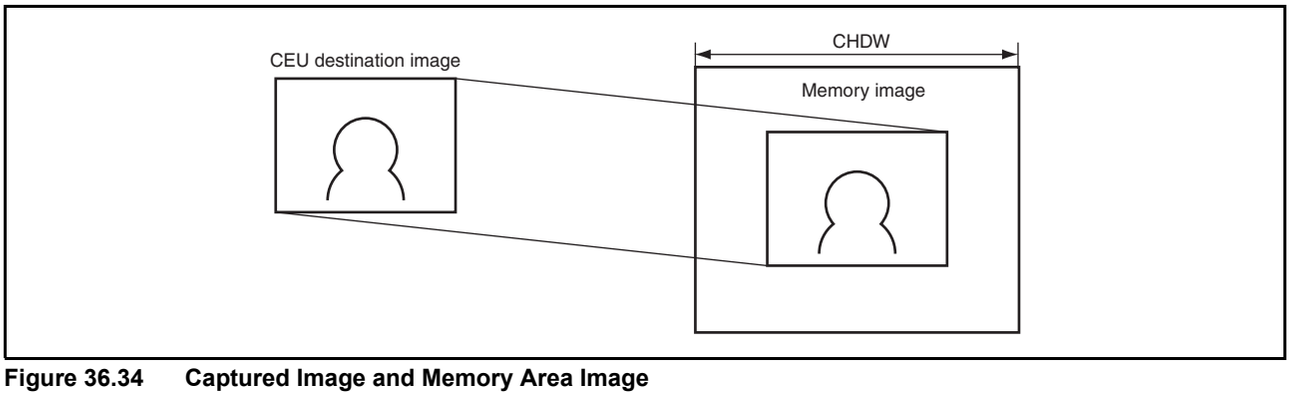


Figure 36.34 Captured Image and Memory Area Image

chanhvinh.com

36.4.13 Capture Data Address Y Register (CDAYR)

CDAYR specifies the address where the luminance (Y) component of the captured data is to be stored in frame image capture or one-field image capture, the address where the luminance (Y) component of the captured top field is to be stored in both-field image capture, and the address where the fetched data is to be stored in data fetch. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. In frame image capture or one-field image capture, set the start address of the memory area where the Y (luminance) component of the captured data is to be stored by CDAYR. In both-field image capture, set the start address of the memory area where the Y (luminance) component of the captured top-field image is to be stored by CDAYR. In data fetch, set the start address of the memory area where data is to be stored by CDAYR.

Because the address must be specified in 32 bits, the address set by CDAYR must be in longword units. As the setting is in 4-pixel units for image capture and 4-byte units for data fetch, the lower two bits are always fixed to 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CAYR[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CAYR[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	CAYR[31:2]	H'0000 0000	R/W	• Frame image capture: These bits set the address for storing the Y (luminance) component data of the captured data (4-pixel units). • One-field image capture: These bits set the address for storing the Y (luminance) component data of the captured data (4-pixel units). • Both-field image capture: These bits set the address for storing the Y (luminance) component data of the captured top-field data (4-pixel units). • Data fetch: These bits set the address for storing data (4-byte units). • Data enable fetch bundle write: These bits set the address for storing data (32-byte units).
1, 0	CAYR[1:0]		R	

Set the address of the starting point of the memory area where the fetched data is to be stored in this register, as shown in Figure 36.35.

- Frame image capture: Set the address of the starting point of the memory area where the Y component of the captured image is to be stored.
- One-field image capture: Set the address of the starting point of the memory area where the Y component of the captured image is to be stored.
- Both-field image capture: Set the address of the starting point of the memory area where the Y component of the captured top-field image is to be stored.
- Data fetch: Set the address of the starting point of the memory area where the fetched data is to be stored. In data fetch mode, the data is simply stuffed in order from the start address so the end address becomes as follows:

  End address = CDAYR + number of fetched bytes

- Data enable fetch bundle write: Set the address in 32-byte units.

chanhvinh.com

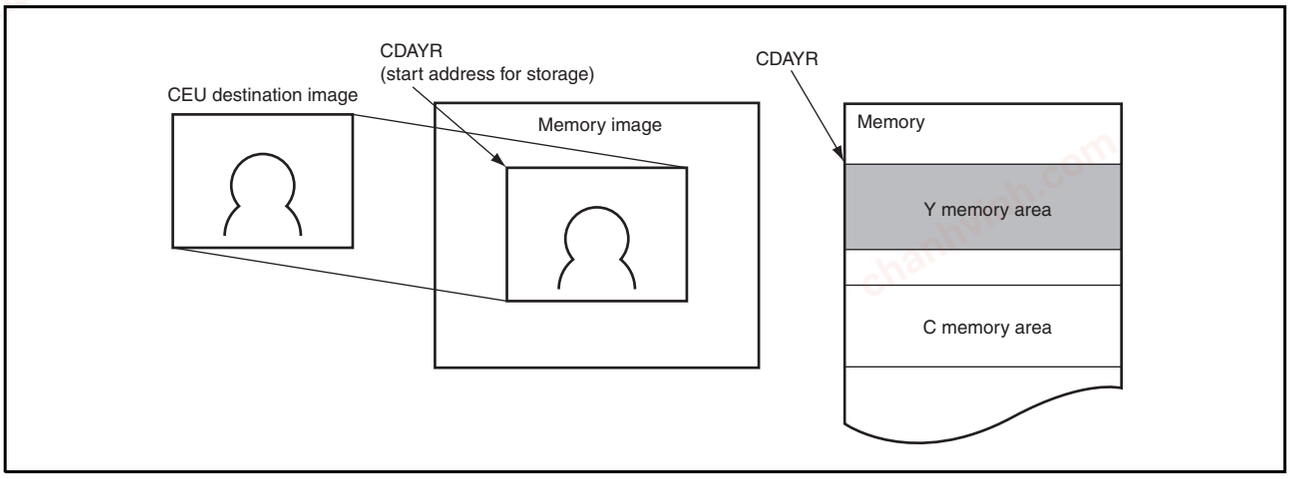


Figure 36.35 Relationship between Captured Image and Y Component Memory Area

36.4.14 Capture Data Address C Register (CDACR)

CDACR specifies the address where the chrominance (C) component of the captured data is to be stored in frame image capture or one-field image capture, and the address where the chrominance (C) component of the captured top field is to be stored in both-field image capture. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. In frame image capture or one-field image capture, set the start address of the memory area where the C (chrominance) component of the captured data is to be stored by CDACR. In both-field image capture, set the start address of the memory area where the C (chrominance) component of the captured top-field image is to be stored by CDACR. CDACR is not used in data fetch.

Because the address must be specified in 32 bits, the address set by CDACR must be in longword units. As the setting is in 4-pixel units, the lower two bits are always fixed to 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CACR[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CACR[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	CACR[31:2]	H'0000 0000	R/W	• Frame image capture: These bits set the address for storing the C (chrominance) component data of the captured data (4-pixel units). • One-field image capture: These bits set the address for storing the C (chrominance) component data of the captured data (4-pixel units). • Both-field image capture: These bits set the address for storing the C (chrominance) component data of the captured top-field data (4-pixel units).
1, 0	CACR[1:0]		R	

chanhvinh.com

Set the address of the starting point of the memory area where the C component of the captured image is to be stored in this register, as shown in Figure 36.36. The C component has an output data format like that in Figure 36.37, and is saved in the memory in this format.

- Frame image capture: Set the address of the starting point of the memory area where the C component of the captured image is to be stored.
- One-field image capture: Set the address of the starting point of the memory area where the C component of the captured image is to be stored.
- Both-field image capture: Set the address of the starting point of the memory area where the C component of the captured top-field image is to be stored.

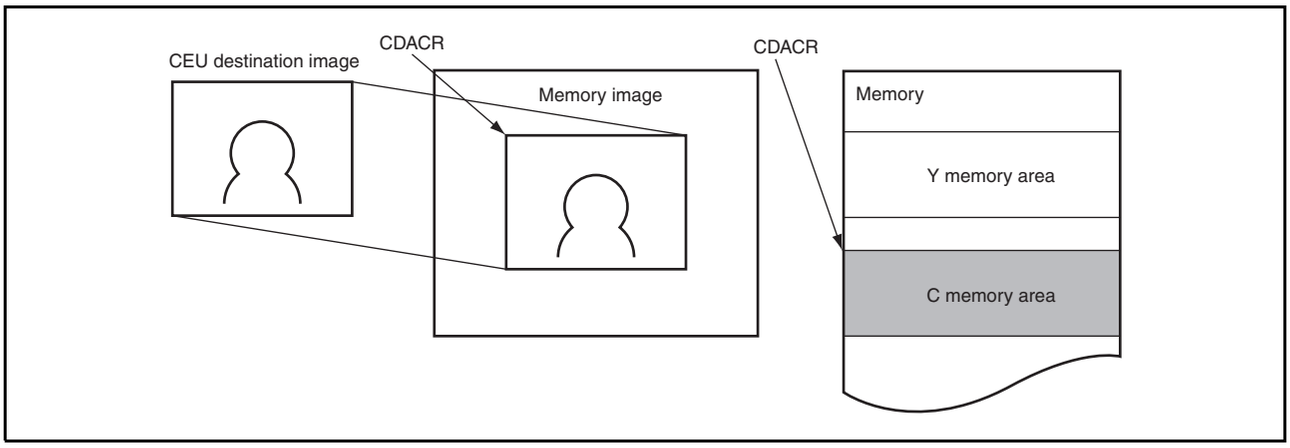


Figure 36.36 Relationship between Captured Image and C Component Memory Area

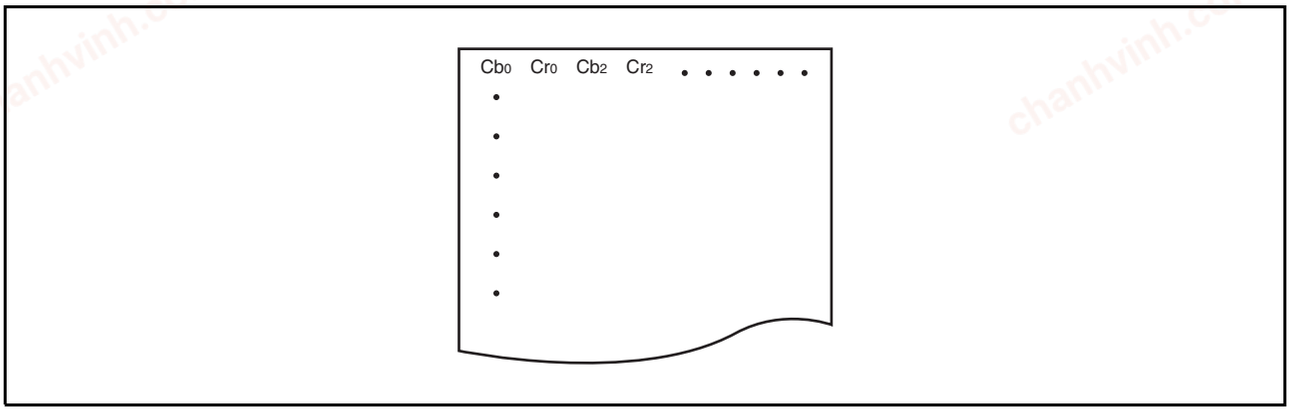


Figure 36.37 Image of Storing C Components in Memory

chanhvinh.com

36.4.15 Capture Data Bottom-Field Address Y Register (CDBYR)

CDBYR specifies the address where the luminance (Y) component of the captured bottom-field data is to be stored in both-field image capture. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. Set the start address of the memory area where the Y (luminance) component of the captured bottom-field image is to be stored by CDBYR. CDBYR is not used in frame image capture, one-field image capture, or data fetch.

Because the address must be specified in 32 bits, the address set by CDBYR must be in longword units. As the setting is in 4-pixel units, the lower two bits are always fixed to 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CBYR[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CBYR[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	CBYR[31:2]	H'0000 0000	R/W	These bits set the address for storing the Y (luminance) component data of the captured bottom-field data (4-pixel units).
1, 0	CBYR[1:0]		R	

Set the address of the starting point of the memory area where the Y component of the captured bottom-field image is to be stored in this register, as shown in Figure 36.38.

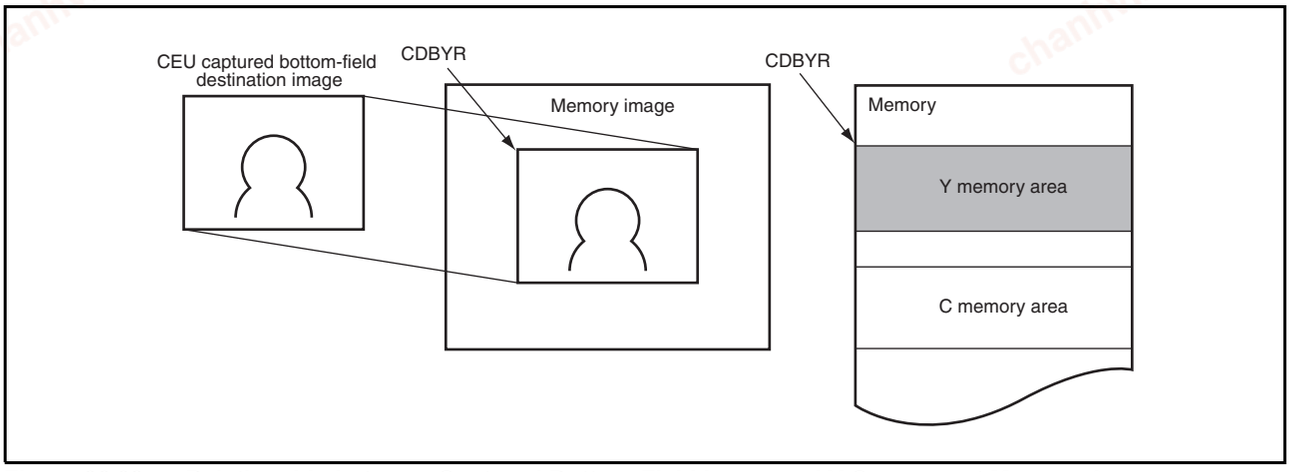


Figure 36.38 Relationship between Captured Bottom-Field Image and Y Component Memory Area

chanhvinh.com

36.4.16 Capture Data Bottom-Field Address C Register (CDBCR)

CDBCR specifies the address where the chrominance (C) component of the captured bottom-field data is to be stored in both-field image capture. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. Set the start address of the memory area where the C (chrominance) component of the captured bottom-field image is to be stored by CDBCR. CDBCR is not used in frame image capture, one-field image capture, or data fetch.

Because the address must be specified in 32 bits, the address set by CDBCR must be in longword units. As the setting is in 4-pixel units, the lower two bits are always fixed to 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CBCR[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CBCR[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	CBCR[31:2]	H'0000 0000	R/W	These bits set the address for storing the C (chrominance) component data of the captured bottom-field data (4-pixel units).
1, 0	CBCR[1:0]		R	

Set the address of the starting point of the memory area where the C component of the captured bottom-field image is to be stored in this register, as shown in Figure 36.39. The C component has an output data format like that in Figure 36.40, and is saved in the memory in this format.

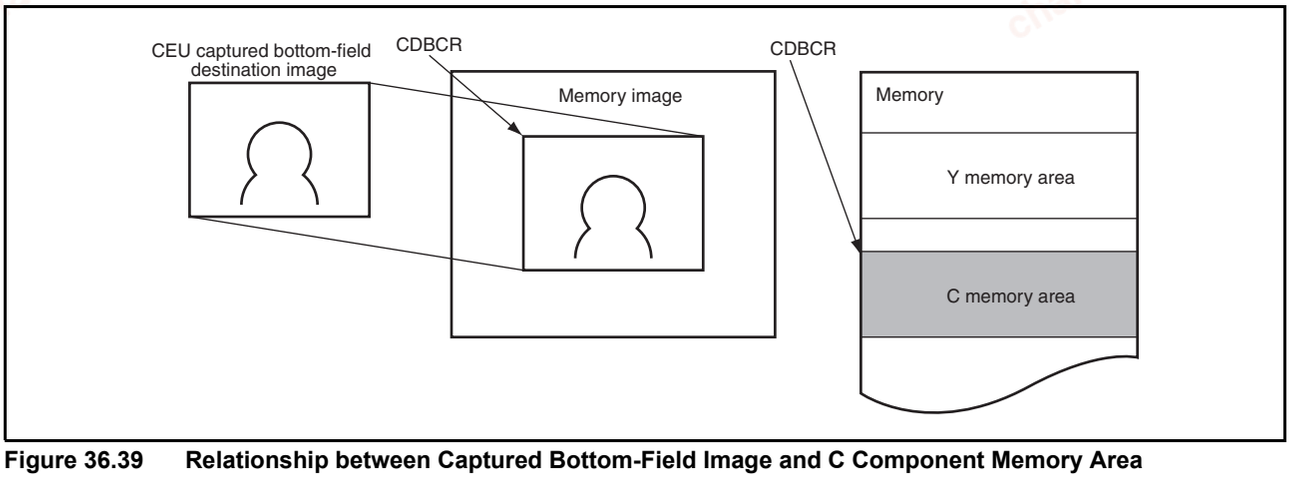


Figure 36.39 Relationship between Captured Bottom-Field Image and C Component Memory Area

chanhvinh.com

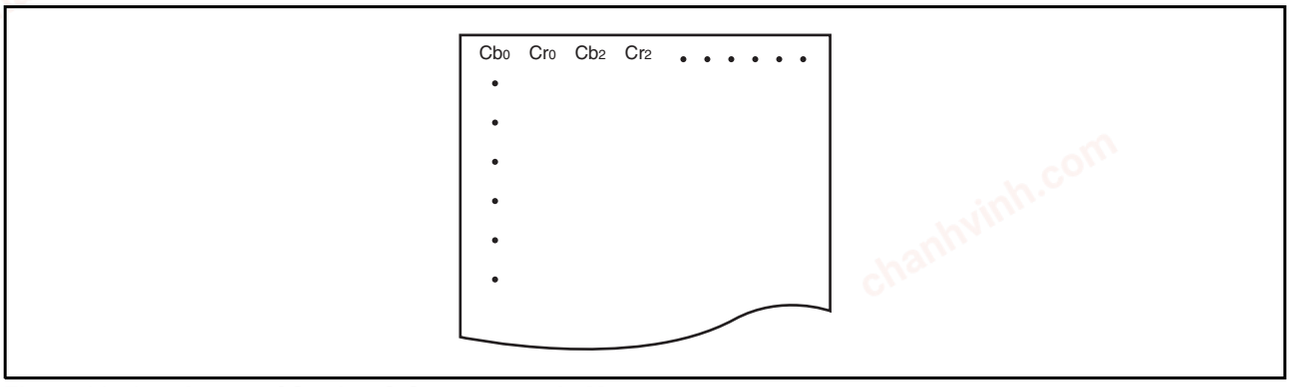


Figure 36.40 Image of Storing C Components in Memory

36.4.17 Capture Bundle Destination Size Register (CBDSR)

CBDSR sets the size of output to memory in a bundle write. The number of output lines should be specified for image capture or data synchronous fetch. The number of bytes should be specified for data enable fetch.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	CBVS[22:16]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CBVS[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
22 to 0	CBVS[22:3] CBVS[2:0]	H'000	R/W R	These bits select the number of lines or number of bytes for output to the memory in a bundle write. Image capture and data synchronous fetch: Number of lines for output to the memory in a bundle write. Unit: 8 lines, min.: 8 lines, max.: 1,920 lines (H'780) Data enable fetch: Number of bytes for output to the memory in a bundle write. Unit: 32 bytes, min.: 512 bytes, max.: 6291456 lines (H'600000)

(a) Image capture and data synchronous fetch

Set the number of lines of captured data to be written to the memory by a bundle write as a multiple of eight. This register is valid only when the CBE bit in CDOCR is 1. When the CBE bit in CDOCR is 1 and this register cleared to H'0, this module operates with the number of lines of captured data to be written to the memory as eight. The maximum number of lines that can be set is 1,920 (H'780). Only bits CBVS[11:3] are valid.

chanhvinh.com

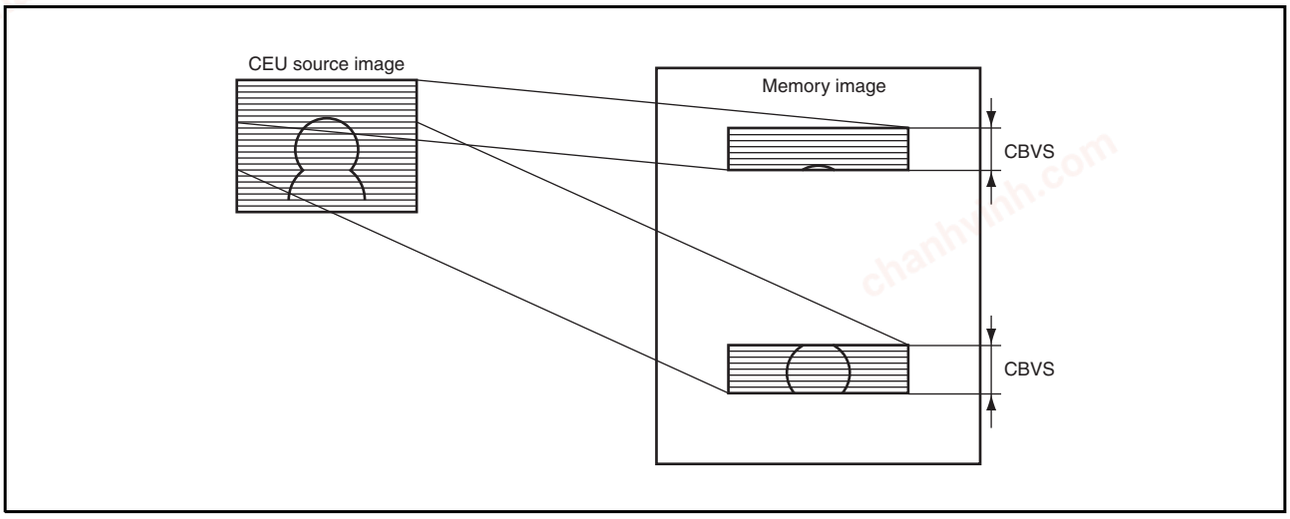


Figure 36.41 Image of Storing Captured Image in Memory by Bundle Write

(b) Data enable fetch

Set the number of bytes of captured data to be written to the memory by a bundle write as a multiple of 32. This register is valid only when the CBE bit in CDOCR is 1. The minimum settable size is 512 bytes. When a number smaller than 512 bytes is specified, operation is not guaranteed.

36.4.18 Capture Low-Pass Filter Control Register (CLFCR)

CLFCR specifies whether or not to operate the low-pass filter. In data fetch mode, clear the LPF bit to B'0.

The characteristic of the low-pass filter installed in the CEU causes the phase location of the image processed by the low-pass filter to be shifted right by one pixel compared to the raw image.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LPF
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	LPF	0	R/W	Enables or disables operation of the low-pass filter. The low-pass filter removes high-frequency components from the destination image in the horizontal direction. Clear this bit to 0 in data fetch mode. 0: Low-pass filter not used 1: Low-pass filter used (only in the horizontal direction)

chanhvinh.com

36.4.19 Firewall Operation Control Register (CFWCR)

CFWCR specifies the upper limit of the write addresses in data enable fetch. When the VD input from an external module dose not go low and end notification is not given, this register can prevent writing to memory from being out of control.

This register is enabled only in data enable fetch.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FWV[26:11]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FWV[10:0]											—	—	—	—	FWE
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	FWV[26:0]	H'0000008	R/W	These bits specify the upper limit of a write address. Specify the upper 27 bits of the 32-bit address. The upper limit of an address is FWV[26:0]<<5 + H'1F.
4 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	FWE	0	R/W	With the setting of FWE = 1, when an address exceeds the value set with FWV, the address is retained and an interrupt source FWF is set. After this, the address is not incremented and data is overwritten on the upper limit address. 0: Firewall is not activated. 1: Firewall is activated.

36.4.20 Capture Data Output Control Register (CDOCR)

CDOCR sets the format for outputting captured data to the memory. In data fetch mode, set the CDS bit to B'1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CBE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	CDS	—	COLS	COWS	COBS
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
16	CBE	0	R/W	Controls the number of lines of captured data to be written to the memory. • Image capture This bit controls the number of lines of captured data to be written to the memory. When bundle write is set by this register, captured data is written in line units specified by CBDSR to the addresses specified by CDAYR and CDACR, and CDAYR2 and CDACR2 (CDBYR and CDBCR, and CDBYR2 and CDBCR2 for the bottom field in both-field capture) alternately (Figure 36.42). When captured data has been written for the number of lines set by CBDSR, a write end interrupt corresponding to each address setting register occurs. However, after write for one-frame (one-field) capture ends, a bundle write end interrupt does not occur even when bundle write has finished. • Data synchronous fetch This bit controls the number of lines of captured data to be written to the memory. When bundle write is set by this register, captured data is written in line units specified by CBDSR to the addresses specified by CDAYR and CDAYR2 alternately. When captured data has been written for the number of lines set by CBDSR, a write end interrupt corresponding to each address setting register occurs. However, after write for one-frame capture ends, a bundle write end interrupt does not occur even when bundle write has finished. • Data enable fetch This bit controls the number of bytes of captured data to be written to the memory. When bundle write is set by this register, captured data is written in byte units specified by CBDSR to the addresses specified by CDAYR and CDAYR2 alternately. When captured data has been written for the number of bytes set by CBDSR, a write end interrupt corresponding to each address setting register occurs. Also, only in data enable fetch, a bundle write end interrupt occurs when bundle write has finished after write for one-frame capture ends. Table 36.9 shows the correspondence between address setting registers and write end interrupt sources. Figure 36.43 shows the timing of write end interrupts in image capture and data synchronous fetch. Figure 36.44 shows the timing of write end interrupts in data enable fetch. 0: Normal write 1: Bundle write
15 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	CDS	0	R/W	Sets the image format when outputting the image data captured in the YCbCr422 format to the memory. When 0 is written to this bit, only the luminance component (Y) is output and no chrominance components (Cb and Cr) are output for the odd-numbered lines. With an interlace source image, similarly only the luminance component (Y) is output and no chrominance components (Cb and Cr) are output for the odd-numbered lines of the field. In data fetch mode, set this bit to 1. 0: Converts the YCbCr422 format to the YCbCr420 format before outputting data to the memory 1: Outputs data in the YCbCr422 format to the memory without conversion
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	COLS	0	R/W	Controls swapping in 32-bit units for data output from the CEU. 0: Data is not swapped in 32-bit units 1: Data is swapped in 32-bit units
1	COWS	0	R/W	Controls swapping in 16-bit units for data output from the CEU. 0: Data is not swapped in 16-bit units 1: Data is swapped in 16-bit units
0	COBS	0	R/W	Controls swapping in 8-bit units for data output from the CEU. 0: Data is not swapped in 8-bit units 1: Data is swapped in 8-bit units

chanhvinh.com

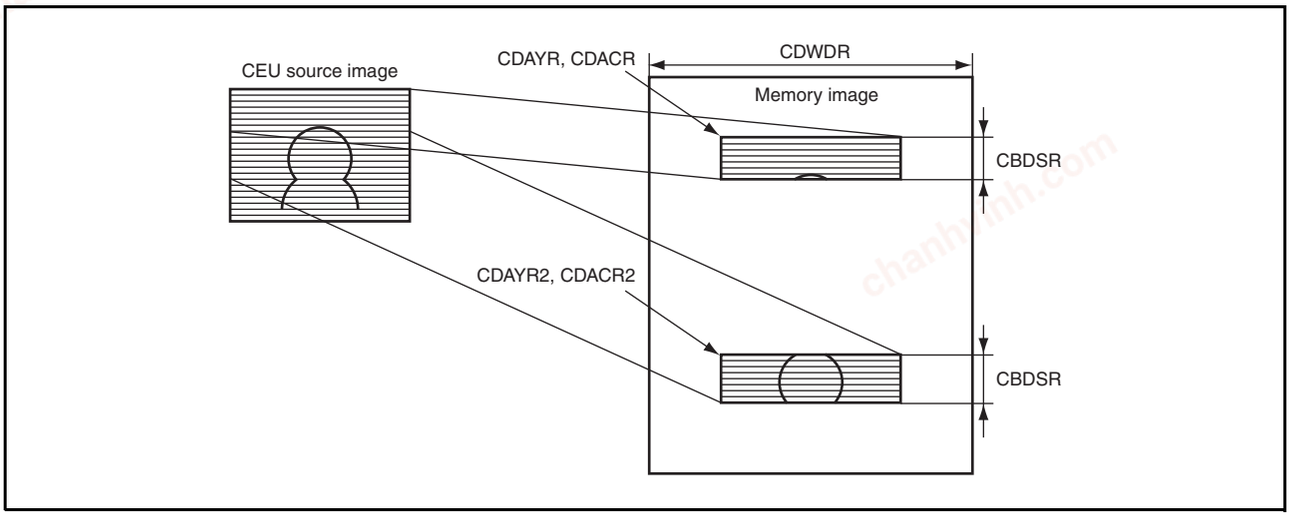


Figure 36.42 Image of Bundle Write to Memory

Table 36.9 Correspondence between Address Setting Registers and Write End Interrupt Sources

Address Setting Registers	Bundle Write End Interrupt Source
CDAYR, CDACR	CPBE1 bit in CETCR
CDAYR2, CDACR2	CPBE2 bit in CETCR
CDBYR, CDBCR	CPBE3 bit in CETCR
CDBYR2, CDBCR2	CPBE4 bit in CETCR

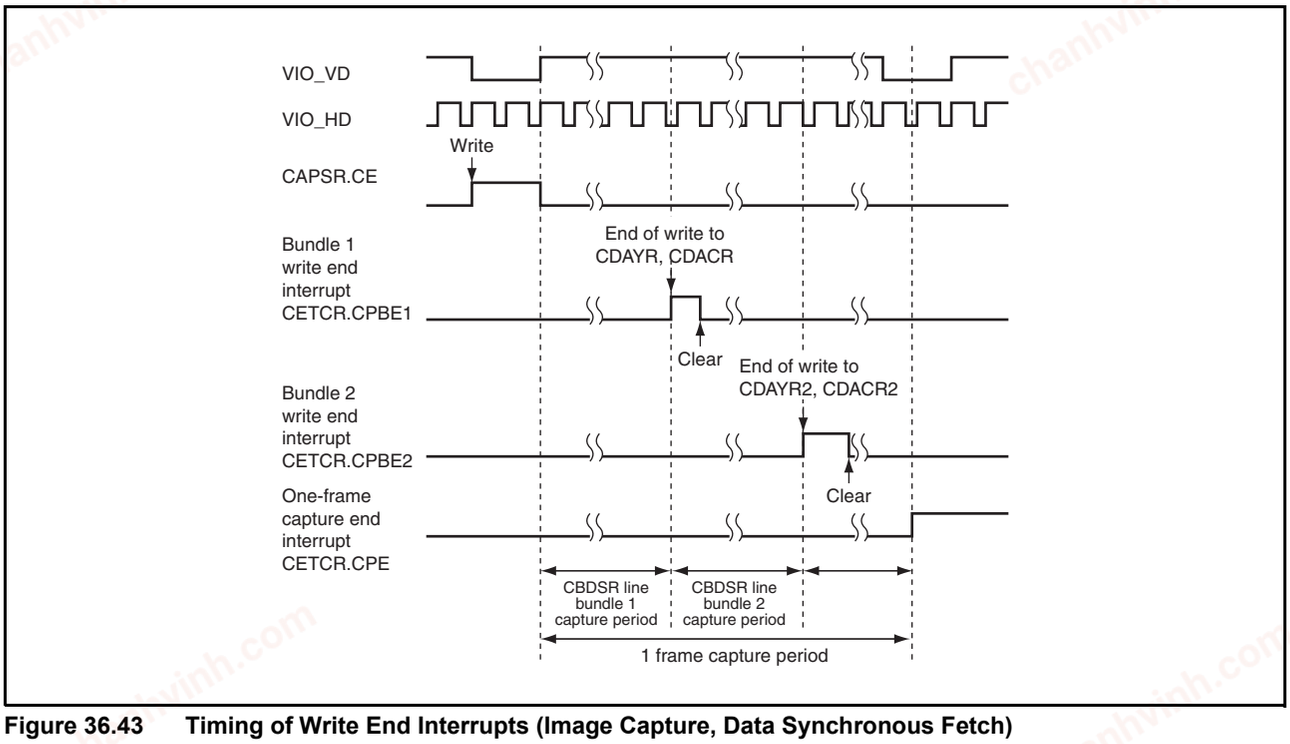


Figure 36.43 Timing of Write End Interrupts (Image Capture, Data Synchronous Fetch)

chanhvinh.com

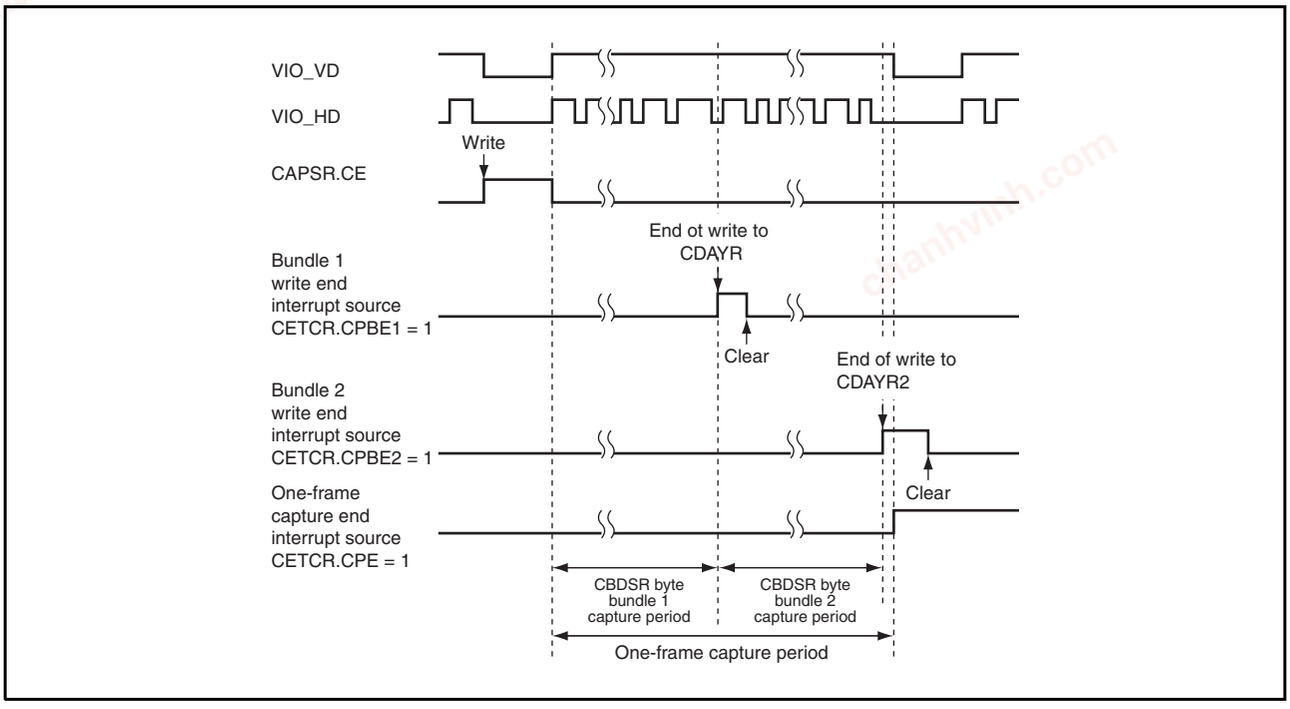


Figure 36.44 Timing of Write End Interrupts (Data Enable Fetch)

For data output from the CEU, the COLS, COWS, and COBS bits control swapping in 32-bit, 16-bit, and 8-bit units, respectively. Set these bits when data is misaligned because of a difference in endian. The data swapping bits are shown below. These bits can be set similarly in data fetch mode.

Data can be swapped in 8-bit, 16-bit, 32-bit units, or in 32 bits, 16 bits and 8 bits, as shown in Figure 36.45. To enable data swapping, set the corresponding control bit to B'1.

chanhvinh.com

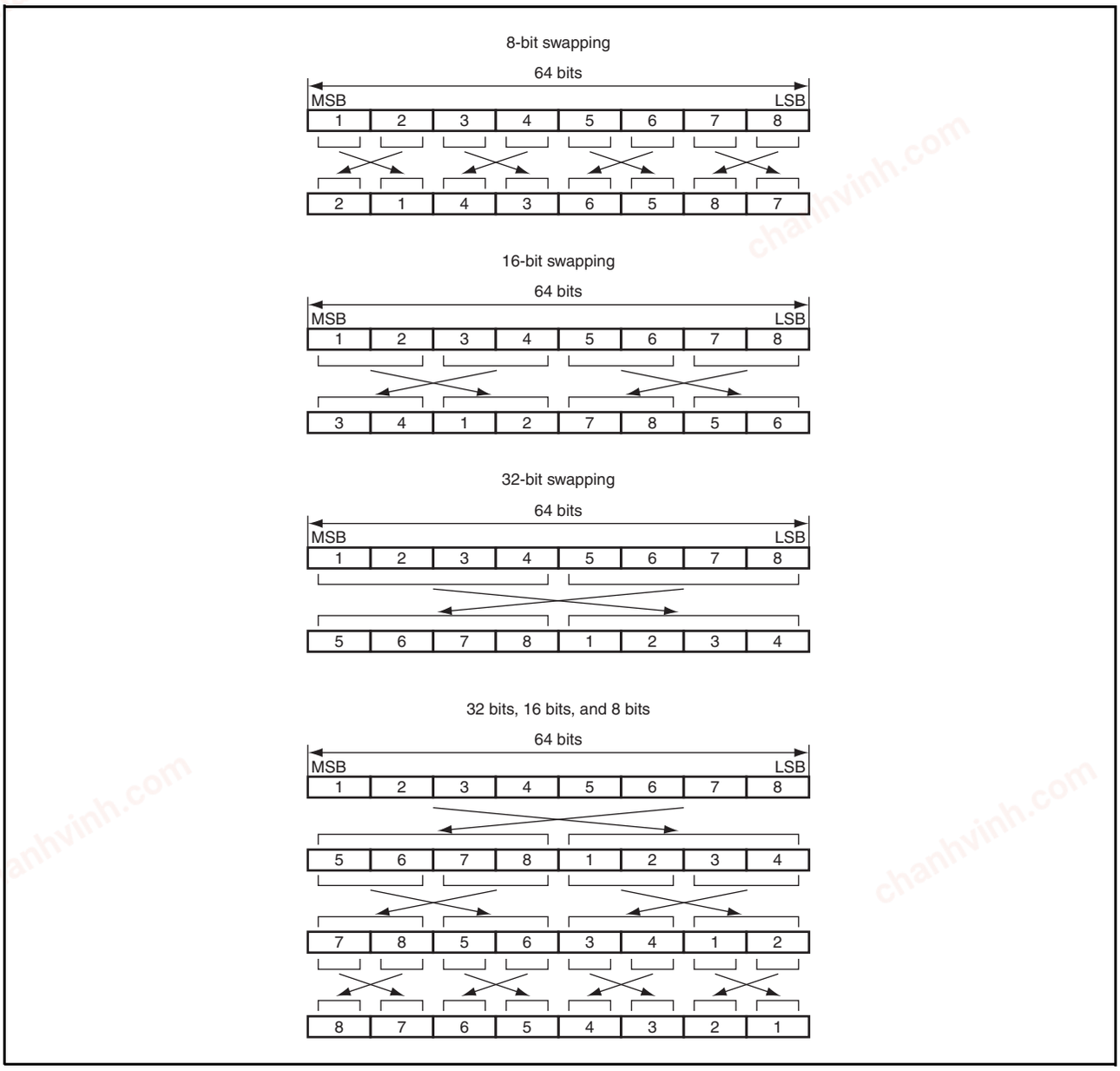


Figure 36.45 Data Swapping by Data Aligner

36.4.21 Capture Event Interrupt Enable Register (CEIER)

CEIER enables or disables interrupts of the event flag register that generates CEU interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	NVDIE	NHDIE	FWFIE	—	—	VBPIE	—	IGVSIE	IGHSIE	CDTOFIE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R	R	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CPBE4IE	CPBE3IE	CPBE2IE	CPBE1IE	—	—	VDIE	HDIE	—	—	—	IGRWIE	—	—	CFEIE	CPEIE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R	R/W	R	R	R/W	R/W

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25	NVDIE	0	R/W	Non-VD Interrupt Enable Disable this interrupt (NVDIE = 0) for data enable fetch. 0: Disables a non-VD interrupt 1: Enables a non-VD interrupt
24	NHDIE	0	R/W	Non-HD Interrupt Enable Disable this interrupt (NHDIE = 0) for data enable fetch. 0: Disables a non-HD interrupt 1: Enables a non-HD interrupt
23	FWFIE	0	R/W	FWF Interrupt Enable 0: Disables a FWF interrupt 1: Enables a FWF interrupt
22, 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	VBPIE	0	R/W	VBP Interrupt Enable 0: Disables a VBP interrupt 1: Enables a VBP interrupt
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18	IGVSIE	0	R/W	IGVS Interrupt Enable 0: Disables an IGVS interrupt 1: Enables an IGVS interrupt
17	IGHSIE	0	R/W	IGHS Interrupt Enable 0: Disables an IGHS interrupt 1: Enables an IGHS interrupt
16	CDTOFIE	0	R/W	CDTOF Interrupt Enable 0: Disables a CDTOF interrupt 1: Enables a CDTOF interrupt
15	CPBE4IE	0	R/W	CPBE4 Interrupt Enable 0: Disables a CPBE4 interrupt 1: Enables a CPBE4 interrupt
14	CPBE3IE	0	R/W	CPBE3 Interrupt Enable 0: Disables a CPBE3 interrupt 1: Enables a CPBE3 interrupt
13	CPBE2IE	0	R/W	CPBE2 Interrupt Enable 0: Disables a CPBE2 interrupt 1: Enables a CPBE2 interrupt
12	CPBE1IE	0	R/W	CPBE1 Interrupt Enable 0: Disables a CPBE1 interrupt 1: Enables a CPBE1 interrupt
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	VDIE	0	R/W	VD Interrupt Enable 0: Disables a VD interrupt 1: Enables a VD interrupt
8	HDIE	0	R/W	HD Interrupt Enable 0: Disables an HD interrupt 1: Enables an HD interrupt
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	IGRWIE	0	R/W	Register-Access-During-Capture Interrupt Enable 0: Disables a register-access-during-capture interrupt 1: Enables a register-access-during-capture interrupt
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
1	CFEIE	0	R/W	CFE Interrupt Enable 0: Disables a CFE interrupt 1: Enables a CFE interrupt
0	CPEIE	0	R/W	One-Frame Capture End Interrupt Enable 0: Disables a one-frame capture end interrupt 1: Enables a one-frame capture end interrupt

36.4.22 Capture Event Flag Clear Register (CETCR)

CETCR notifies the CPU of the source of an interrupt that is generated in the CEU. The flags of CETCR can be used as interrupt signals. When the corresponding interrupt is enabled, an interrupt is generated. To clear the interrupt, clear the bit corresponding to the interrupt source to 0. After several cycles have passed after modifying the bit, the interrupt is cleared.

To clear the bit corresponding to the interrupt source to be cleared to 0 and retain that state, write 1 to that bit. For example, to clear only the CPE bit to 0, write H'FFFF FFFE to CETCR.

In CETCR, only bits to which 0 is written are cleared. Bits to which 1 is written retain their current values. To clear an interrupt source, write 0 only to the bit corresponding to the interrupt source to be cleared, and 1 to the other bits.

Note: Since the CETCR value becomes undefined in the following cases, clear all bits in CETCR to 0.

- VD and HD bits immediately after power-on reset or the deep-standby mode is entered
- All bits after the software standby or module standby mode is entered
- VD and HD bits after the polarities of the capture interface sync signals are changed

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	NVD	NHD	FWF	—	—	VBP	—	IGVS	IGHS	CDTOF
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R	R	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CPBE4	CPBE3	CPBE2	CPBE1	—	—	VD	HD	—	—	—	IGRW	—	—	CFE	CPE
Initial value:	0	0	0	0	0	0	—	—	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25	NVD	0	R/W	This bit is used for an interrupt indicating that no VD was input. A non-VD interrupt occurs when the 14-bit internal counter becomes full. Accordingly, this bit is set to 1 when no VD has been input for at least 16,383 lines since the last VD was input.
24	NHD	0	R/W	This bit is used for an interrupt indicating that no HD was input. A non-HD interrupt occurs when the 11-bit internal counter that is incremented every eight cycles becomes full. Accordingly, this bit is set to 1 when no HD has been input for at least 16,376 cycles since the last HD was input. When connecting a camera whose HD is fixed low when VD is low, this bit may be set to 1. Ignore this interrupt during data enable fetch.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
23	FWF	0	R/W	The interrupt is generated when data is written to the address that exceeds the value specified with CFWCR.FMV. This bit is set to 1 when data is written to the address that exceeds the value specified with CFWCR.FMV while CFWCR.FWE = 1.
22, 21	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	VBP	0	R/W	This bit is used for an interrupt indicating that VD has been input while the CEU holds data (insufficient vertical-sync front porch). The conditions for a VBP interrupt to occur are as follows: • Condition 1 VD is input when there is captured data within the CEU • Condition 2 The last transfer data cannot be internally detected due to a write buffer overflow or an illegal HD so that the end timing is unclear until the next VD (By generating a VBP interrupt at the VD input timing, capture fail can be announced.) When a VBP interrupt occurs, a capture end interrupt (CPE bit in CETCR) does not occur and the image of that frame is not captured correctly. Though a capture end interrupt (CPE bit) will occur on rare occasions, it should be ignored in this case. Capturing cannot be performed until the next VD (even if the CE bit (capture reservation signal) in CAPSR is 1, capturing does not start). In the case of condition 2, instead of waiting for a VBP interrupt to occur, execute a software reset (CPKIL bit in CAPSR) to stop capturing and then restart capturing. In this case, since capture operation is terminated without waiting for the next VD, a VBP interrupt does not occur and capturing can be performed from the next VD.
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18	IGVS	0	R/W	This bit is used for an interrupt generated when the number of VD cycles set in CMCYR differ from the number of VD cycles input from an external module. This bit is set to 1 when there is an illegal VD input from an external module. This bit is set to 1 when the number of HD cycles for the VD input to the CEU differs from the value set in the VCYL bits in CMCYR. Note however that when the VCYL bits are cleared to 0, this interrupt does not occur.
17	IGHS	0	R/W	This bit is used for an interrupt generated when the number of HD cycles set in CMCYR differ from the number of HD cycles input from an external module. This bit is set to 1 when there is an illegal HD input from an external module. This bit is set to 1 when the number of clock cycles for the HD input to the CEU differs from the value set in the HCYL bits in CMCYR. Note however that when the HCYL bits are cleared to 0, this interrupt does not occur.
16	CDTOF	0	R/W	This bit is used for an interrupt indicating that data overflowed in the CRAM of the write buffer. Since data is input at realtime from an external module in capture operations, the frame image is overwritten unless the captured data is transferred from the CEU internal buffer to the memory at a certain or higher transfer rate. This bit is set to 1 when writing the data in the CRAM of the CEU internal write buffer to the bus is not performed within time and data has overflowed.
15	CPBE4	0	R/W	This bit is used for an interrupt indicating that writing to CDBYR2 and CDBCR2 in a bundle write has finished. This interrupt is output when the last captured data has been transferred and the end notification received, regardless of the next HD input. This bit is set to 1 when data for the number of lines set in CBDSR has been captured and the last data transfer to the bus has completed. However, this interrupt does not occur when the last captured data in a bundle write is the last captured data of a frame (field).
14	CPBE3	0	R/W	This bit is used for an interrupt indicating that writing to CDBYR and CDBCR in a bundle write has finished. This interrupt is output when the last captured data has been transferred and the end notification received, regardless of the next HD input. This bit is set to 1 when data for the number of lines set in CBDSR has been captured and the last data transfer to the bus has completed. However, this interrupt does not occur when the last captured data in a bundle write is the last captured data of a frame (field).

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
13	CPBE2	0	R/W	This bit is used for an interrupt indicating that writing to CDAYR2 and CDACR2 in a bundle write has finished. This interrupt is output when the last captured data has been transferred and the end notification received, regardless of the next HD input. This bit is set to 1 when data for the number of lines (number of bytes in data enable fetch) set in CBDSR has been captured and the last data transfer to the bus has completed. However, in image capture or data synchronous fetch, this interrupt does not occur when the last captured data in a bundle write is the last captured data of a frame (field).
12	CPBE1	0	R/W	This bit is used for an interrupt indicating that writing to CDAYR and CDACR in a bundle write has finished. This interrupt is output when the last captured data has been transferred and the end notification received, regardless of the next HD input. This bit is set to 1 when data for the number of lines (number of bytes in data enable fetch) set in CBDSR has been captured and the last data transfer to the bus has completed. However, in image capture or data synchronous fetch, this interrupt does not occur when the last captured data in a bundle write is the last captured data of a frame (field).
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	VD	Undefined	R/W	This bit is used for an interrupt indicating that VD (vertical sync signal) was input from an external module. In data enable fetch mode, this bit is set to 1 when a VD input from an external module is detected. In image capture mode and data synchronous fetch mode, this module generates a VD interrupt in response to the first detection of the active level of HD following detection of the active level of VD from an external module. Note that, when VD and HD are asserted and detected at the same time, this module generates a VD interrupt at that time. Immediately after the VDPOL bit in CAMCR is modified, a pseudo VD is input and this bit is set to 1. The VD interrupt after the VDPOL bit is modified should be ignored.
8	HD	Undefined	R/W	This bit is used for an interrupt indicating that HD (horizontal sync signal) was input from an external module. This bit is set to 1 when an HD input from an external module is detected. Immediately after the HDPOL bit in CAMCR is modified, a pseudo HD is input and this bit is set to 1. The HD interrupt after the HDPOL bit is modified should be ignored.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	IGRW	0	R/W	This bit is used for an interrupt indicating that during capturing, access was attempted to a register to which writing during operation is prohibited. Among the CEU registers, writing during capturing is prohibited for some registers. Table 36.10 shows which registers can/cannot be written to during capturing. This bit is set to 1 when a register to which writing during capturing is prohibited has been written to.
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	CFE	0	R/W	This bit is used for an interrupt indicating that capturing of one field from an external module has finished. This interrupt is output when the last captured data has been transferred and the end notification received, regardless of the next VD input (see Figure 36.46). This interrupt occurs only in both-field capture mode.
0	CPE	0	R/W	This bit is used for an interrupt indicating that capturing of one frame from an external module has finished. This interrupt is output when the last captured data has been transferred and the end notification received, regardless of the next VD input. This interrupt indicates that capturing of one frame has finished. This bit is set to 1 when the image of the size set in CAPWR is captured and the last data transfer to the bus finished (see Figure 36.47).

chanhvinh.com

Table 36.10 Registers that Can/Cannot Be Modified during Capturing

Register Name	Modification during Capturing
CAPSR	Possible
CAPCR	Prohibited
CAMCR	Prohibited
CMCYR	Prohibited
CAMOR	Possible
CAPWR	Possible
CAIFR	Prohibited
CRCNTR	Possible
CRCMPR	Prohibited
CFLCR	Possible
CFSZR	Possible
CDWDR	Possible
CDAYR	Possible
CDACR	Possible
CDBYR	Possible
CDBCR	Possible
CBDSR	Possible
CFWCR	Possible
CLFCR	Possible
CDOCR	Possible
CEIER	Possible
CETCR	Possible
CSTSR	Prohibited
CDSSR	Prohibited
CDAYR2	Possible
CDACR2	Possible
CDBYR2	Possible
CDBCR2	Possible

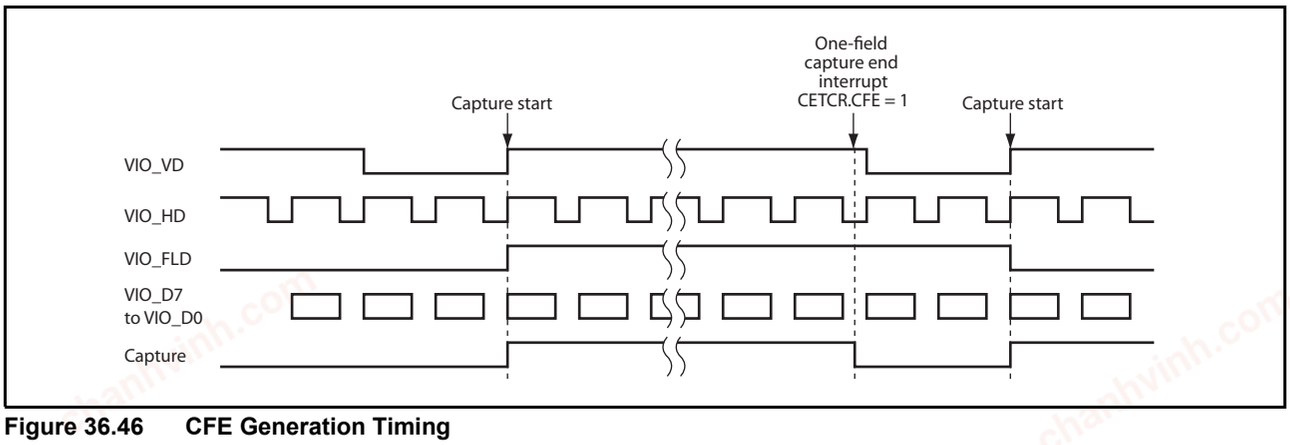


Figure 36.46 CFE Generation Timing

chanhvinh.com

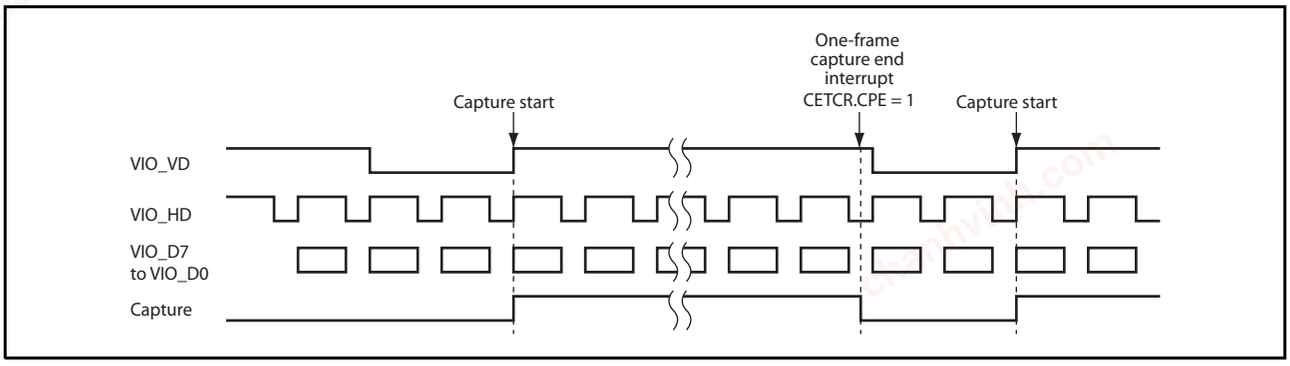


Figure 36.47 CPE Generation Timing

36.4.23 Capture Status Register (CSTSR)

CSTSR indicates the internal status of the CEU. CSTSR differs from CETCR in that no interrupt is generated for the events indicated in CSTSR.

The CEU operating/halt state can be confirmed using CSTSR. To confirm the halt state of the CEU, make sure that the status bit (bit 0) indicating that the CEU is operating is cleared to 0 for sure.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	CRST	—	—	—	—	—	—	—	CP FLD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CP TON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	CRST	0	R	Indicates which register plane is currently used. 0: Plane A of the register is being used 1: Plane B of the register is being used
23 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	CPFLD	0	R	Indicates which field is being captured. 0: Bottom field is being captured 1: Top field is being captured
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	CPTON	0	R	Indicates that the CEU is operating. This bit retains 1 during the period that starts from the internal VD at capture start and ends when a one-frame capture end interrupt occurs. Figure 36.48 shows the CEU operating period.

chanhvinh.com

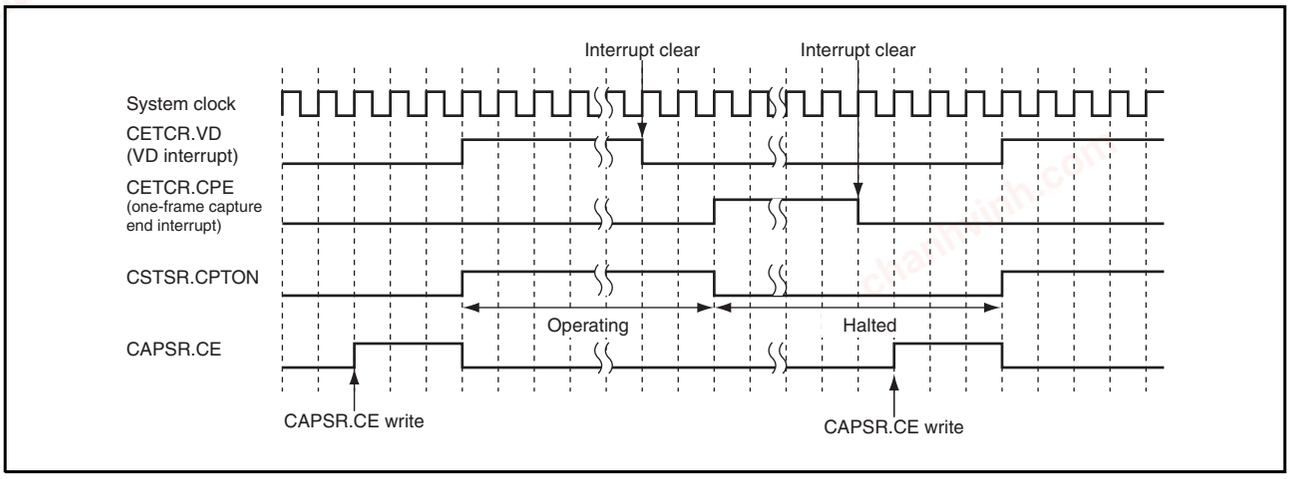


Figure 36.48 Operating Status during Capturing

chanhvinh.com

36.4.24 Capture Data Size Register (CDSSR)

CDSSR indicates the size of data written to the memory in data enable fetch. As this register indicates a correct value at the end of capture, confirm this register when capture is completed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CDSS[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CDSS[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CDSS[31:0]	H'0000 0000	R	Indicate the size of data written to the memory in data enable fetch. In a bundle write, size of data written to the selected address at the end of one-frame capture is indicated. In a bundle write, as soon as the number of bytes specified by CBDSR is transferred to the bus, address to which data is written is switched. Therefore, if one-frame capture is completed at the same time as a bundle write is completed, this register indicates H'0000 0000. Figure 36.49 and Figure 36.50 show the overall timing of the CDSSR operation in a bundle write.

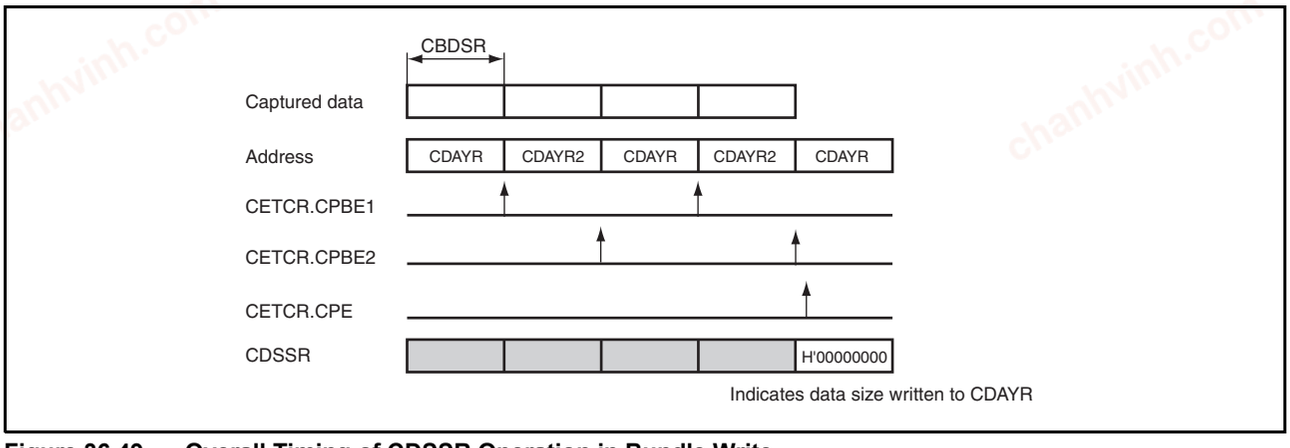


Figure 36.49 Overall Timing of CDSSR Operation in Bundle Write (When Bundle Write End and Capture End Coincide)

chanhvinh.com

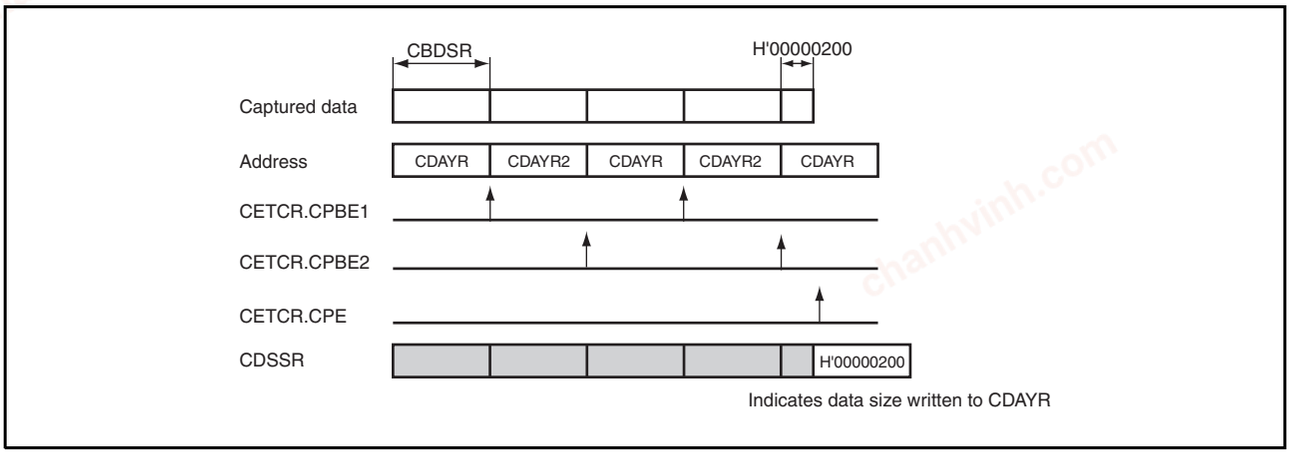


Figure 36.50 Overall Timing of CDSSR Operation in Bundle Write (When Bundle Write End and Capture End Do Not Coincide)

36.4.25 Capture Data Address Y Register 2 (CDAYR2)

CDAYR2 specifies the address for the luminance (Y) component used in a bundle write and the address for data storage in a bundle write in data fetch. CDAYR2 is used only in a bundle write.

CDAYR2 specifies the address where the Y component of the captured data is to be stored in frame image capture or one-field image capture, the address where the Y component of the captured top field is to be stored in both-field image capture, and the address where data is to be stored in data fetch. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. In frame image capture or one-field image capture, set the start address of the memory area where the Y component of the captured data is to be stored by CDAYR2. In both-field image capture, set the start address of the memory area where the Y component of the captured top-field image is to be stored by CDAYR2. In data fetch, set the start address of the memory area to be used for data storage.

The address specified by this register must be in 32-bit units. As the setting is in 4-pixel units, the lower two bits are always fixed to 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CAYR2[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CAYR2[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	CAYR2[31:2]	H'0000 0000	R/W	• Frame image capture: These bits set the address for storing the Y component data of the captured data (4-pixel units). • One-field image capture: These bits set the address for storing the Y component data of the captured data (4-pixel units). • Both-field image capture: These bits set the address for storing the Y component data of the captured top-field data (4-pixel units). • Data synchronous fetch: These bits set the address for storing data (4-byte units). • Dana enable fetch: These bits set the address for storing data (32-byte units).
1, 0	CAYR2[1:0]		R	

chanhvinh.com

Set the address of the starting point of the memory area where the Y component of the captured image is to be stored in this register, as shown in Figure 36.51.

- Frame image capture: Set the address of the starting point of the memory area where the Y component of the captured image is to be stored.
- One-field image capture: Set the address of the starting point of the memory area where the Y component of the captured image is to be stored.
- Both-field image capture: Set the address of the starting point of the memory area where the Y component of the captured top-field image is to be stored.

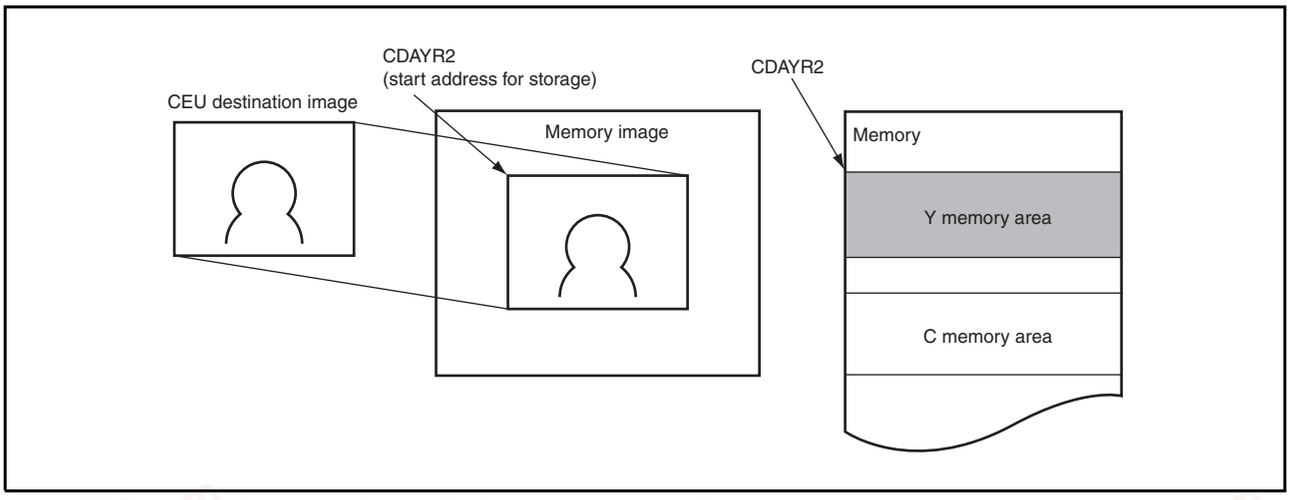


Figure 36.51 Relationship between Captured Image and Y Component Memory Area

chanhvinh.com

36.4.26 Capture Data Address C Register 2 (CDACR2)

CDACR2 specifies the address for the chrominance (C) component used in a bundle write. CDACR2 is used only in a bundle write.

CDACR2 specifies the address where the C component of the captured data is to be stored in frame image capture or one-field image capture, and the address where the C component of the captured top field is to be stored in both-field image capture. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. In frame image capture or one-field image capture, set the start address of the memory area where the C component of the captured data is to be stored by CDACR2. In both-field image capture, set the start address of the memory area where the C component of the captured top-field image is to be stored by CDACR2. CDACR2 is not used in data fetch.

The address specified by this register must be in 32-bit units. As the setting is in 4-pixel units, the lower two bits are always fixed to 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CACR2[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CACR2[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 2 1, 0	CACR2[31:2] CACR2[1:0]	H'0000 0000	R/W R	• Frame image capture: These bits set the address for storing the C component data of the captured data (4-pixel units). • One-field image capture: These bits set the address for storing the C component data of the captured data (4-pixel units). • Both-field image capture: These bits set the address for storing the C component data of the captured top-field data (4-pixel units).

In this register, set the address of the starting point of the memory area where the captured data is to be stored in a bundle write, as shown in Figure 36.52. The C component has an output data format as shown in Figure 36.53, and is saved in the memory in this format.

- Frame image capture: Set the address of the starting point of the memory area where the C component of the captured image is to be stored.
- One-field image capture: Set the address of the starting point of the memory area where the C component of the captured image is to be stored.
- Both-field image capture: Set the address of the starting point of the memory area where the C component of the captured top-field image is to be stored.

The C component has an output data format as shown in Figure 36.52, and is saved in the memory in this format.

chanhvinh.com

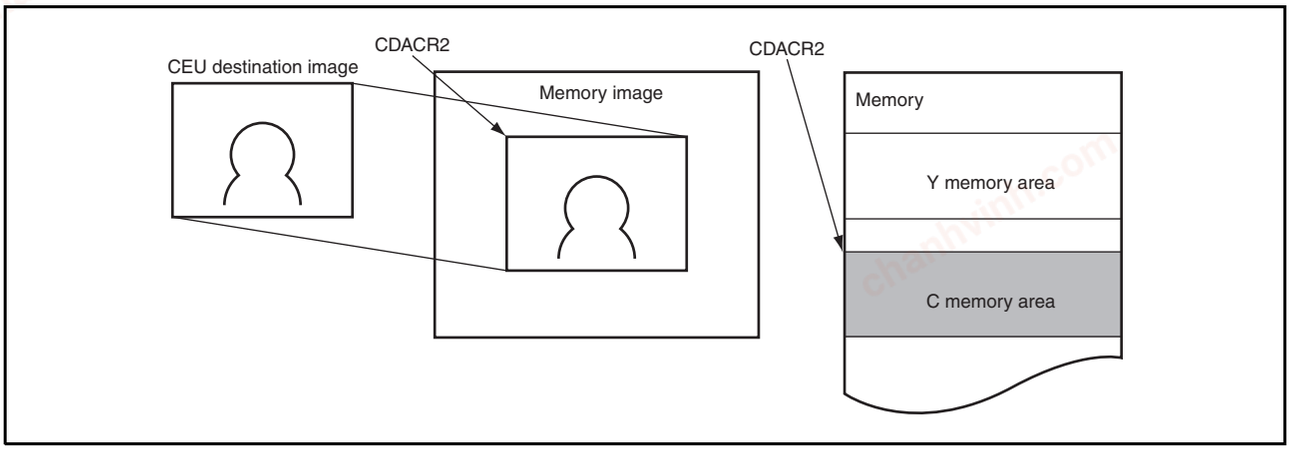


Figure 36.52 Relationship between Captured Image and C Component Memory Area

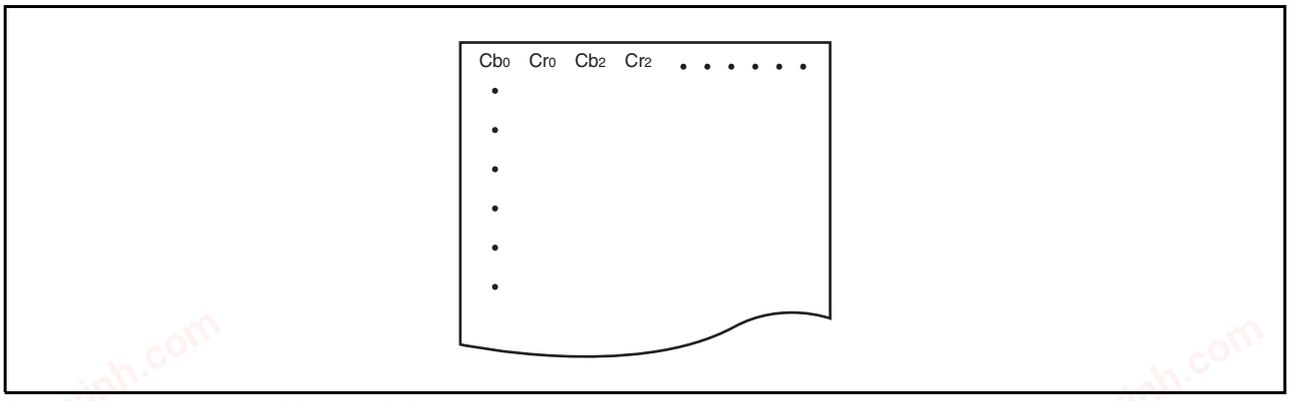


Figure 36.53 Image of Storing C Components in Memory

chanhvinh.com

36.4.27 Capture Data Bottom-Field Address Y Register 2 (CDBYR2)

CDBYR2 specifies the address for the luminance (Y) component of the bottom field used in a bundle write. CDBYR2 is used only in a bundle write.

CDBYR2 specifies the address where the Y component of the captured bottom-field data is to be stored in both-field image capture. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. Set the start address of the memory area where the Y component of the bottom-field image captured in both-field image capture is to be stored by CDBYR2. CDBYR2 is not used in frame image capture, one-field image capture, or data fetch.

The address specified by this register must be in 32-bit units. As the setting is in 4-pixel units, the lower two bits are always fixed to 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CBYR2[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CBYR2[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	CBYR2[31:2]	H'0000 0000	R/W	These bits set the address for storing the Y component data of the captured bottom-field data (4-pixel units).
1, 0	CBYR2[1:0]		R	

In this register, set the address of the starting point of the memory area where the Y component of the captured bottom-field image is to be stored in a bundle write, as shown in Figure 36.54.

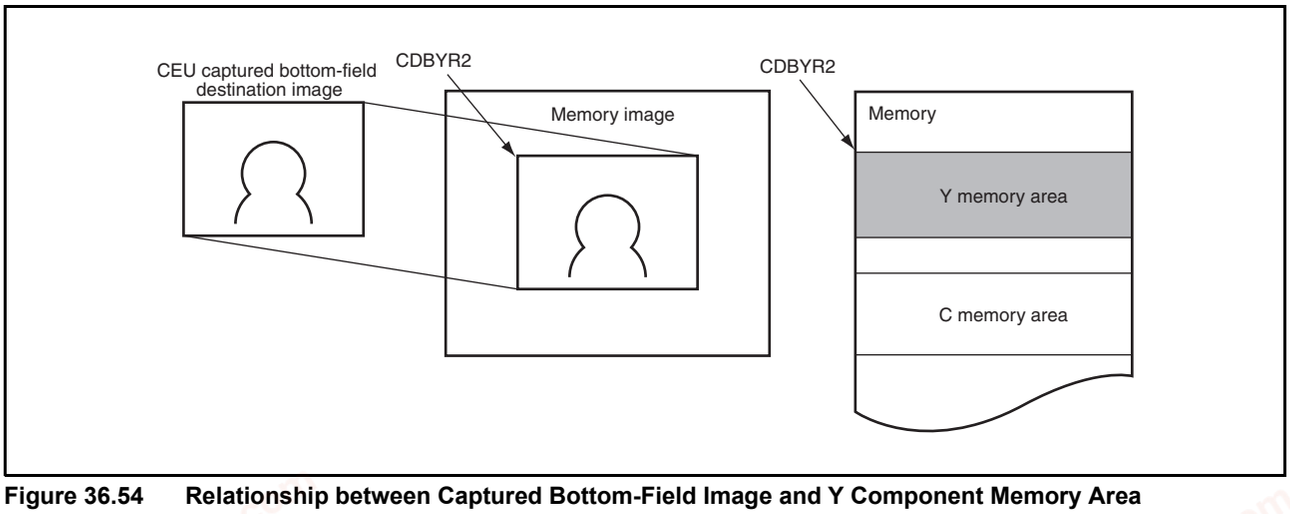


Figure 36.54 Relationship between Captured Bottom-Field Image and Y Component Memory Area

chanhvinh.com

36.4.28 Capture Data Bottom-Field Address C Register 2 (CDBCR2)

CDBCR2 specifies the address for the chrominance (C) component of the bottom field used in a bundle write. CDBCR2 is used only in a bundle write.

CDBCR2 specifies the address where the C component of the captured bottom-field data is to be stored in both-field image capture. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. Set the start address of the memory area where the C component of the bottom-field image captured in both-field image capture is to be stored by CDBCR2. CDBCR2 is not used in frame image capture, one-field image capture, or data fetch.

The address specified by this register must be in 32-bit units. As the setting is in 4-pixel units, the lower two bits are always fixed to 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CBCR2[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CBCR2[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	CBCR2[31:2]	H'0000 0000	R/W	These bits set the address for storing the C component data of the captured bottom-field data (4-pixel units).
1, 0	CBCR2[1:0]		R	

In this register, set the address of the starting point of the memory area where the C component of the captured bottom-field image is to be stored in a bundle write, as shown in Figure 36.55. The C component has an output data format as shown in Figure 36.56, and is saved in the memory in this format.

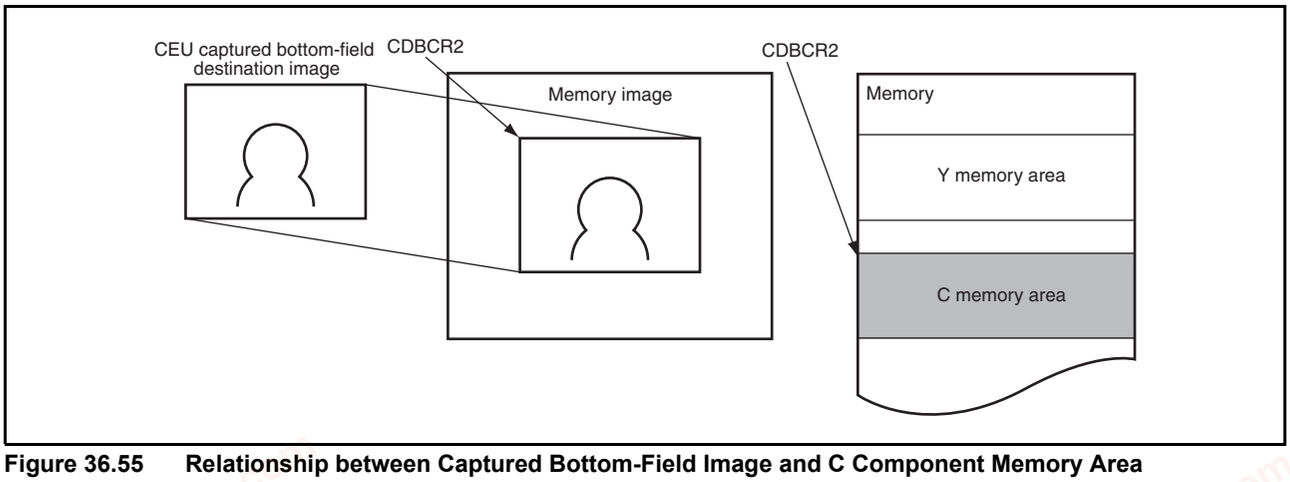


Figure 36.55 Relationship between Captured Bottom-Field Image and C Component Memory Area

chanhvinh.com

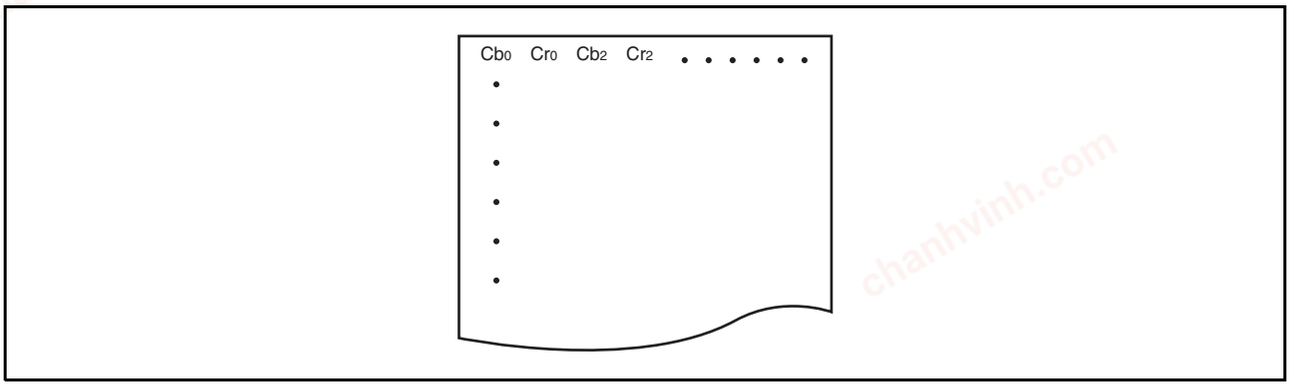


Figure 36.56 Image of Storing C Components in Memory

chanhvinh.com

36.5 Usage Notes for CEU

36.5.1 Conditions for Connection to an External Module

(1) Clock Frequency

The external input clock should have a frequency at most the same as the CEU operating clock frequency (Bφ), with jitter on both sides included.

CEU operating clock frequency (Bφ) ≥ External input clock frequency

(2) Blanking Period

The period from the last valid pixel in each line to the next horizontal sync signal HD must be at least 20 cycles.

(3) Fixed Period of Field Identification Signal

The field identification signal FLD should be fixed for at least 1-HD period since a VD input.

chanhvinh.com

36.5.2 Restrictions on Input/Output Functions

Table 36.11 lists the restrictions regarding the CEU input/output functions.

Table 36.11 Restrictions on CEU Input/Output Functions

Item	Restrictions
External module interface	The operating clock of the external module (VIO_CLK) should always have a frequency at most the same as that of the CEU operating clock (Bφ), with jitter on both sides included.
	Selecting the interface, or modifying the frequency of the external module operating clock or HD/VD polarity must be done when capture operations are halted for sure.
	Specify the capture horizontal size in image capture in 8-cycle units.
	Specify the capture horizontal size in data fetch in 4-cycle units.
	The capture vertical size must be specified in 4-line units.
	The maximum number of cycles in the horizontal sync signal period should be 16,375 cycles of external input clock.
	The maximum number of lines (HD count) in the vertical sync signal should be 16,382 lines.
	The minimum number of captured pixels should be sub-QCIF (128 × 96).
	The maximum number of captured pixels should be 5 megapixels (2,560 × 1,920).
	Captured size in data enable fetch Maximum: 6 Mbytes (2,048 × 1,536 × 2) Minimum: 16 bytes
Memory output	The output address must be specified in 32-bit units.
	The horizontal size of the destination image (memory) must be specified in 4-pixel units.
	The number of horizontal output pixels (= horizontal clipping size) must be specified in 4-pixel units.
	The number of vertical output lines (HD) (= vertical clipping size) must be specified in 4-line (HD) units.
	In data enable fetch bundle write, the output address must be specified in 32-byte units.
Internal processing	The filter clipping size must be specified as a value equal to or lower than the actual output size of the filter.

36.5.3 Cooperation with Video Display Controller 5

Since the input data are written to the memory separately as Y data and CbCr data in image capture mode, the captured data cannot be displayed in the video display controller 5.

36.5.4 Software Reset

For transitions to the software reset state by the CPKIL bit in the CAPSR register, see section 42.3.6, Software Reset. However, where the procedure refers to the SRST bit, read this as the CPKIL bit in the CAPSR register.

chanhvinh.com

37. SCUX

The SCUX consists of sampling converters, digital volume units, and a mixer. It is connected to the SSIF module to have an interface with external modules.

37.1 Features

(1) [SRC] Sampling Rate Conversion

- Asynchronous or synchronous sampling rate conversion is possible*1
- Sampling rate (synchronous mode)*2
  Input [KHz]: 8, 11.025, 12, 16, 22.05, 24, 32, 44.1, 48, 64, 88.2, or 96 is selectable
  Output [KHz]: 8, 16, 24, 32, 44.1, 48, or 96 is selectable
- Sampling rate (asynchronous mode)*2
  Input/output [KHz]: 1 to 96
- Supported bit sizes are 16 bits and 24 bits
- Sound quality: −132 dB or less*3
- 1, 2, 4, 6, or 8 channels are supported*4
- DMA transfer with on-chip memory or external memory and direct transfer with the SSIF module are possible

Note 1. The synchronous mode can be selected only when the SCUX is connected to the FFD and FFU modules. For details, refer to section 37.4.5.
Note 2. The selectable sampling rates depend on the number of used channels and rate ratio.
For details, refer to section 37.3.22 and section 37.4.7.
Note 3. The data format is a 24-bit value.
Note 4. The number of selectable channels depends on the sampling rate and route.
For details, refer to section 37.4.7.

(2) [DVU] Digital Volume and Mute Functions

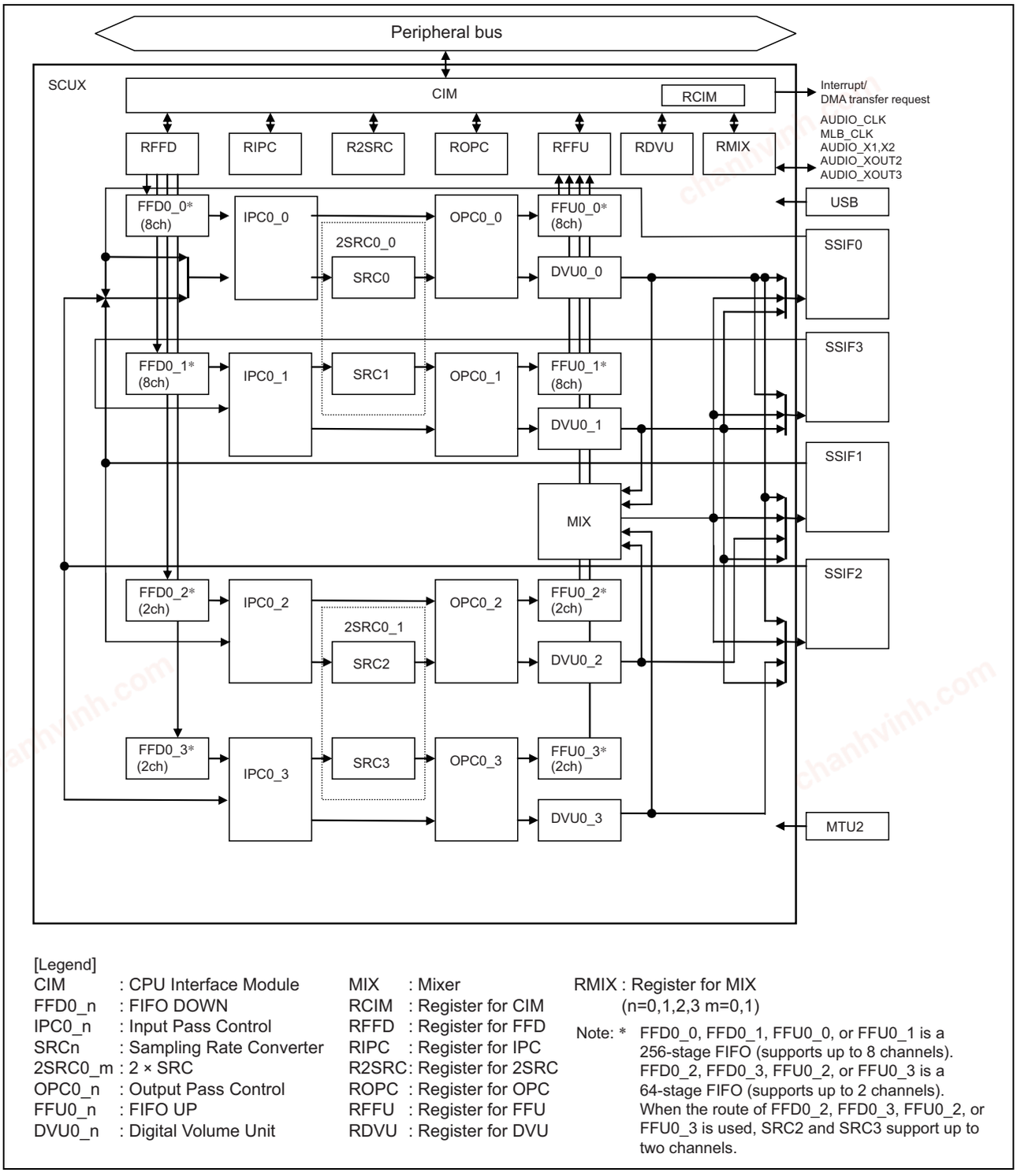
- Digital volume, volume ramp, and zero cross mute are provided as functions to adjust the volume
- The digital volume is set as a 24-bit fixed-point value within the range from a multiple of 0 to 8 (mute, −120 to 18 dB)
- Volume ramp can be used to perform soft mute, fade in, fade out, and volume change as desired
- The ramp time of volume ramp can be changed and set within the sampling range of 2^0 to 2^{23}
- Zero cross mute turns the sound mute at the zero cross point of audio data
- Direct transfer with the SSIF module and transfer with the mixer are possible

(3) [MIX] Mixer

- Data of two to four source systems can be mixed (added together) into one system
- The ratio to add the sources can be set
- The ratio can be changed dynamically
- Volume ramp enables mixing to be performed (ramp time is variable)
- Only direct transfer with the SSIF module is possible

chanhvinh.com

Figure 37.1 shows a block diagram of the SCUX.



[Legend]

CIM	: CPU Interface Module	MIX	: Mixer	RMIX	: Register for MIX (n=0,1,2,3 m=0,1)
FFD0_n	: FIFO DOWN	RCIM	: Register for CIM		
IPC0_n	: Input Pass Control	RFFD	: Register for FFD		
SRCn	: Sampling Rate Converter	RIPC	: Register for IPC		
2SRC0_m	: 2 × SRC	R2SRC	: Register for 2SRC		
OPC0_n	: Output Pass Control	ROPC	: Register for OPC		
FFU0_n	: FIFO UP	RFFU	: Register for FFU		
DVU0_n	: Digital Volume Unit	RDVU	: Register for DVU		

Note: * FFD0_0, FFD0_1, FFU0_0, or FFU0_1 is a 256-stage FIFO (supports up to 8 channels). FFD0_2, FFD0_3, FFU0_2, or FFU0_3 is a 64-stage FIFO (supports up to 2 channels). When the route of FFD0_2, FFD0_3, FFU0_2, or FFU0_3 is used, SRC2 and SRC3 support up to two channels.

Figure 37.1 Block Diagram of SCUX

chanhvinh.com

37.2 Input/Output Pins

Table 37.1 Pin Configuration

Name	I/O	Function
AUDIO_CLK	Input	External clock for audio Can be used in generating the sampling timing.
MLB_CLK*1	Input	External clock for MLB Can be used in generating the sampling timing.
AUDIO_X1	Input	Crystal resonator/external clock for audio Can be used in generating the sampling timing.
AUDIO_X2	Output	
AUDIO_XOUT2	Output	Output clock of AUDIO_X1 divided into two (Output is toggled once per cycle of the AUDIO_X1 clock.)
AUDIO_XOUT3	Output	Output clock of AUDIO_X1 divided into three (Output is toggled once every 1.5 cycles of the AUDIO_X1 clock.)

Note 1. Since the RZ/A1LU and RZ/A1LC do not have a MLB_CLK external pin, a fixed value (0) is input as the signal in this module.

chanhvinh.com

37.3 Register Descriptions

Table 37.2 shows the register configuration.

Table 37.2 Register Configuration

Block Name	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
IPC0_0	IPC0_0 initialization register	IPCIR_IPC0_0	R/W	H'00000001	H'E8208000	32
	IPC0_0 pass select register	IPSLR_IPC0_0	R/W	H'00000000	H'E8208004	32
IPC0_1	IPC0_1 initialization register	IPCIR_IPC0_1	R/W	H'00000001	H'E8208100	32
	IPC0_1 pass select register	IPSLR_IPC0_1	R/W	H'00000000	H'E8208104	32
IPC0_2	IPC0_2 initialization register	IPCIR_IPC0_2	R/W	H'00000001	H'E8208200	32
	IPC0_2 pass select register	IPSLR_IPC0_2	R/W	H'00000000	H'E8208204	32
IPC0_3	IPC0_3 initialization register	IPCIR_IPC0_3	R/W	H'00000001	H'E8208300	32
	IPC0_3 pass select register	IPSLR_IPC0_3	R/W	H'00000000	H'E8208304	32
OPC0_0	OPC0_0 initialization register	OPCIR_OPC0_0	R/W	H'00000001	H'E8208400	32
	OPC0_0 pass select register	OPSLR_OPC0_0	R/W	H'00000000	H'E8208404	32
OPC0_1	OPC0_1 initialization register	OPCIR_OPC0_1	R/W	H'00000001	H'E8208500	32
	OPC0_1 pass select register	OPSLR_OPC0_1	R/W	H'00000000	H'E8208504	32
OPC0_2	OPC0_2 initialization register	OPCIR_OPC0_2	R/W	H'00000001	H'E8208600	32
	OPC0_2 pass select register	OPSLR_OPC0_2	R/W	H'00000000	H'E8208604	32
OPC0_3	OPC0_3 initialization register	OPCIR_OPC0_3	R/W	H'00000001	H'E8208700	32
	OPC0_3 pass select register	OPSLR_OPC0_3	R/W	H'00000000	H'E8208704	32
FFD0_0	FFD0_0 FIFO download initialization register	FFDIR_FFD0_0	R/W	H'00000001	H'E8208800	32
	FFD0_0 FIFO download audio information register	FDAIR_FFD0_0	R/W	H'00000000	H'E8208804	32
	FFD0_0 FIFO download request size register	DRQSR_FFD0_0	R/W	H'00000000	H'E8208808	32
	FFD0_0 FIFO download pass register	FFDPR_FFD0_0	R/W	H'00000000	H'E820880C	32
	FFD0_0 FIFO download boot register	FFDBR_FFD0_0	R/W	H'00000000	H'E8208810	32
	FFD0_0 FIFO download event mask register	DEVMR_FFD0_0	R/W	H'00000000	H'E8208814	32
	FFD0_0 FIFO download event clear register	DEVCR_FFD0_0	R/W*1	H'00000000	H'E820881C	32
FFD0_1	FFD0_1 FIFO download initialization register	FFDIR_FFD0_1	R/W	H'00000001	H'E8208900	32
	FFD0_1 FIFO download audio information register	FDAIR_FFD0_1	R/W	H'00000000	H'E8208904	32
	FFD0_1 FIFO download request size register	DRQSR_FFD0_1	R/W	H'00000000	H'E8208908	32
	FFD0_1 FIFO download pass register	FFDPR_FFD0_1	R/W	H'00000000	H'E820890C	32
	FFD0_1 FIFO download boot register	FFDBR_FFD0_1	R/W	H'00000000	H'E8208910	32
	FFD0_1 FIFO download event mask register	DEVMR_FFD0_1	R/W	H'00000000	H'E8208914	32
	FFD0_1 FIFO download event clear register	DEVCR_FFD0_1	R/W*1	H'00000000	H'E820891C	32
FFD0_2	FFD0_2 FIFO download initialization register	FFDIR_FFD0_2	R/W	H'00000001	H'E8208A00	32
	FFD0_2 FIFO download audio information register	FDAIR_FFD0_2	R/W	H'00000000	H'E8208A04	32
	FFD0_2 FIFO download request size register	DRQSR_FFD0_2	R/W	H'00000000	H'E8208A08	32
	FFD0_2 FIFO download pass register	FFDPR_FFD0_2	R/W	H'00000000	H'E8208A0C	32
	FFD0_2 FIFO download boot register	FFDBR_FFD0_2	R/W	H'00000000	H'E8208A10	32
	FFD0_2 FIFO download event mask register	DEVMR_FFD0_2	R/W	H'00000000	H'E8208A14	32
	FFD0_2 FIFO download event clear register	DEVCR_FFD0_2	R/W*1	H'00000000	H'E8208A1C	32

chanhvinh.com

Table 37.2 Register Configuration

Block Name	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
FFD0_3	FFD0_3 FIFO download initialization register	FFDIR_FFD0_3	R/W	H'00000001	H'E8208B00	32
	FFD0_3 FIFO download audio information register	FDAIR_FFD0_3	R/W	H'00000000	H'E8208B04	32
	FFD0_3 FIFO download request size register	DRQSR_FFD0_3	R/W	H'00000000	H'E8208B08	32
	FFD0_3 FIFO download pass register	FFDPR_FFD0_3	R/W	H'00000000	H'E8208B0C	32
	FFD0_3 FIFO download boot register	FFDBR_FFD0_3	R/W	H'00000000	H'E8208B10	32
	FFD0_3 FIFO download event mask register	DEVMR_FFD0_3	R/W	H'00000000	H'E8208B14	32
	FFD0_3 FIFO download event clear register	DEVCR_FFD0_3	R/W*1	H'00000000	H'E8208B1C	32
FFU0_0	FFU0_0 FIFO upload initialization register	FFUIR_FFU0_0	R/W	H'00000001	H'E8208C00	32
	FFU0_0 FIFO upload audio information register	FUAIR_FFU0_0	R/W	H'00000000	H'E8208C04	32
	FFU0_0 FIFO upload request size register	URQSR_FFU0_0	R/W	H'00000000	H'E8208C08	32
	FFU0_0 FIFO upload pass register	FFUPR_FFU0_0	R/W	H'00000000	H'E8208C0C	32
	FFU0_0 FIFO upload event mask register	UEVMR_FFU0_0	R/W	H'00000000	H'E8208C10	32
	FFU0_0 FIFO upload event clear register	UEVCR_FFU0_0	R/W*1	H'00000000	H'E8208C18	32
FFU0_1	FFU0_1 FIFO upload initialization register	FFUIR_FFU0_1	R/W	H'00000001	H'E8208D00	32
	FFU0_1 FIFO upload audio information register	FUAIR_FFU0_1	R/W	H'00000000	H'E8208D04	32
	FFU0_1 FIFO upload request size register	URQSR_FFU0_1	R/W	H'00000000	H'E8208D08	32
	FFU0_1 FIFO upload pass register	FFUPR_FFU0_1	R/W	H'00000000	H'E8208D0C	32
	FFU0_1 FIFO upload event mask register	UEVMR_FFU0_1	R/W	H'00000000	H'E8208D10	32
	FFU0_1 FIFO upload event clear register	UEVCR_FFU0_1	R/W*1	H'00000000	H'E8208D18	32
FFU0_2	FFU0_2 FIFO upload initialization register	FFUIR_FFU0_2	R/W	H'00000001	H'E8208E00	32
	FFU0_2 FIFO upload audio information register	FUAIR_FFU0_2	R/W	H'00000000	H'E8208E04	32
	FFU0_2 FIFO upload request size register	URQSR_FFU0_2	R/W	H'00000000	H'E8208E08	32
	FFU0_2 FIFO upload pass register	FFUPR_FFU0_2	R/W	H'00000000	H'E8208E0C	32
	FFU0_2 FIFO upload event mask register	UEVMR_FFU0_2	R/W	H'00000000	H'E8208E10	32
	FFU0_2 FIFO upload event clear register	UEVCR_FFU0_2	R/W*1	H'00000000	H'E8208E18	32
FFU0_3	FFU0_3 FIFO upload initialization register	FFUIR_FFU0_3	R/W	H'00000001	H'E8208F00	32
	FFU0_3 FIFO upload audio information register	FUAIR_FFU0_3	R/W	H'00000000	H'E8208F04	32
	FFU0_3 FIFO upload request size register	URQSR_FFU0_3	R/W	H'00000000	H'E8208F08	32
	FFU0_3 FIFO upload pass register	FFUPR_FFU0_3	R/W	H'00000000	H'E8208F0C	32
	FFU0_3 FIFO upload event mask register	UEVMR_FFU0_3	R/W	H'00000000	H'E8208F10	32
	FFU0_3 FIFO upload event clear register	UEVCR_FFU0_3	R/W*1	H'00000000	H'E8208F18	32

chanhvinh.com

Table 37.2 Register Configuration

Block Name	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
2SRC0_0 (SRC0)	2SRC0_0 initialization register 0	SRCIR0_2SRC0_0	R/W	H'00000001	H'E8209000	32
	2SRC0_0 audio information register 0	SADIR0_2SRC0_0	R/W	H'00000000	H'E8209004	32
	2SRC0_0 bypass register 0	SRCBR0_2SRC0_0	R/W	H'00000000	H'E8209008	32
	2SRC0_0 IFS control register 0	IFSCR0_2SRC0_0	R/W	H'00000000	H'E820900C	32
	2SRC0_0 IFS value setting register 0	IFSVR0_2SRC0_0	R/W	H'00000000	H'E8209010	32
	2SRC0_0 control register 0	SRCCR0_2SRC0_0	R/W	H'00000000	H'E8209014	32
	2SRC0_0 minimum FS setting register 0	MNFSR0_2SRC0_0	R/W	H'00000000	H'E8209018	32
	2SRC0_0 buffer size setting register 0	BFSSR0_2SRC0_0	R/W	H'00000000	H'E820901C	32
	2SRC0_0 SCU2 status register 0	SC2SR0_2SRC0_0	R	H'00000000	H'E8209020	32
	2SRC0_0 wait time setting register 0	WATSR0_2SRC0_0	R/W	H'00000000	H'E8209024	32
	2SRC0_0 event mask register 0	SEVMR0_2SRC0_0	R/W	H'00000000	H'E8209028	32
	2SRC0_0 event clear register 0	SEVCR0_2SRC0_0	R/W*1	H'00000000	H'E8209030	32
2SRC0_0 (SRC1)	2SRC0_0 initialization register 1	SRCIR1_2SRC0_0	R/W	H'00000001	H'E8209034	32
	2SRC0_0 audio information register 1	SADIR1_2SRC0_0	R/W	H'00000000	H'E8209038	32
	2SRC0_0 bypass register 1	SRCBR1_2SRC0_0	R/W	H'00000000	H'E820903C	32
	2SRC0_0 IFS control register 1	IFSCR1_2SRC0_0	R/W	H'00000000	H'E8209040	32
	2SRC0_0 IFS value setting register 1	IFSVR1_2SRC0_0	R/W	H'00000000	H'E8209044	32
	2SRC0_0 control register 1	SRCCR1_2SRC0_0	R/W	H'00000000	H'E8209048	32
	2SRC0_0 minimum FS setting register 1	MNFSR1_2SRC0_0	R/W	H'00000000	H'E820904C	32
	2SRC0_0 buffer size setting register 1	BFSSR1_2SRC0_0	R/W	H'00000000	H'E8209050	32
	2SRC0_0 SCU2 status register 1	SC2SR1_2SRC0_0	R	H'00000000	H'E8209054	32
	2SRC0_0 wait time setting register 1	WATSR1_2SRC0_0	R/W	H'00000000	H'E8209058	32
	2SRC0_0 event mask register 1	SEVMR1_2SRC0_0	R/W	H'00000000	H'E820905C	32
	2SRC0_0 event clear register 1	SEVCR1_2SRC0_0	R/W*1	H'00000000	H'E8209064	32
	2SRC0_0 initialization register RIF	SRCIRR_2SRC0_0	R/W	H'00000001	H'E8209068	32
2SRC0_1 (SRC2)	2SRC0_1 initialization register 0	SRCIR0_2SRC0_1	R/W	H'00000001	H'E8209100	32
	2SRC0_1 audio information register 0	SADIR0_2SRC0_1	R/W	H'00000000	H'E8209104	32
	2SRC0_1 bypass register 0	SRCBR0_2SRC0_1	R/W	H'00000000	H'E8209108	32
	2SRC0_1 IFS control register 0	IFSCR0_2SRC0_1	R/W	H'00000000	H'E820910C	32
	2SRC0_1 IFS value setting register 0	IFSVR0_2SRC0_1	R/W	H'00000000	H'E8209110	32
	2SRC0_1 control register 0	SRCCR0_2SRC0_1	R/W	H'00000000	H'E8209114	32
	2SRC0_1 minimum FS setting register 0	MNFSR0_2SRC0_1	R/W	H'00000000	H'E8209118	32
	2SRC0_1 buffer size setting register 0	BFSSR0_2SRC0_1	R/W	H'00000000	H'E820911C	32
	2SRC0_1 SCU2 status register 0	SC2SR0_2SRC0_1	R	H'00000000	H'E8209120	32
	2SRC0_1 wait time setting register 0	WATSR0_2SRC0_1	R/W	H'00000000	H'E8209124	32
	2SRC0_1 event mask register 0	SEVMR0_2SRC0_1	R/W	H'00000000	H'E8209128	32
	2SRC0_1 event clear register 0	SEVCR0_2SRC0_1	R/W*1	H'00000000	H'E8209130	32

chanhvinh.com

Table 37.2 Register Configuration

Block Name	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
2SRC0_1 (SRC3)	2SRC0_1 initialization register 1	SRCIR1_2SRC0_1	R/W	H'00000001	H'E8209134	32
	2SRC0_1 audio information register 1	SADIR1_2SRC0_1	R/W	H'00000000	H'E8209138	32
	2SRC0_1 bypass register 1	SRCBR1_2SRC0_1	R/W	H'00000000	H'E820913C	32
	2SRC0_1 IFS control register 1	IFSCR1_2SRC0_1	R/W	H'00000000	H'E8209140	32
	2SRC0_1 IFS value setting register 1	IFSVR1_2SRC0_1	R/W	H'00000000	H'E8209144	32
	2SRC0_1 control register 1	SRCCR1_2SRC0_1	R/W	H'00000000	H'E8209148	32
	2SRC0_1 minimum FS setting register 1	MNFSR1_2SRC0_1	R/W	H'00000000	H'E820914C	32
	2SRC0_1 buffer size setting register 1	BFSSR1_2SRC0_1	R/W	H'00000000	H'E8209150	32
	2SRC0_1 SCU2 status register 1	SC2SR1_2SRC0_1	R	H'00000000	H'E8209154	32
	2SRC0_1 wait time setting register 1	WATSR1_2SRC0_1	R/W	H'00000000	H'E8209158	32
	2SRC0_1 event mask register 1	SEVMR1_2SRC0_1	R/W	H'00000000	H'E820915C	32
	2SRC0_1 event clear register 1	SEVCR1_2SRC0_1	R/W*1	H'00000000	H'E8209164	32
	2SRC0_1 initialization register RIF	SRCIRR_2SRC0_1	R/W	H'00000001	H'E8209168	32
DVU0_0	DVU0_0 initialization register	DVUIR_DVU0_0	R/W	H'00000001	H'E8209200	32
	DVU0_0 audio information register	VADIR_DVU0_0	R/W	H'00000000	H'E8209204	32
	DVU0_0 bypass register	DVUBR_DVU0_0	R/W	H'00000000	H'E8209208	32
	DVU0_0 control register	DVUCR_DVU0_0	R/W	H'00000000	H'E820920C	32
	DVU0_0 zero cross mute control register	ZCMCR_DVU0_0	R/W	H'00000000	H'E8209210	32
	DVU0_0 volume ramp control register	VRCTR_DVU0_0	R/W	H'00000000	H'E8209214	32
	DVU0_0 volume ramp period register	VRPDR_DVU0_0	R/W	H'00000000	H'E8209218	32
	DVU0_0 volume ramp decibel register	VRDBR_DVU0_0	R/W	H'00000000	H'E820921C	32
	DVU0_0 volume ramp wait time register	VRWTR_DVU0_0	R/W	H'00000000	H'E8209220	32
	DVU0_0 volume value setting 0 register	VOL0R_DVU0_0	R/W	H'00000000	H'E8209224	32
	DVU0_0 volume value setting 1 register	VOL1R_DVU0_0	R/W	H'00000000	H'E8209228	32
	DVU0_0 volume value setting 2 register	VOL2R_DVU0_0	R/W	H'00000000	H'E820922C	32
	DVU0_0 volume value setting 3 register	VOL3R_DVU0_0	R/W	H'00000000	H'E8209230	32
	DVU0_0 volume value setting 4 register	VOL4R_DVU0_0	R/W	H'00000000	H'E8209234	32
	DVU0_0 volume value setting 5 register	VOL5R_DVU0_0	R/W	H'00000000	H'E8209238	32
	DVU0_0 volume value setting 6 register	VOL6R_DVU0_0	R/W	H'00000000	H'E820923C	32
	DVU0_0 volume value setting 7 register	VOL7R_DVU0_0	R/W	H'00000000	H'E8209240	32
	DVU0_0 enable register	DVUER_DVU0_0	R/W	H'00000000	H'E8209244	32
	DVU0_0 status register	DVUSR_DVU0_0	R	H'00000000	H'E8209248	32
	DVU0_0 event mask register	VEVMR_DVU0_0	R/W	H'00000000	H'E820924C	32
	DVU0_0 event clear register	VEVCR_DVU0_0	R/W*1	H'00000000	H'E8209254	32

chanhvinh.com

Table 37.2 Register Configuration

Block Name	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
DVU0_1	DVU0_1 initialization register	DVUIR_DVU0_1	R/W	H'00000001	H'E8209300	32
	DVU0_1 audio information register	VADIR_DVU0_1	R/W	H'00000000	H'E8209304	32
	DVU0_1 bypass register	DVUBR_DVU0_1	R/W	H'00000000	H'E8209308	32
	DVU0_1 control register	DVUCR_DVU0_1	R/W	H'00000000	H'E820930C	32
	DVU0_1 zero cross mute control register	ZCMCR_DVU0_1	R/W	H'00000000	H'E8209310	32
	DVU0_1 volume ramp control register	VRCTR_DVU0_1	R/W	H'00000000	H'E8209314	32
	DVU0_1 volume ramp period register	VRPDR_DVU0_1	R/W	H'00000000	H'E8209318	32
	DVU0_1 volume ramp decibel register	VRDBR_DVU0_1	R/W	H'00000000	H'E820931C	32
	DVU0_1 volume ramp wait time register	VRWTR_DVU0_1	R/W	H'00000000	H'E8209320	32
	DVU0_1 volume value setting 0 register	VOL0R_DVU0_1	R/W	H'00000000	H'E8209324	32
	DVU0_1 volume value setting 1 register	VOL1R_DVU0_1	R/W	H'00000000	H'E8209328	32
	DVU0_1 volume value setting 2 register	VOL2R_DVU0_1	R/W	H'00000000	H'E820932C	32
	DVU0_1 volume value setting 3 register	VOL3R_DVU0_1	R/W	H'00000000	H'E8209330	32
	DVU0_1 volume value setting 4 register	VOL4R_DVU0_1	R/W	H'00000000	H'E8209334	32
	DVU0_1 volume value setting 5 register	VOL5R_DVU0_1	R/W	H'00000000	H'E8209338	32
	DVU0_1 volume value setting 6 register	VOL6R_DVU0_1	R/W	H'00000000	H'E820933C	32
	DVU0_1 volume value setting 7 register	VOL7R_DVU0_1	R/W	H'00000000	H'E8209340	32
	DVU0_1 enable register	DVUER_DVU0_1	R/W	H'00000000	H'E8209344	32
	DVU0_1 status register	DVUSR_DVU0_1	R	H'00000000	H'E8209348	32
	DVU0_1 event mask register	VEVMR_DVU0_1	R/W	H'00000000	H'E820934C	32
	DVU0_1 event clear register	VEVCR_DVU0_1	R/W*1	H'00000000	H'E8209354	32
DVU0_2	DVU0_2 initialization register	DVUIR_DVU0_2	R/W	H'00000001	H'E8209400	32
	DVU0_2 audio information register	VADIR_DVU0_2	R/W	H'00000000	H'E8209404	32
	DVU0_2 bypass register	DVUBR_DVU0_2	R/W	H'00000000	H'E8209408	32
	DVU0_2 control register	DVUCR_DVU0_2	R/W	H'00000000	H'E820940C	32
	DVU0_2 zero cross mute control register	ZCMCR_DVU0_2	R/W	H'00000000	H'E8209410	32
	DVU0_2 volume ramp control register	VRCTR_DVU0_2	R/W	H'00000000	H'E8209414	32
	DVU0_2 volume ramp period register	VRPDR_DVU0_2	R/W	H'00000000	H'E8209418	32
	DVU0_2 volume ramp decibel register	VRDBR_DVU0_2	R/W	H'00000000	H'E820941C	32
	DVU0_2 volume ramp wait time register	VRWTR_DVU0_2	R/W	H'00000000	H'E8209420	32
	DVU0_2 volume value setting 0 register	VOL0R_DVU0_2	R/W	H'00000000	H'E8209424	32
	DVU0_2 volume value setting 1 register	VOL1R_DVU0_2	R/W	H'00000000	H'E8209428	32
	DVU0_2 volume value setting 2 register	VOL2R_DVU0_2	R/W	H'00000000	H'E820942C	32
	DVU0_2 volume value setting 3 register	VOL3R_DVU0_2	R/W	H'00000000	H'E8209430	32
	DVU0_2 volume value setting 4 register	VOL4R_DVU0_2	R/W	H'00000000	H'E8209434	32
	DVU0_2 volume value setting 5 register	VOL5R_DVU0_2	R/W	H'00000000	H'E8209438	32
	DVU0_2 volume value setting 6 register	VOL6R_DVU0_2	R/W	H'00000000	H'E820943C	32
	DVU0_2 volume value setting 7 register	VOL7R_DVU0_2	R/W	H'00000000	H'E8209440	32
	DVU0_2 enable register	DVUER_DVU0_2	R/W	H'00000000	H'E8209444	32
	DVU0_2 status register	DVUSR_DVU0_2	R	H'00000000	H'E8209448	32
	DVU0_2 event mask register	VEVMR_DVU0_2	R/W	H'00000000	H'E820944C	32
	DVU0_2 event clear register	VEVCR_DVU0_2	R/W*1	H'00000000	H'E8209454	32

chanhvinh.com

Table 37.2 Register Configuration

Block Name	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
DVU0_3	DVU0_3 initialization register	DVUIR_DVU0_3	R/W	H'00000001	H'E8209500	32
	DVU0_3 audio information register	VADIR_DVU0_3	R/W	H'00000000	H'E8209504	32
	DVU0_3 bypass register	DVUBR_DVU0_3	R/W	H'00000000	H'E8209508	32
	DVU0_3 control register	DVUCR_DVU0_3	R/W	H'00000000	H'E820950C	32
	DVU0_3 zero cross mute control register	ZCMCR_DVU0_3	R/W	H'00000000	H'E8209510	32
	DVU0_3 volume ramp control register	VRCTR_DVU0_3	R/W	H'00000000	H'E8209514	32
	DVU0_3 volume ramp period register	VRPDR_DVU0_3	R/W	H'00000000	H'E8209518	32
	DVU0_3 volume ramp decibel register	VRDBR_DVU0_3	R/W	H'00000000	H'E820951C	32
	DVU0_3 volume ramp wait time register	VRWTR_DVU0_3	R/W	H'00000000	H'E8209520	32
	DVU0_3 volume value setting 0 register	VOL0R_DVU0_3	R/W	H'00000000	H'E8209524	32
	DVU0_3 volume value setting 1 register	VOL1R_DVU0_3	R/W	H'00000000	H'E8209528	32
	DVU0_3 volume value setting 2 register	VOL2R_DVU0_3	R/W	H'00000000	H'E820952C	32
	DVU0_3 volume value setting 3 register	VOL3R_DVU0_3	R/W	H'00000000	H'E8209530	32
	DVU0_3 volume value setting 4 register	VOL4R_DVU0_3	R/W	H'00000000	H'E8209534	32
	DVU0_3 volume value setting 5 register	VOL5R_DVU0_3	R/W	H'00000000	H'E8209538	32
	DVU0_3 volume value setting 6 register	VOL6R_DVU0_3	R/W	H'00000000	H'E820953C	32
	DVU0_3 volume value setting 7 register	VOL7R_DVU0_3	R/W	H'00000000	H'E8209540	32
	DVU0_3 enable register	DVUER_DVU0_3	R/W	H'00000000	H'E8209544	32
	DVU0_3 status register	DVUSR_DVU0_3	R	H'00000000	H'E8209548	32
	DVU0_3 event mask register	VEVMR_DVU0_3	R/W	H'00000000	H'E820954C	32
	DVU0_3 event clear register	VEVCR_DVU0_3	R/W*1	H'00000000	H'E8209554	32
MIX	MIX0_0 initialization register	MIXIR_MIX0_0	R/W	H'00000001	H'E8209600	32
	MIX0_0 audio information register	MADIR_MIX0_0	R/W	H'00000000	H'E8209604	32
	MIX0_0 bypass register	MIXBR_MIX0_0	R/W	H'00000000	H'E8209608	32
	MIX0_0 mode register	MIXMR_MIX0_0	R/W	H'00000000	H'E820960C	32
	MIX0_0 volume period register	MVPDR_MIX0_0	R/W	H'00000000	H'E8209610	32
	MIX0_0 decibel A register	MDBAR_MIX0_0	R/W	H'00000000	H'E8209614	32
	MIX0_0 decibel B register	MDBBR_MIX0_0	R/W	H'00000000	H'E8209618	32
	MIX0_0 decibel C register	MDBCR_MIX0_0	R/W	H'00000000	H'E820961C	32
	MIX0_0 decibel D register	MDBDR_MIX0_0	R/W	H'00000000	H'E8209620	32
	MIX0_0 decibel enable register	MDBER_MIX0_0	R/W	H'00000000	H'E8209624	32
	MIX0_0 status register	MIXSR_MIX0_0	R	H'00000000	H'E8209628	32

chanhvinh.com

Table 37.2 Register Configuration

Block Name	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
CIM	Software reset register	SWRSR_CIM	R/W	H'00000001	H'E8209700	32
	DMA control register	DMACR_CIM	R/W	H'00000000	H'E8209704	32
	DMA transfer register for FFD0_0 RAM	DMATD0_CIM	W	—	H'E8209708	16, 32*2
	DMA transfer register for FFD0_1 RAM	DMATD1_CIM	W	—	H'E820970C	16, 32*3
	DMA transfer register for FFD0_2 RAM	DMATD2_CIM	W	—	H'E8209710	16, 32*4
	DMA transfer register for FFD0_3 RAM	DMATD3_CIM	W	—	H'E8209714	16, 32*5
	DMA transfer register for FFU0_0 RAM	DMATU0_CIM	R	—	H'E8209718	16, 32*6
	DMA transfer register for FFU0_1 RAM	DMATU1_CIM	R	—	H'E820971C	16, 32*7
	DMA transfer register for FFU0_2 RAM	DMATU2_CIM	R	—	H'E8209720	16, 32*8
	DMA transfer register for FFU0_3 RAM	DMATU3_CIM	R	—	H'E8209724	16, 32*9
	SSI route select register	SSIRSEL_CIM	R/W	H'00000000	H'E8209738	32
	FFD0_0 timing select register	FDTSEL0_CIM	R/W	H'00000000	H'E820973C	32
	FFD0_1 timing select register	FDTSEL1_CIM	R/W	H'00000000	H'E8209740	32
	FFD0_2 timing select register	FDTSEL2_CIM	R/W	H'00000000	H'E8209744	32
	FFD0_3 timing select register	FDTSEL3_CIM	R/W	H'00000000	H'E8209748	32
	FFU0_0 timing select register	FUTSEL0_CIM	W	H'00000000	H'E820974C	32
	FFU0_1 timing select register	FUTSEL1_CIM	W	H'00000000	H'E8209750	32
	FFU0_2 timing select register	FUTSEL2_CIM	W	H'00000000	H'E8209754	32
	FFU0_3 timing select register	FUTSEL3_CIM	W	H'00000000	H'E8209758	32
	SSI pin mode register	SSIPMD_CIM	R/W	H'00000000	H'E820975C	32
	SSI control register	SSICTRL_CIM	W	H'00000000	H'E8209760	32
	SRC0 route select register	SRCRSEL0_CIM	R/W	H'76543210	H'E8209764	32
	SRC1 route select register	SRCRSEL1_CIM	R/W	H'76543210	H'E8209768	32
	SRC2 route select register	SRCRSEL2_CIM	R/W	H'76543210	H'E820976C	32
	SRC3 route select register	SRCRSEL3_CIM	R/W	H'76543210	H'E8209770	32
	MIX route select register	MIXRSEL_CIM	R/W	H'76543210	H'E8209774	32

Note 1. Only 0 can be written to clear the flag. Writing 0 is ignored.
Note 2. Address H'E8209708 can be accessed in only 16-bit or 32-bit units.
Note 3. Address H'E820970C can be accessed in only 16-bit or 32-bit units.
Note 4. Address H'E8209710 can be accessed in only 16-bit or 32-bit units.
Note 5. Address H'E8209714 can be accessed in only 16-bit or 32-bit units.
Note 6. Address H'E8209718 can be accessed in only 16-bit or 32-bit units.
Note 7. Address H'E820971C can be accessed in only 16-bit or 32-bit units.
Note 8. Address H'E8209720 can be accessed in only 16-bit or 32-bit units.
Note 9. Address H'E8209724 can be accessed in only 16-bit or 32-bit units.

chanhvinh.com

37.3.1 IPC0_n Initialization Register (IPCIR_IPC0_n) (n = 0, 1, 2, 3)

IPCIR_IPC0_n is a 32-bit readable/writable register that initializes the IPC internal circuit.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	INIT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	INIT	1	R/W	Initialization When this bit is set to 1, the IPC internal circuit is initialized. To cancel the initialization, set this bit to 0 with it set to 1. 0: Processing State 1: Initialization (sets the initial setting of other registers)

37.3.2 IPC0_n Pass Select Register (IPSLR_IPC0_n) (n = 0, 1, 2, 3)

IPSLR_IPC0_n is a 32-bit readable/writable register that selects the input source and output destination of the input timing signal and audio data.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	IPC_PASS_SEL		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	IPC_ PASS_SEL	0	R/W	IPC Pass Select These bits select the pass. 000: No operation 001: Input Timing Signal/Input Audio Data : EXTERNAL(SSIF) → IPC → SRC (Async mode) 010: Reserved 011: Input Timing Signal : EXTERNAL → IPC → FFD/SRC Input Audio Data : FFD → IPC → SRC (Async mode) 100: Input Timing Signal/Input Audio Data : FFD → IPC → SRC (Sync mode) 101 to 111: No operation

chanhvinh.com

37.3.3 OPC0_n Initialization Register (OPCIR_OPC0_n) (n = 0, 1, 2, 3)

OPCIR_OPC0_n is a 32-bit readable/writable register that initializes the OPC internal circuit.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	INIT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	INIT	1	R/W	Initialization When this bit is set to 1, the OPC internal circuit is initialized. To cancel the initialization, set this bit to 0 with it set to 1. 0: Processing State 1: Initialization (sets the initial setting of other registers)

37.3.4 OPC0_n Pass Select Register (OPSLR_OPC0_n) (n = 0, 1, 2, 3)

OPSLR_OPC0_n is a 32-bit readable/writable register that selects the input source and output destination of the output timing signal and audio data.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	OPC_PASS_SEL		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	OPC_PASS_SEL	0	R/W	OPC Pass Select These bits select the pass. 000: No operation 001: Output Timing Signal : EXTERNAL → OPC → SRC/DVU Output Audio Data : SRC(Async mode) → OPC → DVU 010: Reserved 011: Output Timing Signal : EXTERNAL → OPC → FFU/SRC Output Audio Data : SRC(Async mode) → OPC → FFU 100: Output Timing Signal : FFU → OPC → SRC Output Audio Data : SRC(Sync mode) → OPC → FFU 101 to 111: No operation

chanhvinh.com

37.3.5 FFD0_n FIFO Download Initialization Register (FFDIR_FFD0_n) (n = 0, 1, 2, 3)

FFDIR_FFD0_n is a 32-bit readable/writable register that initializes the FFD internal circuit.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	INIT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	INIT	1	R/W	Initialization When this bit is set to 1, the FFD internal circuit is initialized. To cancel the initialization, set this bit to 0 with it set to 1. 0: Processing State 1: Initialization (sets the initial setting of other registers)

37.3.6 FFD0_n FIFO Download Audio Information Register (FDAIR_FFD0_n) (n = 0, 1, 2, 3)

FDAIR_FFD0_n is a 32-bit readable/writable register that sets the number of FFD data channels.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	CHNUM			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	CHNUM	All 0	R/W	Number of Channels These bits set the number of channels. 0000: Zero (None) 0001: 1 channel 0010: 2 channels* 0011: Reserved 0100: 4 channels* 0101: Reserved 0110: 6 channels* 0111: Reserved 1000: 8 channels* 1001 to 1111: Reserved Note: * These bits are only settable for FFD0_0 and FFD0_1. Setting these bits in the registers for FFD0_2 and FFD0_3 is prohibited.

chanhvinh.com

37.3.7 FFD0_n FIFO Download Request Size Register (DRQSR_FFD0_n) (n = 0, 1, 2, 3)

DRQSR_FFD0_n is a 32-bit readable/writable register that sets the request size of data transfer requests.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	SIZE			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	SIZE	All 0	R/W	Request Size These bits set the PCM data size per request. The size must be less than the FIFO size. 0000: 256 data. (FFD0_0,FFD0_1), 64 data. (FFD0_2,FFD0_3) *1 0001: 128 data. (FFD0_0,FFD0_1), 32 data. (FFD0_2,FFD0_3) 0010: 64 data. (FFD0_0,FFD0_1), 16 data. (FFD0_2,FFD0_3) 0011: 32 data. (FFD0_0,FFD0_1), 8 data. (FFD0_2,FFD0_3) 0100: 16 data. (FFD0_0,FFD0_1), 4 data. (FFD0_2,FFD0_3) 0101: 8 data. (FFD0_0,FFD0_1), 2 data. (FFD0_2,FFD0_3) 0110: 4 data. (FFD0_0,FFD0_1), 1 data. (FFD0_2,FFD0_3) 0111: 2 data. (FFD0_0,FFD0_1)*2 1000: 1 data. (FFD0_0,FFD0_1)*2 1001 to 1111: Reserved. Notes: 1. If the request size equals the FIFO size, the request size must be a multiple of the number of channels so that the operation is correct. For example, the request size and FIFO size are 256, the number of channels can be set to 1, 2, 4, or 8. If the number of channels is not set to any of the numbers, FIFO overflow may occur. 2. These bits are only settable for FFD0_0 and FFD0_1. Setting these bits in the registers for FFD0_2 and FFD0_3 is prohibited.

chanhvinh.com

37.3.8 FFD0_n FIFO Download Pass Register (FFDPR_FFD0_n) (n = 0, 1, 2, 3)

FFDPR_FFD0_n is a 32-bit readable/writable register that sets the input/output route of audio data.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PASS	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PASS	All 0	R/W	Data Pass Select These bits set the data pass of audio data. 00: No pass selected 01: CIM → FFD → IPC (Async Mode) *1 10: CIM → FFD → IPC (Sync Mode) *2 11: Reserved Notes: 1.- Async mode: The FFD use external timing. 2.- Sync mode: The FFD use the request which is generated by SRC module. Set Sync mode in SRC.

37.3.9 FFD0_n FIFO Download Boot Register (FFDBR_FFD0_n) (n = 0, 1, 2, 3)

FFDBR_FFD0_n is a 32-bit readable/writable register that controls the starting and stopping of data requests sent to RFFD.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	BOOT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	BOOT	0	R/W	Data Request Start/Stop This bit controls the start/stop of data request. This bit can be set when the PASS bits in the FFDPR register are set to B'01 or B'10. This bit should be set after setting all FFD registers. 0: Stops the data request. 1: Starts the data request.

chanhvinh.com

37.3.10 FFD0_n FIFO Download Event Mask Register (DEVMR_FFD0_n) (n = 0, 1, 2, 3)

DEVMR_FFD0_n is a 32-bit readable/writable register that controls SCUFDIn interrupt requests.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEVMUF	DEVMOF	DEVMOL	DEVMIUF	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DEVMRQ	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	DEVMUF	0	R/W	FFD Underflow Mask This bit sets whether to mask the interrupt request from the DEVCUF bit in the DEVCR register. 0: Disables interrupts. 1: Enables interrupts.
30	DEVMOF	0	R/W	FFD Overflow Mask This bit sets whether to mask the interrupt request from the DEVCOF bit in the DEVCR register. 0: Disables interrupts. 1: Enables interrupts.
29	DEVMOL	0	R/W	FFD Overlap Mask This bit sets whether to mask the interrupt request from the DEVCOL bit in the DEVCR register. 0: Disables interrupts. 1: Enables interrupts.
28	DEVMIUF	0	R/W	FFD Initialization Underflow Mask This bit sets whether to mask the interrupt request from the DEVCIUF bit in the DEVCR register. 0: Disables interrupts. 1: Enables interrupts.
27 to 16	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
15	DEVMRQ	0	R/W	FFD Request Packet Mask This bit sets whether to mask the interrupt request from the DEVCRQ bit in the DEVCR register. 0: Disables interrupts. 1: Enables interrupts.
14 to 0	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.

chanhvinh.com

37.3.11 FFD0_n FIFO Download Event Clear Register (DEVCR_FFD0_n) (n = 0, 1, 2, 3)

DEVCR_FFD0_n is a 32-bit readable/writable register that clears SCUFDIn interrupt requests. When an interrupt event is generated, the relevant bit in this register is automatically set to 1 and 1 is retained until 0 is written to that bit.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEVCUF	DEVCOF	DEVCOL	DEVCIUF	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DEVCRQ	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	DEVCUF	0	R/(W)*1	FFD Underflow Clear This is an interrupt flag to indicate whether DPRAM underflow occurs. When it occurs and the first packet has already been written to DPRAM, this bit is set to 1. 0: Clears the flag. 1: Retains the flag.
30	DEVCOF	0	R/(W)*1	FFD Overflow Clear This is an interrupt flag to indicate whether DPRAM overflow occurs. When it occurs, this bit is set to 1. 0: Clears the flag. 1: Retains the flag.
29	DEVCOL	0	R/(W)*1	FFD Overlap Clear This is an interrupt flag to indicate whether FFD overlap occurs. When the next read request occurs while FFD reads data from DPRAM upon the previous read request, the overflow occurs. When it occurs, this bit is set to 1. 0: Clears the flag. 1: Waiting.
28	DEVCIUF	0	R/(W)*1	FFD Initialization Underflow Clear This is an interrupt flag to indicate whether DPRAM initialization underflow occurs. When it occurs and the first packet has not been written to DPRAM, this bit is set to 1. 0: Clears the flag. 1: Retains the flag.
27 to 16	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
15	DEVCRQ	0	R/(W)*1	FFD Request Packet Clear This is an interrupt flag to send a request to write one packet in FFD, to the CPU. When the request occurs, this bit is set to 1. 0: Clears the flag. 1: Retains the flag.
14 to 0	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.

Note 1. Readable/writable. When 0 is written, the bit is initialized. Writing 1 to the bit is ignored. Write 0 only to each bit corresponding to the interrupt source to be cleared; write 1 to other bits.

chanhvinh.com

37.3.12 FFU0_n FIFO Upload Initialization Register (FFUIR_FFU0_n) (n = 0, 1, 2, 3)

FFUIR_FFU0_n is a 32-bit readable/writable register that initializes the FFU internal circuit.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	INIT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	INIT	1	R/W	Initialization When this bit is set to 1, the FFU internal circuit is initialized. To cancel the initialization, set this bit to 0 with it set to 1. 0: Processing State 1: Initialization (sets the initial setting of other registers)

37.3.13 FFU0_n FIFO Upload Audio Information Register (FUAIR_FFU0_n) (n = 0, 1, 2, 3)

FUAIR_FFU0_n is a 32-bit readable/writable register that sets the number of FFU data channels.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	CHNUM			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	CHNUM	All 0	R/W	Number of Channels These bits set the number of channels. For details, see below. 0000: Zero (None) 0001: 1 channel 0010: 2 channels* 0011: Reserved 0100: 4 channels* 0101: Reserved 0110: 6 channels* 0111: Reserved 1000: 8 channels* 1001 to 1111: Reserved Note: *These bits are only settable for FFU0_0 and FFU0_1. Setting these bits in the registers for FFU0_2 and FFU0_3 is prohibited.

chanhvinh.com

37.3.14 FFU0_n FIFO Upload Request Size Register (URQSR_FFU0_n) (n = 0, 1, 2, 3)

URQSR_FFU0_n is a 32-bit readable/writable register that sets the request size of data transfer requests.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	SIZE			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	SIZE	All 0	R/W	Request Size These bits set the PCM data size per request. The size must be less than the FIFO size. 0000: 256 data. (FFU0_0,FFU0_1), 64 data. (FFU0_2,FFU0_3) *1 0001: 128 data. (FFU0_0,FFU0_1), 32 data. (FFU0_2,FFU0_3) 0010: 64 data. (FFU0_0,FFU0_1), 16 data. (FFU0_2,FFU0_3) 0011: 32 data. (FFU0_0,FFU0_1), 8 data. (FFU0_2,FFU0_3) 0100: 16 data. (FFU0_0,FFU0_1), 4 data. (FFU0_2,FFU0_3) 0101: 8 data. (FFU0_0,FFU0_1), 2 data. (FFU0_2,FFU0_3) 0110: 4 data. (FFU0_0,FFU0_1), 1 data. (FFU0_2,FFU0_3) 0111: 2 data. (FFU0_0,FFU0_1)*2 1000: 1 data. (FFU0_0,FFU0_1)*2 1001 to 1111: Reserved. Note 1. If the request size equals the FIFO size, the request size must be a multiple of the number of channels so that the operation is correct. For example, the request size and FIFO size are 256, the number of channels can be set to 1, 2, 4, or 8. If the number of channels is not set to any of the numbers, FIFO overflow may occur. Note 2. These bits are only settable for FFU0_0 and FFU0_1. Setting these bits in the registers for FFU0_2 and FFU0_3 is prohibited.

chanhvinh.com

37.3.15 FFU0_n FIFO Upload Pass Register (FFUPR_FFU0_n) (n = 0, 1, 2, 3)

FFUPR_FFU0_n is a 32-bit readable/writable register that sets the input/output route of audio data.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PASS	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PASS	All 0	R/W	Data Pass Select These bits set the data pass of audio data. 00: No pass selected 01: CIM ← FFU ← OPC (Async Mode) *1 10: CIM ← FFU ← OPC (Sync Mode) *2 11: Reserved Notes: 1. Async mode: The FFU use external timing. 2. Sync mode: The FFU use the request which is generated by SRC module. Set Sync mode in SRC.

chanhvinh.com

37.3.16 FFU0_n FIFO Upload Event Mask Register (UEVMR_FFU0_n) (n = 0, 1, 2, 3)

UEVMR_FFU0_n is a 32-bit readable/writable register that controls SCUFUIn interrupt requests.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	UEVMUF	UEVMOF	UEVMOL	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UEVMRQ	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	UEVMUF	0	R/W	FFU Underflow Mask This bit sets whether to mask the interrupt request from the UEVCUF bit in the UEVCR register. 0: Disables interrupts. 1: Enables interrupts.
30	UEVMOF	0	R/W	FFU Overflow Mask This bit sets whether to mask the interrupt request from the UEVCOF bit in the UEVCR register. 0: Disables interrupts. 1: Enables interrupts.
29	UEVMOL	0	R/W	FFU Overlap Mask This bit sets whether to mask the interrupt request from the UEVCOL bit in the UEVCR register. 0: Disables interrupts. 1: Enables interrupts.
28 to 16	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
15	UEVMRQ	0	R/W	FFU Request Packet Mask This bit sets whether to mask the interrupt request from the UEVCRQ bit in the UEVCR register. 0: Disables interrupts. 1: Enables interrupts.
14 to 0	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.

chanhvinh.com

37.3.17 FFU0_n FIFO Upload Event Clear Register (UEVCR_FFU0_n) (n = 0, 1, 2, 3)

UEVCR_FFU0_n is a 32-bit readable/writable register that clears SCUFUIn interrupt requests. When an interrupt event is generated, the relevant bit in this register is automatically set to 1 and 1 is retained until 0 is written to that bit.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	UEVCUF	UEVCOF	UEVCOL	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R/(W)*1	R/(W)*1	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UEVCRQ	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	UEVCUF	0	R/(W)*1	FFU Underflow Clear This is a flag to indicate whether DPRAM underflow occurs. When it occurs, this bit is set to 1. 0: Clears the flag. 1: Retains the flag.
30	UEVCOF	0	R/(W)*1	FFU Overflow Clear This is a flag to indicate whether DPRAM overflow occurs. When it occurs, this bit is set to 1. 0: Clears the flag. 1: Retains the flag.
29	UEVCOL	0	R/(W)*1	FFU Overlap Clear This is a flag to indicate whether DPRAM overlap occurs. When the next write request occurs while FFU writes data upon the previous write request, the overlap occurs. When it occurs, this bit is set to 1. 0: Clears the flag. 1: Retains the flag.
28 to 16	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
15	UEVCRQ	0	R/(W)*1	FFU Request Packet Clear This is an interrupt flag to send a request to read one packet from FFU, to the CPU. When the request occurs, this bit is set to 1. 0: Clears the flag. 1: Retains the flag.
14 to 0	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.

Note 1. Readable/writable. When 0 is written, the bit is initialized. Writing 1 to the bit is ignored. Write 0 only to each bit corresponding to the interrupt source to be cleared; write 1 to other bits.

chanhvinh.com

37.3.18 2SRC0_m Initialization Register p (SRCIRp_2SRC0_m) (m = 0, 1; p = 0, 1)

SRCIRp_2SRC0_m is a 32-bit readable/writable register that initializes the SRC internal circuit.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	INIT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	INIT	1	R/W	Initialization When this bit is set to 1, the SRC internal circuit is initialized. To cancel the initialization, set this bit to 0 with it set to 1. 0: Processing State 1: Initialization (sets the initial setting of other registers)

chanhvinh.com

37.3.19 2SRC0_m Audio Information Register p (SADIRp_2SRC0_m) (m = 0, 1; p = 0, 1)

SADIRp_2SRC0_m is a 32-bit readable/writable register that sets the bit length of data and the number of channels.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	OTBL				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	CHNUM			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20 to 16	OTBL	All 0	R/W	Bit Length of Output Audio Data 00000: 24 bits 00001 to 00111: Reserved 01000: 16 bits 01001 to 11111: Reserved
15 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	CHNUM	All 0	R/W	Number of Channels These bits set the number of channels. 0000: Zero (None) 0001: 1 channel 0010: 2 channels 0011: Reserved 0100: 4 channels 0101: Reserved 0110: 6 channels 0111: Reserved 1000: 8 channels 1001 to 1111: Reserved

chanhvinh.com

37.3.20 2SRC0_m Bypass Register p (SRCBRp_2SRC0_m) (m = 0, 1; p = 0, 1)

SRCBRp_2SRC0_m is a 32-bit readable/writable register that sets the bypass mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	BY PASS
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	BYPASS	0	R/W	Bypass Mode This bit controls the data pass of SRC. The bypass function is only used in asynchronous SRC mode. Therefore, this bit must be set to 0 when synchronous SRC mode is used (bit SRCMD0 in SRCCRp_2SRC0_m register is 1). 0: SRC function is used. Input data is processed by SRC and then the result data is connected to output data. 1: SRC function is not used. Input data is directly connected to output data.

37.3.21 2SRC0_m IFS Control Register p (IFSCRp_2SRC0_m) (m = 0, 1; p = 0, 1)

IFSCRp_2SRC0_m is a 32-bit readable/writable register that enables or disables usage of the initial value set by INTIFS.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	INT IFSEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	INTIFSEN	0	R/W	INTIFS Value Enable This bit is enabled only in asynchronous SRC mode (bit SRCMD in SRCCRp_2SRC0_m register is 0). 0: Disables initial value setting in IFSVR register. 1: Enables initial value setting in IFSVR register.

chanhvinh.com

37.3.22 2SRC0_m IFS Value Setting Register p (IFSVRp_2SRC0_m) (m = 0, 1; p = 0, 1)

IFSVRp_2SRC0_m is a 32-bit readable/writable register that sets the INTIFS value.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	INTIFS											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INTIFS															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27 to 0	INTIFS	All 0	R/W	Initial Value of FSI These bits set initial value of FSI. Set the input sampling rate and output sampling rate within the restricted range determined by the number of channels. Example: Fin = 32 kHz Fout = 44.1 kHz FSI = 2^{22} * 32000/44100 = 3043485 = 0x02E709D

When using the SRC in asynchronous mode, INTIFS is given as the initial value of FSI, which is the sampling conversion function. Therefore, after setting the INTIFSEN bit in IFSCRp_2SRC0_m to 1, set the INTIFS bits to the desired value. The SRC automatically detects the input sampling frequency and output sampling frequency. Based on this, the FSI value is calculated so that the following equation is satisfied.

$$\frac{Fin}{Fout} \approx \frac{FSI}{FSO}$$

[Legend]
Fin: Input sampling frequency
Fout: Output sampling frequency
FSI: Input sampling rate
FSO: Output sampling rate (fixed value: 2^{22} = 0x0400000)

When using the SRC in synchronous mode, the value set in INTIFS is used as the FSI value from beginning to end. The value set in INITFS is shown in Table 37.3 to Table 37.8.

chanhvinh.com

Table 37.3 INITFS Setting (Delay Mode: Normal, Number of Channels: 1 or 2)

Input Sampling Rate [KHz]	Output Sampling Rate [KHz]						
	8	16	24	32	44.1	48	96
8	0x0400000	0x0200000	0x0155555	0x0100000	0x00b9c27	0x00aaaaa	0x0055555
11.025	0x0583333	0x02c1999	0x01d6666	0x0160ccc	0x0100000	0x00eb333	0x0075999
12	0x0600000	0x0300000	0x0200000	0x0180000	0x0116a3b	0x0100000	0x0080000
16	0x0800000	0x0400000	0x02aaaaa	0x0200000	0x017384e	0x0155555	0x00aaaaa
22.05	0x0b06666	0x0583333	0x03acccc	0x02c1999	0x0200000	0x01d6666	0x00eb333
24	0x0c00000	0x0600000	0x0400000	0x0300000	0x022d476	0x0200000	0x0100000
32	0x1000000	0x0800000	0x0555555	0x0400000	0x02e709d	0x02aaaaa	0x0155555
44.1	0x160cccc	0x0b06666	0x0759999	0x0583333	0x0400000	0x03acccc	0x01d6666
48	0x1800000	0x0c00000	0x0800000	0x0600000	0x045a8ec	0x0400000	0x0200000
64	0x2000000	0x1000000	0x0aaaaaa	0x0800000	0x05ce13b	0x0555555	0x02aaaaa
88.2	Setting prohibited*1	0x160cccc	0x0eb3333	0x0b06666	0x0800000	0x0759999	0x03acccc
96	Setting prohibited*1	0x1800000	0x1000000	0x0c00000	0x08b51d9	0x0800000	0x0400000

Note 1. This is because the setting is outside the specification of the SRC module. For details, see Table 37.22.

Table 37.4 INITFS Setting (Delay Mode: Normal, Number of Channels: 4)

Input Sampling Rate [KHz]	Output Sampling Rate [KHz]						
	8	16	24	32	44.1	48	96
8	0x0400000	0x0200000	0x0155555	0x0100000	0x00b9c27	0x00aaaaa	0x0055555
11.025	0x0583333	0x02c1999	0x01d6666	0x0160ccc	0x0100000	0x00eb333	0x0075999
12	0x0600000	0x0300000	0x0200000	0x0180000	0x0116a3b	0x0100000	0x0080000
16	0x0800000	0x0400000	0x02aaaaa	0x0200000	0x017384e	0x0155555	0x00aaaaa
22.05	0x0b06666	0x0583333	0x03acccc	0x02c1999	0x0200000	0x01d6666	0x00eb333
24	0x0c00000	0x0600000	0x0400000	0x0300000	0x022d476	0x0200000	0x0100000
32	0x1000000	0x0800000	0x0555555	0x0400000	0x02e709d	0x02aaaaa	0x0155555
44.1	Setting prohibited*1	0x0b06666	0x0759999	0x0583333	0x0400000	0x03acccc	0x01d6666
48	Setting prohibited*1	0x0c00000	0x0800000	0x0600000	0x045a8ec	0x0400000	0x0200000
64	Setting prohibited*1	0x1000000	0x0aaaaaa	0x0800000	0x05ce13b	0x0555555	0x02aaaaa
88.2	Setting prohibited*1	Setting prohibited*1	0x0eb3333	0x0b06666	0x0800000	0x0759999	0x03acccc
96	Setting prohibited*1	Setting prohibited*1	0x1000000	0x0c00000	0x08b51d9	0x0800000	0x0400000

Note 1. This is because the setting is outside the specification of the SRC module. For details, see Table 37.22.

chanhvinh.com

Table 37.5 INITFS Setting (Delay Mode: Normal, Number of Channels: 6)

Input Sampling Rate [KHz]	Output Sampling Rate [KHz]						
	8	16	24	32	44.1	48	96
8	0x0400000	0x0200000	0x0155555	0x0100000	0x00b9c27	0x00aaaaa	Setting prohibited*1
11.025	0x0583333	0x02c1999	0x01d6666	0x0160ccc	0x0100000	0x00eb333	Setting prohibited*1
12	0x0600000	0x0300000	0x0200000	0x0180000	0x0116a3b	0x0100000	Setting prohibited*1
16	0x0800000	0x0400000	0x02aaaaa	0x0200000	0x017384e	0x0155555	Setting prohibited*1
22.05	Setting prohibited*1	0x0583333	0x03acccc	0x02c1999	0x0200000	0x01d6666	Setting prohibited*1
24	Setting prohibited*1	0x0600000	0x0400000	0x0300000	0x022d476	0x0200000	Setting prohibited*1
32	Setting prohibited*1	0x0800000	0x0555555	0x0400000	0x02e709d	0x02aaaaa	Setting prohibited*1
44.1	Setting prohibited*1	Setting prohibited*1	0x0759999	0x0583333	0x0400000	0x03acccc	Setting prohibited*1
48	Setting prohibited*1	Setting prohibited*1	0x0800000	0x0600000	0x045a8ec	0x0400000	Setting prohibited*1
64	Setting prohibited*1	Setting prohibited*1	0x0aaaaaa	0x0800000	0x05ce13b	0x0555555	Setting prohibited*1
88.2	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1
96	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1

Note 1. This is because the setting is outside the specification of the SRC module. For details, see Table 37.22.

chanhvinh.com

Table 37.6 INITFS Setting (Delay Mode: Normal, Number of Channels: 8)

Input Sampling Rate [KHz]	Output Sampling Rate [KHz]						
	8	16	24	32	44.1	48	96
8	0x0400000	0x0200000	0x0155555	0x0100000	0x00b9c27	0x00aaaaa	Setting prohibited*1
11.025	0x0583333	0x02c1999	0x01d6666	0x0160ccc	0x0100000	0x00eb333	Setting prohibited*1
12	0x0600000	0x0300000	0x0200000	0x0180000	0x0116a3b	0x0100000	Setting prohibited*1
16	0x0800000	0x0400000	0x02aaaaa	0x0200000	0x017384e	0x0155555	Setting prohibited*1
22.05	Setting prohibited*1	0x0583333	0x03acccc	0x02c1999	0x0200000	0x01d6666	Setting prohibited*1
24	Setting prohibited*1	0x0600000	0x0400000	0x0300000	0x022d476	0x0200000	Setting prohibited*1
32	Setting prohibited*1	0x0800000	0x0555555	0x0400000	0x02e709d	0x02aaaaa	Setting prohibited*1
44.1	Setting prohibited*1	Setting prohibited*1	0x0759999	0x0583333	0x0400000	0x03acccc	Setting prohibited*1
48	Setting prohibited*1	Setting prohibited*1	0x0800000	0x0600000	0x045a8ec	0x0400000	Setting prohibited*1
64	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1
88.2	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1
96	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1

Note 1. This is because the setting is outside the specification of the SRC module. For details, see Table 37.22.

chanhvinh.com

Table 37.7 INITFS Setting (Delay Mode: Low Delay 1, Number of Channels: 1 or 2)

Input Sampling Rate [KHz]	Output Sampling Rate [KHz]						
	8	16	24	32	44.1	48	96
8	0x0400000	0x0200000	0x0155555	0x0100000	0x00b9c27	0x00aaaaa	0x0055555
11.025	0x0583333	0x02c1999	0x01d6666	0x0160ccc	0x0100000	0x00eb333	0x0075999
12	0x0600000	0x0300000	0x0200000	0x0180000	0x0116a3b	0x0100000	0x0080000
16	0x0800000	0x0400000	0x02aaaaa	0x0200000	0x017384e	0x0155555	0x00aaaaa
22.05	Setting prohibited*1	0x0583333	0x03acccc	0x02c1999	0x0200000	0x01d6666	0x00eb333
24	Setting prohibited*1	0x0600000	0x0400000	0x0300000	0x022d476	0x0200000	0x0100000
32	Setting prohibited*1	0x0800000	0x0555555	0x0400000	0x02e709d	0x02aaaaa	0x0155555
44.1	Setting prohibited*1	Setting prohibited*1	0x0759999	0x0583333	0x0400000	0x03acccc	0x01d6666
48	Setting prohibited*1	Setting prohibited*1	0x0800000	0x0600000	0x045a8ec	0x0400000	0x0200000
64	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	0x0800000	0x05ce13b	0x0555555	0x02aaaaa
88.2	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	0x0800000	0x0759999	0x03acccc
96	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	0x0800000	0x0400000

Note 1. This is because the setting is outside the specification of the SRC module. For details, see Table 37.22.

Table 37.8 INITFS Setting (Delay Mode: Low Delay 2, Number of Channels: 1 or 2)

Input Sampling Rate [KHz]	Output Sampling Rate [KHz]						
	8	16	24	32	44.1	48	96
8	0x0400000	0x0200000	0x0155555	0x0100000	0x00b9c27	0x00aaaaa	0x0055555
11.025	Setting prohibited*1	0x02c1999	0x01d6666	0x0160ccc	0x0100000	0x00eb333	0x0075999
12	Setting prohibited*1	0x0300000	0x0200000	0x0180000	0x0116a3b	0x0100000	0x0080000
16	Setting prohibited*1	0x0400000	0x02aaaaa	0x0200000	0x017384e	0x0155555	0x00aaaaa
22.05	Setting prohibited*1	Setting prohibited*1	0x03acccc	0x02c1999	0x0200000	0x01d6666	0x00eb333
24	Setting prohibited*1	Setting prohibited*1	0x0400000	0x0300000	0x022d476	0x0200000	0x0100000
32	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	0x0400000	0x02e709d	0x02aaaaa	0x0155555
44.1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	0x0400000	0x03acccc	0x01d6666
48	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	0x0400000	0x0200000
64	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	0x02aaaaa
88.2	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	0x03acccc
96	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	Setting prohibited*1	0x0400000

Note 1. This is because the setting is outside the specification of the SRC module. For details, see Table 37.22.

chanhvinh.com

37.3.23 2SRC0_m Control Register p (SRCCRp_2SRC0_m) (m = 0, 1; p = 0, 1)

SRCCRp_2SRC0_m is a 32-bit readable/writable register that selects the low delay mode, and synchronous or asynchronous mode of the SRC.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	WATMD	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	BUFMD	-	-	-	-	-	-	-	-	-	-	-	SRCMD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	WATMD	0	R/W	Wait Time Control of SRC 0: Disables wait time control. 1: Enables wait time control.
19 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	—	0	R/W	The write value should always be 1.
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	BUFMD	0	R/W	Low Delay Mode 0: Disables low delay mode. 1: Enables low delay mode.
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	—	0	R/W	The write value should always be 1.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	—	0	R/W	The write value should always be 1.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SRCMD	0	R/W	SRC Mode Select 0: Asynchronous SRC 1: Synchronous SRC

chanhvinh.com

37.3.24 2SRC0_m Minimum FS Setting Register p (MNFSRp_2SRC0_m) (m = 0, 1; p = 0, 1)

MNFSRp_2SRC0_m is a 32-bit readable/writable register that sets the minimum value of FS.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	MINFS											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MINFS															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27 to 0	MINFS	All 0	R/W	Minimum Value of FS Ratio These bits set the FS ratio to start fading out the output data to mute under the situation that the input timing stops during SRC processing. This setting value should be less than the value of IFSVRp_2SRC0_m register. Example: Fin = 32kHz Fout = 44.1kHz FSI = 2^{22} * 32000/44100 = 3043485 = 0x02E709D MINFS = FSI * 98% = 2982615 = 0x02D82D7 Set the ratio which is out of the range of the jitter. (The recommended value is from 90 to 98 percent.)

37.3.25 2SRC0_m Buffer Size Setting Register p (BFSSRp_2SRC0_m) (m = 0, 1; p = 0, 1)

BFSSRp_2SRC0_m is a 32-bit readable/writable register that sets the buffer size. This register setting is valid only when SRCCRp_2SRC0_m.BUFMD = 1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	BUFDATA									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	BUFIN			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 16	BUFDATA	All 0	R/W	Set the buffer size of 1 channel in DATA RAM. Sets 0x80 for low delay mode 1. Sets 0x40 for low delay mode 2.
15 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	BUFIN	All 0	R/W	Set the buffer size of 1 channel in INDATA RAM. Sets 0x5 for low delay mode.

chanhvinh.com

37.3.26 2SRC0_m SCU2 Status Register p (SC2SRp_2SRC0_m) (m = 0, 1; p = 0, 1)

SC2SRp_2SRC0_m is a 32-bit read-only register that indicates the status of the SRC.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SRCWSTS	SC2MUTE	SC2STS		SC2FSI											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC2FSI															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	SRCWSTS	0	R	Status of Wait Time Operation of SRC 0: Wait for count up to wait time 1: Reached to wait time
30	SC2MUTE	0	R	SRC Mute Status This bit is only used for debug. 0: SRC output data is not fixed and it should be muted. This happens when SRC cannot synchronize the input sampling rate and output sampling rate. 1: SRC output data is fixed.
29, 28	SC2STS	All 0	R	Status of SRC 00: SRC is reset. 01: SRC is initialized. 10: SRC is processing. 11: Reserved
27 to 0	SC2FSI	All 0	R	Latest value of FSI which is calculated by SRC

37.3.27 2SRC0_m Wait Time Setting Register p (WATSRp_2SRC0_m) (m = 0, 1; p = 0, 1)

WATSRp_2SRC0_m is a 32-bit readable/writable register that sets the wait time for data input to the SRC.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	WTIME							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WTIME															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 0	WTIME	All 0	R/W	These bits set the wait time when WATMD bit of SRCCRp_2SRC0_m register is 1. After the INIT bit of SRCIRp_2SRC0_m register is set to 0, 0 data is stored into the SRC until the number of input data reaches to the setting value of these bits. The data input before the INIT bit is set to 0 is discarded. After the number of input data reaches to the setting value, input data is stored into the SRC.

chanhvinh.com

37.3.28 2SRC0_m Event Mask Register p (SEVMRp_2SRC0_m) (m = 0, 1; p = 0, 1)

SEVMRp_2SRC0_m is a 32-bit readable/writable register that enables or disables SCUAIm interrupt requests. The name of the interrupt source corresponding to the interrupt request from SEVMRp_2SRC0_0 is SCUAI0 and that corresponding to the interrupt request from SEVMRp_2SRC0_1 is SCUAI1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EVMUF	EVMOF	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	EVM WAIT	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	EVMUF	0	R/W	SRC Underflow Mask This bit sets whether to mask the interrupt request from the EVCUF bit in the SEVCRp_2SRC0_m register. 0: Disables interrupts. 1: Enables interrupts.
30	EVMOF	0	R/W	SRC Overflow Mask This bit sets whether to mask the EVCOF bit in the SEVCRp_2SRC0_m register. 0: Disables interrupts. 1: Enables interrupts.
29 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
14	EVMWAIT	0	R/W	SRC Wait Time Mask This bit sets whether to mask the interrupt request from the EVCWAIT bit of SEVCRp_2SRC0_m register. 0: Disables interrupts. 1: Enables interrupts.
13 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

37.3.29 2SRC0_m Event Clear Register p (SEVCRp_2SRC0_m) (m = 0, 1; p = 0, 1)

SEVCRp_2SRC0_m is a 32-bit readable/writable register that clears SCUAIm interrupt requests. When an interrupt event is generated, the relevant bit in this register is automatically set to 1 and 1 is retained until 0 is written to that bit.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EVCUF	EVCOF	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R/(W)*1	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	EVC WAIT	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/(W)*1	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	EVCUF	0	R/(W)*1	SRC Underflow Clear This is an interrupt flag to indicate whether INDATA RAM underflow occurs. When it occurs, this bit is set to 1. 0: Clears the flag. 1: Retains the flag.
30	EVCOF	0	R/(W)*1	SRC Overflow Clear This is an interrupt flag to indicate whether INDATA RAM overflow occurs. When it occurs, this bit is set to 1. 0: Clears the flag. 1: Retains the flag.
29 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
14	EVCWAIT	0	R/(W)*1	SRC Wait Time Clear This is an interrupt flag to indicate whether SRC has started processing after the wait time. 0: Clears the flag. 1: Retains the flag.
13 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note 1. Readable/writable. When 0 is written, the bit is initialized. Writing 1 to the bit is ignored. Write 0 only to each bit corresponding to the interrupt source to be cleared; write 1 to other bits.

chanhvinh.com

37.3.30 2SRC0_m Initialization Register RIF (SRCIRR_2SRC0_m) (m = 0, 1)

SRCIRR_2SRC0_m is a 32-bit readable/writable register that initializes COEF-ROMIF which is mounted on 2SRC0_m.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	INIT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	INIT	1	R/W	Initialization When this bit is set to 1, the COEF-ROMIF internal circuit is initialized. To cancel the initialization, set this bit to 0 with it set to 1. 0: Processing State 1: Initialization (sets the initial setting of other registers)

37.3.31 DVU0_n Initialization Register (DVUIR_DVU0_n) (n = 0, 1, 2, 3)

DVUIR_DVU0_n is a 32-bit readable/writable register that initializes the DVU internal circuit.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	INIT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	INIT	1	R/W	Initialization When this bit is set to 1, the DVU internal circuit is initialized. To cancel the initialization, set this bit to 0 with it set to 1. 0: Processing State 1: Initialization (sets the initial setting of other registers)

chanhvinh.com

37.3.32 DVU0_n Audio Information Register (VADIR_DVU0_n) (n = 0, 1, 2, 3)

VADIR_DVU0_n is a 32-bit readable/writable register that sets the bit length of data and the number of channels.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	OTBL				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	CHNUM			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20 to 16	OTBL	All 0	R/W	Bit Length of Output Audio Data. 00000: 24 bits 00001 to 00111: Reserved 01000: 16 bits 01001 to 11111: Reserved
15 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	CHNUM	All 0	R/W	Number of Channels These bits set the number of channels. 0000: Zero (None) 0001: 1 channel 0010: 2 channels 0011: Reserved 0100: 4 channels 0101: Reserved 0110: 6 channels 0111: Reserved 1000: 8 channels 1001 to 1111: Reserved

chanhvinh.com

37.3.33 DVU0_n Bypass Register (DVUBR_DVU0_n) (n = 0, 1, 2, 3)

DVUBR_DVU0_n is a 32-bit readable/writable register that sets the bypass mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	BY PASS
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	BYPASS	0	R/W	Bypass Mode This bit controls the data pass of DVU. 0: DVU function is used. Input data is processed by DVU and then the result data is connected to output data. 1: DVU function is not used. Input data is directly connected to output data.

37.3.34 DVU0_n Control Register (DVUCR_DVU0_n) (n = 0, 1, 2, 3)

DVUCR_DVU0_n is a 32-bit readable/writable register that selects the DVU mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	VVMD	-	-	-	VRMD	-	-	-	ZCMD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	VVMD	0	R/W	Digital Volume Mode Select This bit selects the digital volume function. 0: Disables the digital volume function. 1: Enables the digital volume function.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	VRMD	0	R/W	Volume Ramp Mode Select This bit selects the volume ramp function. 0: Disables the volume ramp function. 1: Enables the volume ramp function.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ZCMD	0	R/W	Zero Cross Mute Mode Select This bit selects the zero cross mute function. 0: Disables the zero cross mute function. 1: Enables the zero cross mute function.

chanhvinh.com

37.3.35 DVU0_n Zero Cross Mute Control Register (ZCMCR_DVU0_n) (n = 0, 1, 2, 3)

ZCMCR_DVU0_n is a 32-bit readable/writable register that controls operation of the zero cross mute function for each channel.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	ZCEN7	ZCEN6	ZCEN5	ZCEN4	ZCEN3	ZCEN2	ZCEN1	ZCEN0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	ZCEN7	0	R/W	Zero Cross Mute Enable for Channel 7 This bit controls the operation of the zero cross mute function for channel 7. 0: Disables the operation of the zero cross mute function. 1: Enables the operation the zero cross mute function.
6	ZCEN6	0	R/W	Zero Cross Mute Enable for Channel 6 This bit controls the operation of the zero cross mute function for channel 6. 0: Disables the operation of the zero cross mute function. 1: Enables the operation the zero cross mute function.
5	ZCEN5	0	R/W	Zero Cross Mute Enable for Channel 5 This bit controls the operation of the zero cross mute function for channel 5. 0: Disables the operation of the zero cross mute function. 1: Enables the operation the zero cross mute function.
4	ZCEN4	0	R/W	Zero Cross Mute Enable for Channel 4 This bit controls the operation of the zero cross mute function for channel 4. 0: Disables the operation of the zero cross mute function. 1: Enables the operation the zero cross mute function.
3	ZCEN3	0	R/W	Zero Cross Mute Enable for Channel 3 This bit controls the operation of the zero cross mute function for channel 3. 0: Disables the operation of the zero cross mute function. 1: Enables the operation the zero cross mute function.
2	ZCEN2	0	R/W	Zero Cross Mute Enable for Channel 2 This bit controls the operation of the zero cross mute function for channel 2. 0: Disables the operation of the zero cross mute function. 1: Enables the operation the zero cross mute function.
1	ZCEN1	0	R/W	Zero Cross Mute Enable for Channel 1 This bit controls the operation of the zero cross mute function for channel 1. 0: Disables the operation of the zero cross mute function. 1: Enables the operation the zero cross mute function.
0	ZCEN0	0	R/W	Zero Cross Mute Enable for Channel 0 This bit controls the operation of the zero cross mute function for channel 0. 0: Disables the operation of the zero cross mute function. 1: Enables the operation the zero cross mute function.

chanhvinh.com

37.3.36 DVU0_n Volume Ramp Control Register (VRCTR_DVU0_n) (n = 0, 1, 2, 3)

VRCTR_DVU0_n is a 32-bit readable/writable register that controls operation of the volume ramp function.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	VREN7	VREN6	VREN5	VREN4	VREN3	VREN2	VREN1	VREN0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	VREN7	0	R/W	Volume Ramp Enable for Channel 7 This bit controls the operation of the volume ramp function for channel 7. 0: Disables the operation of the volume ramp function. 1: Enables the operation of the volume ramp function.
6	VREN6	0	R/W	Volume Ramp Enable for Channel 6 This bit controls the operation of the volume ramp function for channel 6. 0: Disables the operation of the volume ramp function. 1: Enables the operation of the volume ramp function.
5	VREN5	0	R/W	Volume Ramp Enable for Channel 5 This bit controls the operation of the volume ramp function for channel 5. 0: Disables the operation of the volume ramp function. 1: Enables the operation of the volume ramp function.
4	VREN4	0	R/W	Volume Ramp Enable for Channel 4 This bit controls the operation of the volume ramp function for channel 4. 0: Disables the operation of the volume ramp function. 1: Enables the operation of the volume ramp function.
3	VREN3	0	R/W	Volume Ramp Enable for Channel 3 This bit controls the operation of the volume ramp function for channel 3. 0: Disables the operation of the volume ramp function. 1: Enables the operation of the volume ramp function.
2	VREN2	0	R/W	Volume Ramp Enable for Channel 2 This bit controls the operation of the volume ramp function for channel 2. 0: Disables the operation of the volume ramp function. 1: Enables the operation of the volume ramp function.
1	VREN1	0	R/W	Volume Ramp Enable for Channel 1 This bit controls the operation of the volume ramp function for channel 1. 0: Disables the operation of the volume ramp function. 1: Enables the operation of the volume ramp function.
0	VREN0	0	R/W	Volume Ramp Enable for Channel 0 This bit controls the operation of the volume ramp function for channel 0. 0: Disables the operation of the volume ramp function. 1: Enables the operation of the volume ramp function.

chanhvinh.com

37.3.37 DVU0_n Volume Ramp Period Register (VRPDR_DVU0_n) (n = 0, 1, 2, 3)

VRPDR_DVU0_n is a 32-bit readable/writable register that sets the ramp time of volume ramp.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	VRPDUP					-	-	-	VRPDDW				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12 to 8	VRPDUP	All 0	R/W	Volume Ramp Period for Volume Up 00000: 1 [sample] (128 dB/1 step) 00001: 2 [sample] (64 dB/1 step) 00010: 4 [sample] (32 dB/1 step) 00011: 8 [sample] (16 dB/1 step) 00100: 16 [sample] (8 dB/1 step) 00101: 32 [sample] (4 dB/1 step) 00110: 64 [sample] (2 dB/1 step) 00111: 128 [sample] (1 dB/1 step) 01000: 256 [sample] (0.5 dB/1 step) 01001: 512 [sample] (0.25 dB/1 step) 01010: 1024 [sample] (0.125 dB/1 step) 01011: 2048 [sample] (0.125 dB/2 steps) 01100: 4096 [sample] (0.125 dB/4 steps) 01101: 8192 [sample] (0.125 dB/8 steps) 01110: 16384 [sample] (0.125 dB/16 steps) 01111: 32768 [sample] (0.125 dB/32 steps) 10000: 65536 [sample] (0.125 dB/64 steps) 10001: 131072 [sample] (0.125 dB/128 steps) 10010: 262144 [sample] (0.125 dB/256 steps) 10011: 524288 [sample] (0.125 dB/512 steps) 10100: 1048576 [sample] (0.125 dB/1024 steps) 10101: 2097152 [sample] (0.125 dB/2048 steps) 10110: 4194304 [sample] (0.125 dB/4096 steps) 10111: 8388608 [sample] (0.125 dB/8192 steps) 11000 to 11111: Reserved
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
4 to 0	VRPDDW	0	R/W	Volume Ramp Period for Volume Down 00000: 1 [sample] (-128 dB/1 step) 00001: 2 [sample] (-64 dB/1 step) 00010: 4 [sample] (-32 dB/1 step) 00011: 8 [sample] (-16 dB/1 step) 00100: 16 [sample] (-8 dB/1 step) 00101: 32 [sample] (-4 dB/1 step) 00110: 64 [sample] (-2 dB/1 step) 00111: 128 [sample] (-1 dB/1 step) 01000: 256 [sample] (-0.5 dB/1 step) 01001: 512 [sample] (-0.25 dB/1 step) 01010: 1024 [sample] (-0.125 dB/1 step) 01011: 2048 [sample] (-0.125 dB/2 steps) 01100: 4096 [sample] (-0.125 dB/4 steps) 01101: 8192 [sample] (-0.125 dB/8 steps) 01110: 16384 [sample] (-0.125 dB/16 steps) 01111: 32768 [sample] (-0.125 dB/32 steps) 10000: 65536 [sample] (-0.125 dB/64 steps) 10001: 131072 [sample] (-0.125 dB/128 steps) 10010: 262144 [sample] (-0.125 dB/256 steps) 10011: 524288 [sample] (-0.125 dB/512 steps) 10100: 1048576 [sample] (-0.125 dB/1024 steps) 10101: 2097152 [sample] (-0.125 dB/2048 steps) 10110: 4194304 [sample] (-0.125 dB/4096 steps) 10111: 8388608 [sample] (-0.125 dB/8192 steps) 11000 to 11111: Reserved

37.3.38 DVU0_n Volume Ramp Decibel Register (VRDBR_DVU0_n) (n = 0, 1, 2, 3)

VRDBR_DVU0_n is a 32-bit readable/writable register that sets the gain level (decibel units) of volume ramp.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	VRDB									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	VRDB	All 0	R/W	Decibel of Volume Ramp These bits set the decibel (gain level) of volume ramp. The value can be selected from 1024 points in the range of f0 dB to -∞ dB at intervals of 0.125 dB.

Setting Value	Time	Gain[dB]	Setting Value	Time	Gain[dB]
0x000	1	0	.....	.....	.....
.....	.....	.....	0x091	0.125	-18.125
0x031	0.5	-6.125	.....	.....	.....
.....	.....	.....	0x3FE	4.1×10^{-7}	-127.75
0x061	0.25	-12.125	0x3FF	0 (Mute)	-∞

chanhvinh.com

37.3.39 DVU0_n Volume Ramp Wait Time Register (VRWTR_DVU0_n) (n = 0, 1, 2, 3)

VRWTR_DVU0_n is a 32-bit readable/writable register that sets the wait time before executing the volume ramp function.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	VRWT							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VRWT															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 0	VRWT	All 0	R/W	Volume Ramp Wait Time These bits set the wait time before starting the volume ramp function when the setting of VRDBR register is changed. If the internal counter of the DVU logic reached to the value of these bits, the volume ramp function starts operating to change the volume to the value changed in the VRDBR register as a target.

37.3.40 DVU0_n Volume Value Setting 0 Register (VOL0R_DVU0_n) (n = 0, 1, 2, 3)

VOL0R_DVU0_n is a 32-bit readable/writable register that sets the digital volume value of channel 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	VOLVAL0							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VOLVAL0															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 0	VOLVAL0	All 0	R/W	Digital Volume Value for Channel 0 These bits set the digital volume of channel 0. The maximum value is octupled gain (18 dB) and the minimum value is 0 (-∞ dB). VOLVAL0[23] : Sign bit VOLVAL0[22:20] : Integer bits VOLVAL0[19:0] : Decimal bits

Setting Value	Time	Gain[dB]	Setting Value	Time	Gain[dB]
0x7F_FFFF	8	18	0x08_0000	0.5	-6
...	...	...	...	...	...
0x10_0000	1	0	0x00_0001	9.5×10^{-7}	-120
...	...	...	0x00_0000	0	-∞

chanhvinh.com

37.3.41 DVU0_n Volume Value Setting 1 Register (VOL1R_DVU0_n) (n = 0, 1, 2, 3)

VOL1R_DVU0_n is a 32-bit readable/writable register that sets the digital volume value of channel 1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	VOLVAL1							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VOLVAL1															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 0	VOLVAL1	All 0	R/W	Digital Volume Value for Channel 1 These bits set the digital volume of channel 1. The maximum value is octupled gain (18 dB) and the minimum value is 0 (-∞ dB). VOLVAL1[23] : Sign bit VOLVAL1[22:20] : Integer bits VOLVAL1[19:0] : Decimal bits

Setting Value	Time	Gain[dB]	Setting Value	Time	Gain[dB]
0x7F_FFFF	8	18	0x08_0000	0.5	-6
...	...	...	...	...	...
0x10_0000	1	0	0x00_0001	9.5×10^{-7}	-120
...	...	...	0x00_0000	0	-∞

chanhvinh.com

37.3.42 DVU0_n Volume Value Setting 2 Register (VOL2R_DVU0_n) (n = 0, 1, 2, 3)

VOL2R_DVU0_n is a 32-bit readable/writable register that sets the digital volume value of channel 2.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	VOLVAL2							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VOLVAL2															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 0	VOLVAL2	All 0	R/W	Digital Volume Value for Channel 2 These bits set the digital volume of channel 2. The maximum value is octupled gain (18 dB) and the minimum value is 0 (-∞ dB). VOLVAL2[23] : Sign bit VOLVAL2[22:20] : Integer bits VOLVAL2[19:0] : Decimal bits

Setting Value	Time	Gain[dB]	Setting Value	Time	Gain[dB]
0x7F_FFFF	8	18	0x08_0000	0.5	-6
...	...	...	...	...	...
0x10_0000	1	0	0x00_0001	9.5×10^{-7}	-120
...	...	...	0x00_0000	0	-∞

chanhvinh.com

37.3.43 DVU0_n Volume Value Setting 3 Register (VOL3R_DVU0_n) (n = 0, 1, 2, 3)

VOL3R_DVU0_n is a 32-bit readable/writable register that sets the digital volume value of channel 3.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	VOLVAL3							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VOLVAL3															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 0	VOLVAL3	All 0	R/W	Digital Volume Value for Channel 3 These bits set the digital volume of channel 3. The maximum value is octupled gain (18 dB) and the minimum value is 0 (-∞ dB). VOLVAL3[23] : Sign bit VOLVAL3[22:20] : Integer bits VOLVAL3[19:0] : Decimal bits

Setting Value	Time	Gain[dB]	Setting Value	Time	Gain[dB]
0x7F_FFFF	8	18	0x08_0000	0.5	-6
...	...	...	...	...	...
0x10_0000	1	0	0x00_0001	9.5×10^{-7}	-120
...	...	...	0x00_0000	0	-∞

chanhvinh.com

37.3.44 DVU0_n Volume Value Setting 4 Register (VOL4R_DVU0_n) (n = 0, 1, 2, 3)

VOL4R_DVU0_n is a 32-bit readable/writable register that sets the digital volume value of channel 4.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	VOLVAL4							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VOLVAL4															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 0	VOLVAL4	All 0	R/W	Digital Volume Value for Channel 4 These bits set the digital volume of channel 4. The maximum value is octupled gain (18 dB) and the minimum value is 0 (-∞ dB). VOLVAL4[23] : Sign bit VOLVAL4[22:20] : Integer bits VOLVAL4[19:0] : Decimal bits

Setting Value	Time	Gain[dB]	Setting Value	Time	Gain[dB]
0x7F_FFFF	8	18	0x08_0000	0.5	-6
...	...	...	...	...	...
0x10_0000	1	0	0x00_0001	9.5×10^{-7}	-120
...	...	...	0x00_0000	0	-∞

chanhvinh.com

37.3.45 DVU0_n Volume Value Setting 5 Register (VOL5R_DVU0_n) (n = 0, 1, 2, 3)

VOL5R_DVU0_n is a 32-bit readable/writable register that sets the digital volume value of channel 5.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	VOLVAL5							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VOLVAL5															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 0	VOLVAL5	All 0	R/W	Digital Volume Value for Channel 5 These bits set the digital volume of channel 5. The maximum value is octupled gain (18 dB) and the minimum value is 0 (-∞ dB). VOLVAL5[23] : Sign bit VOLVAL5[22:20] : Integer bits VOLVAL5[19:0] : Decimal bits

Setting Value	Time	Gain[dB]	Setting Value	Time	Gain[dB]
0x7F_FFFF	8	18	0x08_0000	0.5	-6
...	...	...	...	...	...
0x10_0000	1	0	0x00_0001	9.5×10^{-7}	-120
...	...	...	0x00_0000	0	-∞

chanhvinh.com

37.3.46 DVU0_n Volume Value Setting 6 Register (VOL6R_DVU0_n) (n = 0, 1, 2, 3)

VOL6R_DVU0_n is a 32-bit readable/writable register that sets the digital volume value of channel 6.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	VOLVAL6							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VOLVAL6															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 0	VOLVAL6	All 0	R/W	Digital Volume Value for Channel 6 These bits set the digital volume of channel 6. The maximum value is octupled gain (18 dB) and the minimum value is 0 (-∞ dB). VOLVAL6[23] : Sign bit VOLVAL6[22:20] : Integer bits VOLVAL6[19:0] : Decimal bits

Setting Value	Time	Gain[dB]	Setting Value	Time	Gain[dB]
0x7F_FFFF	8	18	0x08_0000	0.5	-6
...	...	...	...	...	...
0x10_0000	1	0	0x00_0001	9.5×10^{-7}	-120
...	...	...	0x00_0000	0	-∞

chanhvinh.com

37.3.47 DVU0_n Volume Value Setting 7 Register (VOL7R_DVU0_n) (n = 0, 1, 2, 3)

VOL7R_DVU0_n is a 32-bit readable/writable register that sets the digital volume value of channel 7.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	VOLVAL7							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VOLVAL7															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 0	VOLVAL7	All 0	R/W	Digital Volume Value for Channel7 These bits set the digital volume of channel 7. The maximum value is octupled gain (18 dB) and the minimum value is 0 (-∞ dB). VOLVAL7[23] : Sign bit VOLVAL7[22:20] : Integer bits VOLVAL7[19:0] : Decimal bits

Setting Value	Time	Gain[dB]	Setting Value	Time	Gain[dB]
0x7F_FFFF	8	18	0x08_0000	0.5	-6
...	...	...	...	...	...
0x10_0000	1	0	0x00_0001	9.5×10^{-7}	-120
...	...	...	0x00_0000	0	-∞

37.3.48 DVU0_n Enable Register (DVUER_DVU0_n) (n = 0, 1, 2, 3)

DVUER_DVU0_n is a 32-bit readable/writable register that enables or disables the DVU register settings.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DVUEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	DVUEN	0	R/W	DVU Register Setting Enable This bit controls the setting of DVU registers (ZCMCR, VRCTR, VRPDR, VRDBR, VOL0R, VOL1R, VOL2R, VOL3R, VOL4R, VOL5R, VOL6R, and VOL7R). 0: Disables the setting of DVU registers. 1: Enables the setting of DVU registers.

chanhvinh.com

37.3.49 DVU0_n Status Register (DVUSR_DVU0_n) (n = 0, 1, 2, 3)

DVUSR_DVU0_n is a 32-bit read-only register that indicates the status of zero cross mute and volume ramp.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	ZSTS7	ZSTS6	ZSTS5	ZSTS4	ZSTS3	ZSTS2	ZSTS1	ZSTS0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	VRSTS		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23	ZSTS7	0	R	Zero Cross Mute Status of Channel 7 This bit indicates the zero cross mute status of channel 7. 0: Not in mute status 1: Mute status
22	ZSTS6	0	R	Zero Cross Mute Status of Channel 6 This bit indicates the zero cross mute status of channel 6. 0: Not in mute status 1: Mute status
21	ZSTS5	0	R	Zero Cross Mute Status of Channel 5 This bit indicates the zero cross mute status of channel 5. 0: Not in mute status 1: Mute status
20	ZSTS4	0	R	Zero Cross Mute Status of Channel 4 This bit indicates the zero cross mute status of channel 4. 0: Not in mute status 1: Mute status
19	ZSTS3	0	R	Zero Cross Mute Status of Channel 3 This bit indicates the zero cross mute status of channel 3. 0: Not in mute status 1: Mute status
18	ZSTS2	0	R	Zero Cross Mute Status of Channel 2 This bit indicates the zero cross mute status of channel 2. 0: Not in mute status 1: Mute status
17	ZSTS1	0	R	Zero Cross Mute Status of Channel 1 This bit indicates the zero cross mute status of channel 1. 0: Not in mute status 1: Mute status
16	ZSTS0	0	R	Zero Cross Mute Status of Channel 0 This bit indicates the zero cross mute status of channel 0. 0: Not in mute status 1: Mute status
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	VRSTS	All 0	R	Volume Ramp Status These bits indicate the volume ramp status. 000: Mute status 001: Volume ramp down 010: Volume ramp up 011: Level of volume ramp is in the state of the value of VRDBR register 100: Volume of input data is maintained (Volume is not multiplied). 101 to 111: Reserved

chanhvinh.com

37.3.50 DVU0_n Event Mask Register (VEVMR_DVU0_n) (n = 0, 1, 2, 3)

VEVMR_DVU0_n is a 32-bit readable/writable register that enables or disables SCUDVIn interrupt requests.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VEVMZCM7	VEVMZCM6	VEVMZCM5	VEVMZCM4	VEVMZCM3	VEVMZCM2	VEVMZCM1	VEVMZCM0	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VEVMVR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	VEVMZCM7	0	R/W	Zero Cross Mute Status Change Mask of Channel 7 This bit sets whether to mask the interrupt request from the VEVCZCM7 bit in the VEVCR_DVU0_n register. 0: Disables interrupts. 1: Enables interrupts.
30	VEVMZCM6	0	R/W	Zero Cross Mute Status Change Mask of Channel 6 This bit sets whether to mask the interrupt request from the VEVCZCM6 bit in the VEVCR_DVU0_n register. 0: Disables interrupts. 1: Enables interrupts.
29	VEVMZCM5	0	R/W	Zero Cross Mute Status Change Mask of Channel 5 This bit sets whether to mask the interrupt request from the VEVCZCM5 bit in the VEVCR_DVU0_n register. 0: Disables interrupts. 1: Enables interrupts.
28	VEVMZCM4	0	R/W	Zero Cross Mute Status Change Mask of Channel 4 This bit sets whether to mask the interrupt request from the VEVCZCM4 bit in the VEVCR_DVU0_n register. 0: Disables interrupts. 1: Enables interrupts.
27	VEVMZCM3	0	R/W	Zero Cross Mute Status Change Mask of Channel 3 This bit sets whether to mask the interrupt request from the VEVCZCM3 bit in the VEVCR_DVU0_n register. 0: Disables interrupts. 1: Enables interrupts.
26	VEVMZCM2	0	R/W	Zero Cross Mute Status Change Mask of Channel 2 This bit sets whether to mask the interrupt request from the VEVCZCM2 bit in the VEVCR_DVU0_n register. 0: Disables interrupts. 1: Enables interrupts.
25	VEVMZCM1	0	R/W	Zero Cross Mute Status Change Mask of Channel 1 This bit sets whether to mask the interrupt request from the VEVCZCM1 bit in the VEVCR_DVU0_n register. 0: Disables interrupts. 1: Enables interrupts.
24	VEVMZCM0	0	R/W	Zero Cross Mute Status Change Mask of Channel 0 This bit sets whether to mask the interrupt request from the VEVCZCM0 bit in the VEVCR_DVU0_n register. 0: Disables interrupts. 1: Enables interrupts.
23 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15	VEVMVR	0	R/W	Volume Ramp Status Change This bit sets whether to mask the interrupt request from the VEVCVR bit in the VEVCR_DVU0_n register. 0: Disables interrupts. 1: Enables interrupts.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
14 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

37.3.51 DVU0_n Event Clear Register (VEVCR_DVU0_n) (n = 0, 1, 2, 3)

VEVCR_DVU0_n is a 32-bit readable/writable register that clears SCUDVIn interrupt requests. When an interrupt event is generated, the relevant bit in this register is automatically set to 1 and 1 is retained until 0 is written to that bit.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VEVCZCM7	VEVCZCM6	VEVCZCM5	VEVCZCM4	VEVCZCM3	VEVCZCM2	VEVCZCM1	VEVCZCM0	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VEVCVR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	VEVCZCM7	0	R/(W)*1	Zero Cross Mute Status Change Clear of Channel 7 This is an interrupt flag to indicate whether the zero cross mute status change of channel 7 occurs. When it occurs, this bit is set to 1. 0: Clears the flag. 1: Retains the flag.
30	VEVCZCM6	0	R/(W)*1	Zero Cross Mute Status Change Clear of Channel 6 This is an interrupt flag to indicate whether the zero cross mute status change of channel 6 occurs. When it occurs, this bit is set to 1. 0: Clears the flag. 1: Retains the flag.
29	VEVCZCM5	0	R/(W)*1	Zero Cross Mute Status Change Clear of Channel 5 This is an interrupt flag to indicate whether the zero cross mute status change of channel 5 occurs. When it occurs, this bit is set to 1. 0: Clears the flag. 1: Retains the flag.
28	VEVCZCM4	0	R/(W)*1	Zero Cross Mute Status Change Clear of Channel 4 This is an interrupt flag to indicate whether the zero cross mute status change of channel 4 occurs. When it occurs, this bit is set to 1. 0: Clears the flag. 1: Retains the flag.
27	VEVCZCM3	0	R/(W)*1	Zero Cross Mute Status Change Clear of Channel 3 This is an interrupt flag to indicate whether the zero cross mute status change of channel 3 occurs. When it occurs, this bit is set to 1. 0: Clears the flag. 1: Retains the flag.
26	VEVCZCM2	0	R/(W)*1	Zero Cross Mute Status Change Clear of Channel 2 This is an interrupt flag to indicate whether the zero cross mute status change of channel 2 occurs. When it occurs, this bit is set to 1. 0: Clears the flag. 1: Retains the flag.
25	VEVCZCM1	0	R/(W)*1	Zero Cross Mute Status Change Clear of Channel 1 This is an interrupt flag to indicate whether the zero cross mute status change of channel 1 occurs. When it occurs, this bit is set to 1. 0: Clears the flag. 1: Retains the flag.
24	VEVCZCM0	0	R/(W)*1	Zero Cross Mute Status Change Clear of Channel 0 This is an interrupt flag to indicate whether the zero cross mute status change of channel 0 occurs. When it occurs, this bit is set to 1. 0: Clears the flag. 1: Retains the flag.
23 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
15	VEVCVR	0	R/(W)*1	Volume Ramp Status Change Clear This is an interrupt flag to judge volume ramp mute occurs or not. When it occurs, this bit is set to 1. 0: Clears the flag. 1: Retains the flag.
14 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note 1. Readable/writable. When 0 is written, the bit is initialized. Writing 1 to the bit is ignored. Write 0 only to each bit corresponding to the interrupt source to be cleared; write 1 to other bits.

37.3.52 MIX0_0 Initialization Register (MIXIR_MIX0_0)

MIXIR_MIX0_0 is a 32-bit readable/writable register that initializes the MIX internal circuit.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	INIT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	INIT	1	R/W	Initialization When this bit is set to 1, the MIX internal circuit is initialized. To cancel the initialization, set this bit to 0 with it set to 1. 0: Processing State 1: Initialization (sets the initial setting of other registers)

chanhvinh.com

37.3.53 MIX0_0 Audio Information Register (MADIR_MIX0_0)

MADIR_MIX0_0 is a 32-bit readable/writable register that sets the number of channels.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	CHNUM			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	CHNUM	All 0	R/W	Number of Channels These bits set the number of channels. 0000: Zero (None) 0001: 1 channel 0010: 2 channels 0011: Reserved 0100: 4 channels 0101: Reserved 0110: 6 channels 0111: Reserved 1000: 8 channels 1001 to 1111: Reserved

chanhvinh.com

37.3.54 MIX0_0 Bypass Register (MIXBR_MIX0_0)

MIXBR_MIX0_0 is a 32-bit readable/writable register that sets the bypass mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	BPSYS	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	BY PASS
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	BPSYS	All 0	R/W	Bypass System Select This bit selects the input data output from the MIX module when the BYPASS bit is set to 1. 00: Input data of system A 01: Input data of system B 10: Input data of system C 11: Input data of system D
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	BYPASS	0	R/W	Bypass Mode This bit controls the data pass of MIX function. 0: MIX function is used. Input data is processed by MIX and then the result data is connected to output data. 1: SRC function is not used. Input data is connected to output data according to the BPSYS bits.

37.3.55 MIX0_0 Mode Register (MIXMR_MIX0_0)

MIXMR_MIX0_0 is a 32-bit readable/writable register that selects the MIX mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	MIX MODE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	MIXMODE	0	R/W	MIX mode This bit sets the mix mode. 0: Selects volume step mixer. 1: Selects volume ramp mixer.

chanhvinh.com

37.3.56 MIX0_0 Volume Period Register (MVPDR_MIX0_0)

MVPDR_MIX0_0 is a 32-bit readable/writable register that sets the ramp time of volume ramp.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	MXPDUP				-	-	-	-	MXPDDW			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 8	MXPDUP	0	R/W	Volume Up Period These bits set the decibel which changes per sample during volume up. This setting is used when the MIXMODE bit in the MIXMR register is set to 1. 0000: 128 dB/1 sample 0001: 64 dB/1 sample 0010: 32 dB/1 sample 0011: 16 dB/1 sample 0100: 8 dB/1 sample 0101: 4 dB/1 sample 0110: 2 dB/1 sample 0111: 1 dB/1 sample 1000: 0.5 dB/1 sample 1001: 0.25 dB/1 sample 1010: 0.125 dB/1 sample 1011 to 1111: Reserved
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	MXPDDW	0	R/W	Volume Down Period These bits set the decibel which changes per sample during volume down. This setting is used when the MIXMODE bit in the MIXMR register is set to 1. 0000: -128 dB/1 sample 0001: -64 dB/1 sample 0010: -32 dB/1 sample 0011: -16 dB/1 sample 0100: -8 dB/1 sample 0101: -4 dB/1 sample 0110: -2 dB/1 sample 0111: -1 dB/1 sample 1000: -0.5 dB/1 sample 1001: -0.25 dB/1 sample 1010: -0.125 dB/1 sample 1011 to 1111: Reserved

chanhvinh.com

37.3.57 MIX0_0 Decibel A Register (MDBAR_MIX0_0)

MDBAR_MIX0_0 is a 32-bit readable/writable register that sets the gain level (decibel units) of system A.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	MIXDBA									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	MIXDBA	All 0	R/W	dDecibel of System A These bits set the decibel (gain level) of volume ramp. The value can be selected from 1024 points in the range of f0 dB to -∞ dB at intervals of 0.125 dB.

Setting Value	Time	Gain[dB]	Setting Value	Time	Gain[dB]
0x000	1	0	.....	.....	.....
.....	.....	.....	0x091	0.125	-18.125
0x031	0.5	-6.125	.....	.....	.....
.....	.....	.....	0x3FE	4.1×10^{-7}	-127.75
0x061	0.25	-12.125	0x3FF	0 (Mute)	-∞

chanhvinh.com

37.3.58 MIX0_0 Decibel B Register (MDBBR_MIX0_0)

MDBBR_MIX0_0 is a 32-bit readable/writable register that sets the gain level (decibel units) of system B.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	MIXDBB									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	MIXDBB	All 0	R/W	Decibel of System B These bits set the decibel (gain level) of volume ramp. The value can be selected from 1024 points in the range of f0 dB to -∞ dB at intervals of 0.125 dB.

Setting Value	Time	Gain[dB]	Setting Value	Time	Gain[dB]
0x000	1	0	.....	.....	.....
.....	.....	.....	0x091	0.125	-18.125
0x031	0.5	-6.125	.....	.....	.....
.....	.....	.....	0x3FE	4.1×10^{-7}	-127.75
0x061	0.25	-12.125	0x3FF	0 (Mute)	-∞

chanhvinh.com

37.3.59 MIX0_0 Decibel C Register (MDBCR_MIX0_0)

MDBCR_MIX0_0 is a 32-bit readable/writable register that sets the gain level (decibel units) of system C.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	MIXDBC									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	MIXDBC	All 0	R/W	Decibel of System C These bits set the decibel (gain level) of volume ramp. The value can be selected from 1024 points in the range of f0 dB to -∞ dB at intervals of 0.125 dB.

Setting Value	Time	Gain[dB]	Setting Value	Time	Gain[dB]
0x000	1	0	.....	.....	.....
.....	.....	.....	0x091	0.125	-18.125
0x031	0.5	-6.125	.....	.....	.....
.....	.....	.....	0x3FE	4.1×10^{-7}	-127.75
0x061	0.25	-12.125	0x3FF	0 (Mute)	-∞

chanhvinh.com

37.3.60 MIX0_0 Decibel D Register (MDBDR_MIX0_0)

MDBDR_MIX0_0 is a 32-bit readable/writable register that sets the gain level (decibel units) of system D.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	MIXDBD									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	MIXDBD	All 0	R/W	Decibel of System D These bits set the decibel (gain level) of volume ramp. The value can be selected from 1024 points in the range of f0 dB to -∞ dB at intervals of 0.125 dB.

Setting Value	Time	Gain[dB]	Setting Value	Time	Gain[dB]
0x000	1	0	.....	.....	.....
.....	.....	.....	0x091	0.125	-18.125
0x031	0.5	-6.125	.....	.....	.....
.....	.....	.....	0x3FE	4.1×10^{-7}	-127.75
0x061	0.25	-12.125	0x3FF	0 (Mute)	-∞

chanhvinh.com

37.3.61 MIX0_0 Decibel Enable Register (MDBER_MIX0_0)

MDBER_MIX0_0 is a 32-bit readable/writable register that enable or disables the MIX decibel register settings.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	MIXDBEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	MIXDBEN	0	R/W	MIX Decibel Enable This bit controls the decibel dB set in MDBAR, MDBBR, MDBCR and MDBDR registers. 0: Disables the setting of decibel. 1: Enables the setting of decibel.

37.3.62 MIX0_0 Status Register (MIXSR_MIX0_0)

MIXSR_MIX0_0 is a 32-bit read-only register that indicates the status of the MIX.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	MIXSTS	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	MIXSTS	All 0	R	MIX Status This bit specifies the status of ramp process. 00: Ramp process is unchanged. 01: Ramp down 10: Ramp up

chanhvinh.com

37.3.63 Software Reset Register (SWRSR_CIM)

SWRSR_CIM is a 32-bit readable/writable register that initializes the SCUX internal circuit.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SWRST
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SWRST	1	R/W	While this bit is set to 0, the SCUX internal circuit is reset. Registers except this register are reset. Therefore, they should be set again after the reset is canceled. 0: The SCUX is reset. 1: The SCUX is operating.

chanhvinh.com

37.3.64 DMA Control Register (DMACR_CIM)

DMACR_CIM is a 32-bit readable/writable register that controls DMA execution.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	DMAM DFFU3	DMAM DFFU2	DMAM DFFU1	DMAM DFFU0	DMAM DFFD3	DMAM DFFD2	DMAM DFFD1	DMAM DFFD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	—	All 0	R/W	Reserved The write value should always be 0.
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	DMAMDFFU3	0	R/W	FFU0_3 DMA Mode 0: Disables DMA transfer to DMATD3_CIM. 1: Enables DMA transfer to DMATD3_CIM. Note: When DMA transfer is enabled, disable packet requests sent to the interrupt controller. (Set the UEVMR_FFU0_3.UEVMRQ bit to 0.)
6	DMAMDFFU2	0	R/W	FFU0_2 DMA Mode 0: Disables DMA transfer to DMATD2_CIM. 1: Enables DMA transfer to DMATD2_CIM. Note: When DMA transfer is enabled, disable packet requests sent to the interrupt controller. (Set the UEVMR_FFU0_2.UEVMRQ bit to 0.)
5	DMAMDFFU1	0	R/W	FFU0_1 DMA Mode 0: Disables DMA transfer to DMATD1_CIM. 1: Enables DMA transfer to DMATD1_CIM. Note: When DMA transfer is enabled, disable packet requests sent to the interrupt controller. (Set the UEVMR_FFU0_1.UEVMRQ bit to 0.)
4	DMAMDFFU0	0	R/W	FFU0_0 DMA Mode 0: Disables DMA transfer to DMATD0_CIM. 1: Enables DMA transfer to DMATD0_CIM. Note: When DMA transfer is enabled, disable packet requests sent to the interrupt controller. (Set the UEVMR_FFU0_0.UEVMRQ bit to 0.)
3	DMAMDFFD3	0	R/W	FFD0_3 DMA Mode 0: Disables DMA transfer to DMATD3_CIM. 1: Enables DMA transfer to DMATD3_CIM. Note: When DMA transfer is enabled, disable packet requests sent to the interrupt controller. (Set the DEVMR_FFD0_3.DEVMRQ bit to 0.)
2	DMAMDFFD2	0	R/W	FFD0_2 DMA Mode 0: Disables DMA transfer to DMATD2_CIM. 1: Enables DMA transfer to DMATD2_CIM. Note: When DMA transfer is enabled, disable packet requests sent to the interrupt controller. (Set the DEVMR_FFD0_2.DEVMRQ bit to 0.)
1	DMAMDFFD1	0	R/W	FFD0_1 DMA Mode 0: Disables DMA transfer to DMATD1_CIM. 1: Enables DMA transfer to DMATD1_CIM. Note: When DMA transfer is enabled, disable packet requests sent to the interrupt controller. (Set the DEVMR_FFD0_1.DEVMRQ bit to 0.)
0	DMAMDFFD0	0	R/W	FFD0_0 DMA Mode 0: Disables DMA transfer to DMATD0_CIM. 1: Enables DMA transfer to DMATD0_CIM. Note: When DMA transfer is enabled, disable packet requests sent to the interrupt controller. (Set the DEVMR_FFD0_0.DEVMRQ bit to 0.)

chanhvinh.com

37.3.65 DMA Transfer Register for FFD0_n (DMATDn_CIM) (n = 0, 1, 2, 3)

DMATDn_CIM is a 32-bit register that stores the data to be transmitted to FFD0_n. The data written to DMATDn_CIM is automatically stored in FFD0_n. When the number of idle data bytes in FFD0_n becomes equal to or greater than the value set in DRQSR_FFD0_n.SIZE, a data transfer request is issued. When the DMA controller is used (DMACR_CIM.DMAMDFFDn = 1), a transfer request for the number of counts set in DRQSR_FFD0_n.SIZE is sent to the DMA controller. When the interrupt controller is used (DEVMR_FFD0_n.DEVMRQ = 1), data is written in this register for the number of counts set in DRQSR_FFD0_n.SIZE.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

In the SCUX internal block, audio data is handled in the 24-bit big endian format. When this register is accessed in 32-bit units, the lower eight bits are not used and only the upper 24 bits are used, as shown in Figure 37.2. When the SCUX is used in 16-bit mode, the valid data needs to be located at the MSB of audio data. Therefore, when this register is accessed in 16-bit units, the written data is located in the upper 16 bits of audio data and the lower eight bits are fixed to 0 in the CIM block. When the SCUX is used in 16-bit mode and this register is accessed in 32-bit units, since bits [15:8] of the data to be written is passed to the FFD block without change, 0 should be written to bits [15:8].

DMATD	[31:24]	[23:16]	[15:8]	[7:0]
	MSB ↓	↓	↓	LSB
SCUX audio data	[23:16]	[15:8]	[7:0]	Not used
16-bit mode, 16-bit access to DMATD				
SCUX audio data	Write data [15:8]	Write data [7:0]	Clamped to 0 in CIM	
16-bit mode, 32-bit access to DMATD				
SCUX audio data	Write data [31:24] Valid data	Write data [23:16] Valid data	Write data [15:8] Write 0 to these bits	

Figure 37.2 Data Alignment of DMATDn_CIM

chanhvinh.com

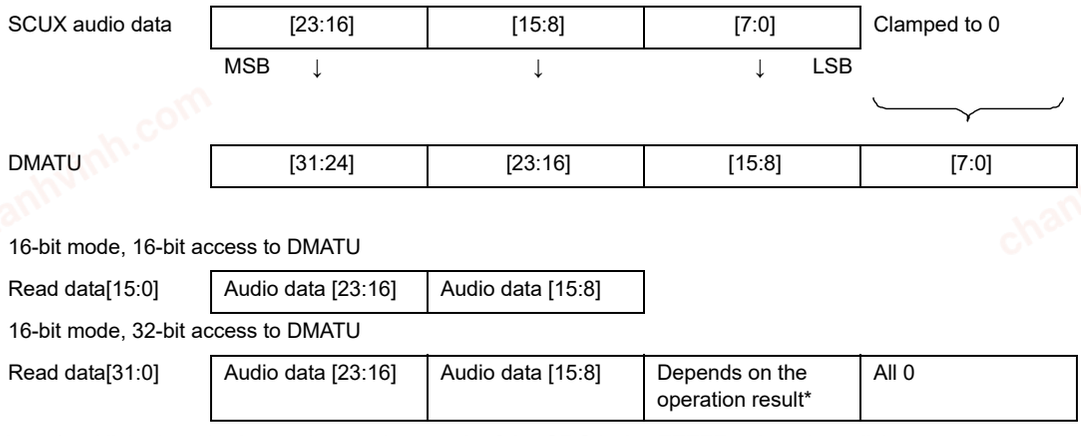
37.3.66 DMA Transfer Register for FFU0_n (DMATUn_CIM) (n = 0, 1, 2, 3)

DMATUn_CIM is a 32-bit register that stores the data received from FFU0_n. When data is read from DMATUn_CIM, the next data is automatically transferred from FFU0_n. When the number of data bytes in FFU0_n becomes equal to or greater than the value set in URQSR_FFU0_n.SIZE, a data transfer request is issued. When the DMA controller is used (DMACR_CIM.DMAMDFFUn = 1), a transfer request for the number of counts set in URQSR_FFU0_n.SIZE is sent to the DMA controller. When the interrupt controller is used (UEVMR_FFU0_n.UEVMRQ = 1), data is read from this register for the number of counts set in URQSR_FFU0_n.SIZE.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

When this register is accessed in 32-bit units, audio data is located in the upper 24 bits and the lower eight bits are always read as 0, as shown in Figure 37.3. When this register is accessed in 16-bit units, the upper 16 bits of audio data are read as valid data. When the SCUX is used in 16-bit mode and this register is accessed in 32-bit units, the lower eight bits are always read as 0 and bits [15:8] of the read data are read as a value dependent on the internal operation result in the SRC.



Note: * All 1 when the internal operation result of the SRC is 0x7FFFFF; otherwise, all 0.
Even when used in 16-bit mode, operation is performed in 24-bit units.

Figure 37.3 Data Alignment of DMATUn_CIM

chanhvinh.com

37.3.67 SSI Route Select Register (SSIRSEL_CIM)

SSIRSEL_CIM is a 32-bit readable/writable register that selects the SSIF channel to be connected to each SRC during direct transfer with the SSIF module. This register setting has no meaning when a route between the FFD or FFU module is selected. The IPSLR_IPC0_n and OPSLR_OPC0_n registers are used to select connection between the SSIF or FFD/FFU module. The DVUBR_DVU0_n register is used to specify whether the DVU is used or not.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SISEL3		SISEL2		SISEL1		SISEL0		-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	SOSEL3		-	-	SOSEL2		-	-	SOSEL1		-	-	SOSEL0	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	SISEL3	00	R/W	SRC3 SSIF Input Select These bits select the input route of SRC3. 00: Uses the input from SSIF2. 01, 10, 11: Setting prohibited Note: When the route from the FFD module is selected by the IPSLR_IPC0_3 register, this bit setting has no meaning.
29, 28	SISEL2	00	R/W	SRC2 SSIF Input Select These bits select the input route of SRC2. 00: Uses the input from SSIF1. 01, 10, 11: Setting prohibited Note: When the route from the FFD module is selected by the IPSLR_IPC0_2 register, this bit setting has no meaning.
27, 26	SISEL1	00	R/W	SRC1 SSIF Input Select These bits select the input route of SRC1. 00: Uses the input from SSIF3. 01, 10, 11: Setting prohibited Note: When the route from the FFD module is selected by the IPSLR_IPC0_1 register, this bit setting has no meaning.
25, 24	SISEL0	00	R/W	SRC0 SSIF Input Select These bits select the input route of SRC0. 00: Uses the input from SSIF0. 01: Uses the input from SSIF012.* 10, 11: Setting prohibited Note: When the route from the FFD module is selected by the IPSLR_IPC0_0 register, this bit setting has no meaning. * These bits can be set to 01 only when the SSI1PMD and SSI2PMD bits of the SSIPMD_CIM register are both set to 01 or 10.
23, 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21, 20	—	All 0	R/W	Reserved The write value should always be 0.
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	—	All 0	R/W	Reserved The write value should always be 0.
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
13, 12	SOSEL3	00	R/W	SSIF3 Output Select These bits select the output route to SSIF3. 00: Uses the output from SRC1 (DVU0_1).*1 01: Uses the output from SRC0 (DVU0_0).*2 10: Uses the output from MIX. 11: Setting prohibited Notes: 1. When the route to the FFU module is selected by the OPSLR_OPC0_1 register, this bit setting has no meaning. 2. When the route to the FFU module is selected by the OPSLR_OPC0_0 register, this bit setting has no meaning.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	SOSEL2	00	R/W	SSIF2 Output Select These bits select the output route to SSIF2. 00: Uses the output from SRC3 (DVU0_3).*1 01: Uses the output from SRC0 (DVU0_0).*2, *4 10: Uses the output from SRC1 (DVU0_1).*3, *4 11: Uses the output from MIX.*4 Notes: 1. When the route to the FFU module is selected by the OPSLR_OPC0_3 register, this bit setting has no meaning. 2. When the route to the FFU module is selected by the OPSLR_OPC0_0 register, this bit setting has no meaning. 3. When the route to the FFU module is selected by the OPSLR_OPC0_1 register, this bit setting has no meaning. 4. Valid only when SSIPMD_CIM.SSI012EN = 1.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	SOSEL1	00	R/W	SSIF1 Output Select These bits select the output route to SSIF1. 00: Uses the output from SRC2 (DVU0_2).*1 01: Uses the output from SRC0 (DVU0_0).*2, *4 10: Uses the output from SRC1 (DVU0_1).*3, *4 11: Uses the output from MIX.*4 Notes: 1. When the route to the FFU module is selected by the OPSLR_OPC0_2 register, this bit setting has no meaning. 2. When the route to the FFU module is selected by the OPSLR_OPC0_0 register, this bit setting has no meaning. 3. When the route to the FFU module is selected by the OPSLR_OPC0_1 register, this bit setting has no meaning. 4. Valid only when SSIPMD_CIM.SSI012EN = 1.
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	SOSEL0	00	R/W	SSIF0 Output Select These bits select the output route to SSIF0. 00: Uses the output from SRC0 (DVU0_0).*1 01: Uses the output from SRC1 (DVU0_1).*2 10: Uses the output from MIX. 11: Setting prohibited. Notes: 1. When the route to the FFU module is selected by the OPSLR_OPC0_0 register, this bit setting has no meaning. 2. When the route to the FFU module is selected by the OPSLR_OPC0_1 register, this bit setting has no meaning.

chanhvinh.com

37.3.68 FFD0_n Timing Select Register (FDTSELn_CIM) (n = 0, 1, 2, 3)

FDTSELn_CIM is a 32-bit readable/writable register that selects the input timing signal to be used in SRCn (n = 0, 1, 2, 3) when asynchronous mode is selected. This register setting is used only when the route with the FFD module is selected.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	SCKDIV										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	DIVEN	-	-	-	MTUSEL	SCKSEL			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	SCKDIV	All 0	R/W	Division Ratio These bits specify the ratio for dividing the input clock specified in the SCKSEL bits to generate the timing signal. Note: Always write 0 to bit 16. (Writing 1 to bit 16 is ignored.) When all of these bits are set to 0, the division ratio is 1. Only the input clock selected by SCKSEL[3] = 0 will be divided. The WS signal selected by SCKSEL[3] = 1 will not be divided. The DIVEN bit must be set to 0 before changing these bits.
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	DIVEN	0	R/W	Division Enable Frequency division starts the moment 1 is written to this bit. 1: Starts frequency division. 0: Outputs 0.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	MTUSEL	0	R/W	MTU Select This bit selects the clock signal specified by setting the SCKSEL bits to 0101 from the multi-function timer pulse unit 2. 0: Uses TIOC3A. 1: Uses TIOC4A
3 to 0	SCKSEL	All 0	R/W	Clock Select These bits select the clock that is used in generating the timing signal. 0000: Uses AUDIO_CLK. 0001: Uses the AUDIO_X1 input. 0010: Uses MLB_CLK. 0011: Uses USB_X1. 0100: Uses the peripheral clock 1 (P1ϕ) divided into 2. 0101: Uses the signal specified by the MTUSEL bit. 0110 and 0111: Setting prohibited 1000: Uses the WS signal of SSIF0. 1001: Uses the WS signal of SSIF1. 1010: Uses the WS signal of SSIF2. 1011: Uses the WS signal of SSIF3. 1100 to 1111: Setting prohibited Note: The DIVEN bit must be set to 0 before changing these bits.

chanhvinh.com

The input timing signal selector has the configuration shown in Figure 37.4.

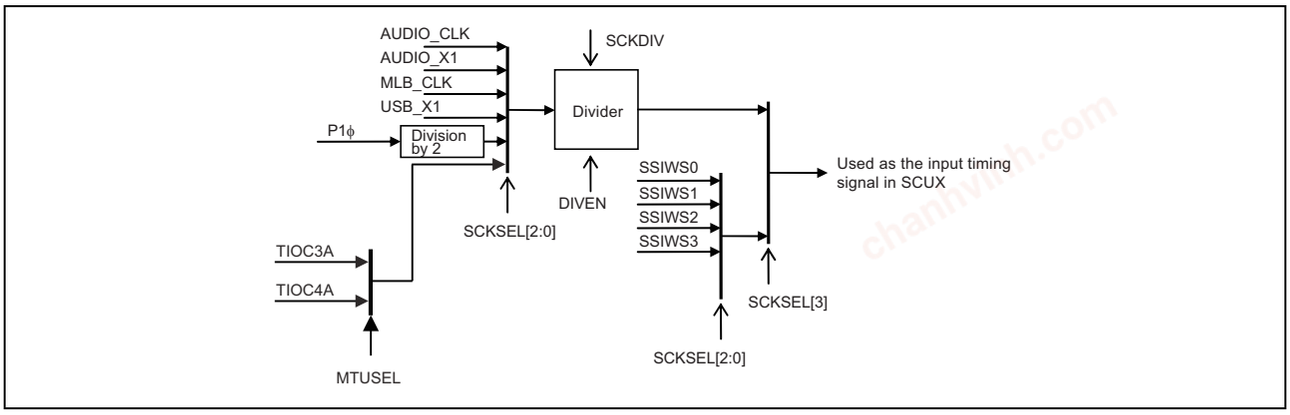


Figure 37.4 Configuration Diagram of Input Timing Signal Selector

chanhvinh.com

37.3.69 FFU0_n Timing Select Register (FDTSELn_CIM) (n = 0, 1, 2, 3)

FUTSELn_CIM is a 32-bit writable register that selects the output timing signal to be used in SRCn (n = 0, 1, 2, 3) when asynchronous mode is selected. This register setting is used when the route with the FFU module is selected or when output to the MIX is selected.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	SCKDIV										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	-	-	-	-	-	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	DIVEN	-	-	-	MTUSEL	SCKSEL			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	-	-	-	-	-	-	-	W	-	-	-	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	—	Reserved The write value should always be 0.
26 to 16	SCKDIV	All 0	W	Division Ratio These bits specify the ratio for dividing the input clock specified in the SCKSEL bits to generate the timing signal. Note: Always write 0 in bit 16. (Writing 1 to bit 16 is ignored.) When all of these bits are set to 0, the division ratio is 1. Only the input clock selected by SCKSEL[3] = 0 will be divided. The WS signal selected by SCKSEL[3] = 1 will not be divided. The DIVEN bit must be set to 0 before changing these bits.
15 to 9	—	All 0	—	Reserved The write value should always be 0.
8	DIVEN	0	W	Division Enable Frequency division starts the moment 1 is written to this bit. 1: Starts frequency division. 0: Outputs 0.
7 to 5	—	All 0	—	Reserved The write value should always be 0.
4	MTUSEL	0	W	MTU Select This bit selects the clock signal specified by setting the SCKSEL bits to 0101 from the multi-function timer pulse unit 2. 0: Uses TIOC3A. 1: Uses TIOC4A.
3 to 0	SCKSEL	All 0	W	Clock Select These bits select the clock that is used in generating the timing signal. 0000: Uses AUDIO_CLK. 0001: Uses AUDIO_X1 input. 0010: Uses MLB_CLK. 0011: Uses USB_X1. 0100: Uses the peripheral clock 1 (P1ϕ) divided into 2. 0101: Uses the signal specified by the MTUSEL bit. 0110 and 0111: Setting prohibited 1000: Uses the WS signal of SSIF0. 1001: Uses the WS signal of SSIF1. 1010: Uses the WS signal of SSIF2. 1011: Uses the WS signal of SSIF3. 1100 to 1111: Setting prohibited Note: The DIVEN bit must be set to 0 before changing these bits.

chanhvinh.com

The output timing signal selector has the configuration shown in Figure 37.5.

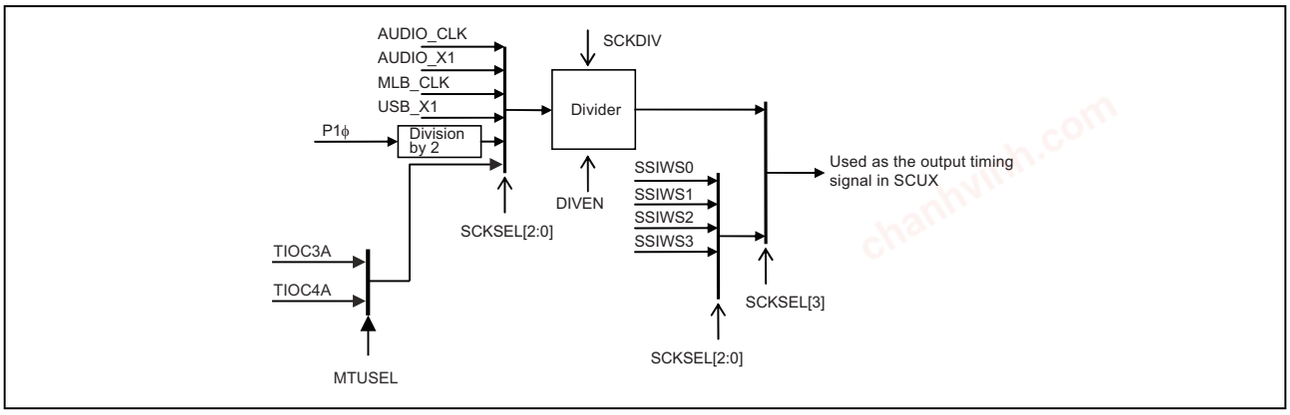


Figure 37.5 Configuration Diagram of Output Timing Signal Selector

chanhvinh.com

37.3.70 SSI Pin Mode Register (SSIPMD_CIM)

SSIPMD_CIM is a 32-bit readable/writable register that sets the pin mode of the SSIF.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	SSI3 CKS	SSI2 CKS	SSI1 CKS	SSI0 CKS
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SSI3PMD		-	-	-	-	-	-	-	-	-	SSI 012EN	SSI2PMD		SSI1PMD	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21, 20	—	All 0	R/W	Reserved The write value should always be 0.
19	SSI3CKS	0	R/W	SSIF3 Clock Select This bit selects the clock to be connected to the AUDIO_CLK pin of the SSIF3 module. 0: Selects the AUDIO_CLK input. 1: Selects the MLB_CLK input. Note: When selecting AUDIO_CLK or MLB_CLK as the oversampling clock of the SSIF3 module, SSICR3.CKS = 1 must also be set in the SSIF3 module.
18	SSI2CKS	0	R/W	SSIF2 Clock Select This bit selects the clock to be connected to the AUDIO_CLK pin of the SSIF2 module. 0: Selects the AUDIO_CLK input. 1: Selects the MLB_CLK input. Note: When selecting AUDIO_CLK or MLB_CLK as the oversampling clock of the SSIF2 module, SSICR2.CKS = 1 must also be set in the SSIF2 module.
17	SSI1CKS	0	R/W	SSIF1 Clock Select This bit selects the clock to be connected to the AUDIO_CLK pin of the SSIF1 module. 0: Selects the AUDIO_CLK input. 1: Selects the MLB_CLK input. Note: When selecting AUDIO_CLK or MLB_CLK as the oversampling clock of the SSIF1 module, SSICR1.CKS = 1 must also be set in the SSIF1 module.
16	SSI0CKS	0	R/W	SSIF0 Clock Select This bit selects the clock to be connected to the AUDIO_CLK pin of the SSIF0 module. 0: Selects the AUDIO_CLK input. 1: Selects the MLB_CLK input. Note: When selecting AUDIO_CLK or MLB_CLK as the oversampling clock of the SSIF0 module, SSICR0.CKS = 1 must also be set in the SSIF0 module.
15, 14	SSI3PMD	00	R/W	SSIF3 Pin Mode These bits select connection of the SSISCK3 and SSIWS3 pins. 00: The pins are used independently. 01: The SSIF0 pins are used in common. Both SSIF0 and SSIF3 are slaves. 10: The SSIF0 pins are used in common. SSIF0 is the master and SSIF3 is the slave. 11: Setting prohibited
13	—	All 0	R/W	Reserved This bits is always read as 0. The write value should always be 0.
12 to 8	—	All 0	R/W	Reserved The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SSI012EN	0	R/W	SSIF012 Enable This bit selects whether the three modules SSIF0, SSIF1, and SSIF2 are used to provide six channels at transmission. 0: Not handled as six channels. 1: Handled as six channels. Note: This bit can be set to 1 only when the SSI1PMD and SSI2PMD bits of this register are both set to 01 or 10. Usage of six channels at reception can be specified by SSIRSEL_CIM.SISEL0.
3, 2	SSI2PMD	00	R/W	SSIF2 Pin Mode These bits select connection of the SSISCK2 and SSIWS2 pins. 00: The pins are used independently. 01: The SSIF0 pins are used in common. Both SSIF0 and SSIF2 are slaves. 10: The SSIF0 pins are used in common. SSIF0 is the master and SSIF2 is the slave. 11: Setting prohibited
1, 0	SSI1PMD	00	R/W	SSIF1 Pin Mode These bits select connection of the SSISCK1 and SSIWS1 pins. 00: The pins are used independently. 01: The SSIF0 pins are used in common. Both SSIF0 and SSIF1 are slaves. 10: The SSIF0 pins are used in common. SSIF0 is the master and SSIF1 is the slave. 11: Setting prohibited

chanhvinh.com

37.3.71 SSI Control Register (SSICTRL_CIM)

SSIPCTRL_CIM is a 32-bit readable/writable register that controls connection with the SSIF module and starting and stopping of the module.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	SSI3TX	-	-	-	SSI3RX	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	-	W	W	W	-	W	W	W	-	-	-	-	-	-	-	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	SSI0TX	SSI1TX	SSI2TX	-	SSI0RX	SSI1RX	SSI2RX	-	-	-	-	-	-	SSI 012TEN	SSI 012REN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	-	W	W	W	-	W	W	W	-	-	-	-	-	-	W	W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	—	Reserved The write value should always be 0.
30	SSI3TX	0	W	SSIF3 Direct Transmission 0: Disables direct transmission to SSIF3. 1: Enables direct transmission to SSIF3.
29, 28	—	All 0	W	Reserved The write value should always be 0.
27	—	0	—	Reserved The write value should always be 0.
26	SSI3RX	0	W	SSIF3 Direct Reception 0: Disables direct reception from SSIF3. 1: Enables direct reception from SSIF3.
25, 24	—	All 0	W	Reserved The write value should always be 0.
23 to 18	—	All 0	—	Reserved The write value should always be 0.
17, 16	—	All 0	W	Reserved The write value should always be 0.
15	—	0	—	Reserved The write value should always be 0.
14	SSI0TX	0	W	SSIF0 Direct Transmission 0: Disables direct transmission to SSIF0. 1: Enables direct transmission to SSIF0.
13	SSI1TX	0	W	SSIF1 Direct Transmission 0: Disables direct transmission to SSIF1. 1: Enables direct transmission to SSIF1.
12	SSI2TX	0	W	SSIF2 Direct Transmission 0: Disables direct transmission to SSIF2. 1: Enables direct transmission to SSIF2.
11	—	0	—	Reserved The write value should always be 0.
10	SSI0RX	0	W	SSIF0 Direct Reception 0: Disables direct reception from SSIF0. 1: Enables direct reception from SSIF0.
9	SSI1RX	0	W	SSIF1 Direct Reception 0: Disables direct reception from SSIF1. 1: Enables direct reception from SSIF1.
8	SSI2RX	0	W	SSIF2 Direct Reception 0: Disables direct reception from SSIF2. 1: Enables direct reception from SSIF2.
7 to 2	—	All 0	—	Reserved The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
1	SSI012TEN	0	W	SSIF012 Transmission Enable This bit specifies transmission in three modules SSIF0, SSIF1, and SSIF2 to be started or stopped simultaneously. 0: Stops transmission in SSIF0, SSIF1, and SSIF2. 1: Starts transmission in SSIF0, SSIF1, and SSIF2. Note: This function can be used only when SSIPMD_CIM.SSI012EN = 1. When SSIPMD_CIM.SSI012EN = 0, SSIF modules should be controlled using SSICRn.TEN. This bit must be set to 0 when performing direct transmission to SSIF0, SSIF1, or SSIF2.
0	SSI012REN	0	W	SSIF012 Reception Enable This bit specifies reception in three modules SSIF0, SSIF1, and SSIF2 to be started or stopped simultaneously. 0: Stops reception in SSIF0, SSIF1, and SSIF2. 1: Starts reception in SSIF0, SSIF1, and SSIF2. Note: This function can be used only when SSIRSEL_CIM.SISEL0 = 01. When SSIRSEL_CIM.SISEL0 = 00, SSIF modules should be controlled using SSICRn.REN. This bit must be set to 0 when performing direct reception from SSIF0, SSIF1, or SSIF2.

chanhvinh.com

37.3.72 SRCn Route Select Register (SRCRSELn_CIM) (n = 0, 1, 2, 3)

SRCRSEL_CIM is a 32-bit readable/writable register that sets the route for the data that is input to the SRC.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	PLACE7			-	PLACE6			-	PLACE5			-	PLACE4		
Initial value:	0	1	1	1	0	1	1	0	0	1	0	1	0	1	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PLACE3			-	PLACE2			-	PLACE1			-	PLACE0		
Initial value:	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 28	PLACE7	111	R/W	Place 7 Stream data is rearranged before it is input to the SRC. These bits select data on the input side to be output to place 7 on the output side. 000: Data at place 0 on the input side is output to place 7 on the output side. 001: Data at place 1 on the input side is output to place 7 on the output side. 010: Data at place 2 on the input side is output to place 7 on the output side. 011: Data at place 3 on the input side is output to place 7 on the output side. 100: Data at place 4 on the input side is output to place 7 on the output side. 101: Data at place 5 on the input side is output to place 7 on the output side. 110: Data at place 6 on the input side is output to place 7 on the output side. 111: Data at place 7 on the input side is output to place 7 on the output side. Note: This bit setting is used when eight channels are set. When six or less channels are set, set these bits to the same value as the initial value.
27	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
26 to 24	PLACE6	110	R/W	Place 6 Stream data is rearranged before it is input to the SRC. These bits select data on the input side to be output to place 6 on the output side. 000: Data at place 0 on the input side is output to place 6 on the output side. 001: Data at place 1 on the input side is output to place 6 on the output side. 010: Data at place 2 on the input side is output to place 6 on the output side. 011: Data at place 3 on the input side is output to place 6 on the output side. 100: Data at place 4 on the input side is output to place 6 on the output side. 101: Data at place 5 on the input side is output to place 6 on the output side. 110: Data at place 6 on the input side is output to place 6 on the output side. 111: Data at place 7 on the input side is output to place 6 on the output side. Note: This bit setting is used when eight channels are set. When six or less channels are set, set these bits to the same value as the initial value.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
23	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
22 to 20	PLACE5	101	R/W	Place 5 Stream data is rearranged before it is input to the SRC. These bits select data on the input side to be output to place 5 on the output side. 000: Data at place 0 on the input side is output to place 5 on the output side. 001: Data at place 1 on the input side is output to place 5 on the output side. 010: Data at place 2 on the input side is output to place 5 on the output side. 011: Data at place 3 on the input side is output to place 5 on the output side. 100: Data at place 4 on the input side is output to place 5 on the output side. 101: Data at place 5 on the input side is output to place 5 on the output side. 110: Data at place 6 on the input side is output to place 5 on the output side. 111: Data at place 7 on the input side is output to place 5 on the output side. Note: This bit setting is used when six or more channels are set. When four or less channels are set, set these bits to the same value as the initial value.
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18 to 16	PLACE4	100	R/W	Place 4 Stream data is rearranged before it is input to the SRC. These bits select data on the input side to be output to place 4 on the output side. 000: Data at place 0 on the input side is output to place 4 on the output side. 001: Data at place 1 on the input side is output to place 4 on the output side. 010: Data at place 2 on the input side is output to place 4 on the output side. 011: Data at place 3 on the input side is output to place 4 on the output side. 100: Data at place 4 on the input side is output to place 4 on the output side. 101: Data at place 5 on the input side is output to place 4 on the output side. 110: Data at place 6 on the input side is output to place 4 on the output side. 111: Data at place 7 on the input side is output to place 4 on the output side. Note: This bit setting is used when six or more channels are set. When four or less channels are set, set these bits to the same value as the initial value.
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
14 to 12	PLACE3	011	R/W	Place 3 Stream data is rearranged before it is input to the SRC. These bits select data on the input side to be output to place 3 on the output side. 000: Data at place 0 on the input side is output to place 3 on the output side. 001: Data at place 1 on the input side is output to place 3 on the output side. 010: Data at place 2 on the input side is output to place 3 on the output side. 011: Data at place 3 on the input side is output to place 3 on the output side. 100: Data at place 4 on the input side is output to place 3 on the output side. 101: Data at place 5 on the input side is output to place 3 on the output side. 110: Data at place 6 on the input side is output to place 3 on the output side. 111: Data at place 7 on the input side is output to place 3 on the output side. Note: This bit setting is used when four or more channels are set. When two or less channels are set, set these bits to the same value as the initial value.
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PLACE2	010	R/W	Place 2 Stream data is rearranged before it is input to the SRC. These bits select data on the input side to be output to place 2 on the output side. 000: Data at place 0 on the input side is output to place 2 on the output side. 001: Data at place 1 on the input side is output to place 2 on the output side. 010: Data at place 2 on the input side is output to place 2 on the output side. 011: Data at place 3 on the input side is output to place 2 on the output side. 100: Data at place 4 on the input side is output to place 2 on the output side. 101: Data at place 5 on the input side is output to place 2 on the output side. 110: Data at place 6 on the input side is output to place 2 on the output side. 111: Data at place 7 on the input side is output to place 2 on the output side. Note: This bit setting is used when four or more channels are set. When two or less channels are set, set these bits to the same value as the initial value.
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	PLACE1	001	R/W	Place 1 Stream data is rearranged before it is input to the SRC. These bits select data on the input side to be output to place 1 on the output side. 000: Data at place 0 on the input side is output to place 1 on the output side. 001: Data at place 1 on the input side is output to place 1 on the output side. 010: Data at place 2 on the input side is output to place 1 on the output side. 011: Data at place 3 on the input side is output to place 1 on the output side. 100: Data at place 4 on the input side is output to place 1 on the output side. 101: Data at place 5 on the input side is output to place 1 on the output side. 110: Data at place 6 on the input side is output to place 1 on the output side. 111: Data at place 7 on the input side is output to place 1 on the output side.
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PLACE0	000	R/W	Place 0 Stream data is rearranged before it is input to the SRC. These bits select data on the input side to be output to place 0 on the output side. 000: Data at place 0 on the input side is output to place 0 on the output side. 001: Data at place 1 on the input side is output to place 0 on the output side. 010: Data at place 2 on the input side is output to place 0 on the output side. 011: Data at place 3 on the input side is output to place 0 on the output side. 100: Data at place 4 on the input side is output to place 0 on the output side. 101: Data at place 5 on the input side is output to place 0 on the output side. 110: Data at place 6 on the input side is output to place 0 on the output side. 111: Data at place 7 on the input side is output to place 0 on the output side.

chanhvinh.com

37.3.73 MIX Route Select Register (MIXRSEL_CIM)

MIXRSEL_CIM is a 32-bit readable/writable register that sets the route for the data that is output from the MIX.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	PLACE7			-	PLACE6			-	PLACE5			-	PLACE4		
Initial value:	0	1	1	1	0	1	1	0	0	1	0	1	0	1	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PLACE3			-	PLACE2			-	PLACE1			-	PLACE0		
Initial value:	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 28	PLACE7	111	R/W	Place 7 Stream data is rearranged after it is output from the MIX. These bits select data on the input side to be output to place 7 on the output side. 000: Data at place 0 on the input side is output to place 7 on the output side. 001: Data at place 1 on the input side is output to place 7 on the output side. 010: Data at place 2 on the input side is output to place 7 on the output side. 011: Data at place 3 on the input side is output to place 7 on the output side. 100: Data at place 4 on the input side is output to place 7 on the output side. 101: Data at place 5 on the input side is output to place 7 on the output side. 110: Data at place 6 on the input side is output to place 7 on the output side. 111: Data at place 7 on the input side is output to place 7 on the output side. Note: This bit setting is used when eight channels are set. When six or less channels are set, set these bits to the same value as the initial value.
27	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
26 to 24	PLACE6	110	R/W	Place 6 Stream data is rearranged after it is output from the MIX. These bits select data on the input side to be output to place 6 on the output side. 000: Data at place 0 on the input side is output to place 6 on the output side. 001: Data at place 1 on the input side is output to place 6 on the output side. 010: Data at place 2 on the input side is output to place 6 on the output side. 011: Data at place 3 on the input side is output to place 6 on the output side. 100: Data at place 4 on the input side is output to place 6 on the output side. 101: Data at place 5 on the input side is output to place 6 on the output side. 110: Data at place 6 on the input side is output to place 6 on the output side. 111: Data at place 7 on the input side is output to place 6 on the output side. Note: This bit setting is used when eight channels are set. When six or less channels are set, set these bits to the same value as the initial value.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
23	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
22 to 20	PLACE5	101	R/W	Place 5 Stream data is rearranged after it is output from the MIX. These bits select data on the input side to be output to place 5 on the output side. 000: Data at place 0 on the input side is output to place 5 on the output side. 001: Data at place 1 on the input side is output to place 5 on the output side. 010: Data at place 2 on the input side is output to place 5 on the output side. 011: Data at place 3 on the input side is output to place 5 on the output side. 100: Data at place 4 on the input side is output to place 5 on the output side. 101: Data at place 5 on the input side is output to place 5 on the output side. 110: Data at place 6 on the input side is output to place 5 on the output side. 111: Data at place 7 on the input side is output to place 5 on the output side. Note: This bit setting is used when six or more channels are set. When four or less channels are set, set these bits to the same value as the initial value.
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18 to 16	PLACE4	100	R/W	Place 4 Stream data is rearranged after it is output from the MIX. These bits select data on the input side to be output to place 4 on the output side. 000: Data at place 0 on the input side is output to place 4 on the output side. 001: Data at place 1 on the input side is output to place 4 on the output side. 010: Data at place 2 on the input side is output to place 4 on the output side. 011: Data at place 3 on the input side is output to place 4 on the output side. 100: Data at place 4 on the input side is output to place 4 on the output side. 101: Data at place 5 on the input side is output to place 4 on the output side. 110: Data at place 6 on the input side is output to place 4 on the output side. 111: Data at place 7 on the input side is output to place 4 on the output side. Note: This bit setting is used when six or more channels are set. When four or less channels are set, set these bits to the same value as the initial value.
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
14 to 12	PLACE3	011	R/W	Place 3 Stream data is rearranged after it is output from the MIX. These bits select data on the input side to be output to place 3 on the output side. 000: Data at place 0 on the input side is output to place 3 on the output side. 001: Data at place 1 on the input side is output to place 3 on the output side. 010: Data at place 2 on the input side is output to place 3 on the output side. 011: Data at place 3 on the input side is output to place 3 on the output side. 100: Data at place 4 on the input side is output to place 3 on the output side. 101: Data at place 5 on the input side is output to place 3 on the output side. 110: Data at place 6 on the input side is output to place 3 on the output side. 111: Data at place 7 on the input side is output to place 3 on the output side. Note: This bit setting is used when four or more channels are set. When two or less channels are set, set these bits to the same value as the initial value.
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PLACE2	010	R/W	Place 2 Stream data is rearranged after it is output from the MIX. These bits select data on the input side to be output to place 2 on the output side. 000: Data at place 0 on the input side is output to place 2 on the output side. 001: Data at place 1 on the input side is output to place 2 on the output side. 010: Data at place 2 on the input side is output to place 2 on the output side. 011: Data at place 3 on the input side is output to place 2 on the output side. 100: Data at place 4 on the input side is output to place 2 on the output side. 101: Data at place 5 on the input side is output to place 2 on the output side. 110: Data at place 6 on the input side is output to place 2 on the output side. 111: Data at place 7 on the input side is output to place 2 on the output side. Note: This bit setting is used when four or more channels are set. When two or less channels are set, set these bits to the same value as the initial value.
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	PLACE1	001	R/W	Place 1 Stream data is rearranged after it is output from the MIX. These bits select data on the input side to be output to place 1 on the output side. 000: Data at place 0 on the input side is output to place 1 on the output side. 001: Data at place 1 on the input side is output to place 1 on the output side. 010: Data at place 2 on the input side is output to place 1 on the output side. 011: Data at place 3 on the input side is output to place 1 on the output side. 100: Data at place 4 on the input side is output to place 1 on the output side. 101: Data at place 5 on the input side is output to place 1 on the output side. 110: Data at place 6 on the input side is output to place 1 on the output side. 111: Data at place 7 on the input side is output to place 1 on the output side.
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PLACE0	000	R/W	Place 0 Stream data is rearranged after it is output from the MIX. These bits select data on the input side to be output to place 0 on the output side. 000: Data at place 0 on the input side is output to place 0 on the output side. 001: Data at place 1 on the input side is output to place 0 on the output side. 010: Data at place 2 on the input side is output to place 0 on the output side. 011: Data at place 3 on the input side is output to place 0 on the output side. 100: Data at place 4 on the input side is output to place 0 on the output side. 101: Data at place 5 on the input side is output to place 0 on the output side. 110: Data at place 6 on the input side is output to place 0 on the output side. 111: Data at place 7 on the input side is output to place 0 on the output side.

chanhvinh.com

37.4 Operation

37.4.1 Initial Setting Procedure

Figure 37.6 to Figure 37.8 show the initial setting procedure of the SCUX. For details on register setting, refer to section 37.3, Register Descriptions.

37.4.2 Transfer Start Procedure and Stop Procedure

Figure 37.9 to Figure 37.11 show the transfer start procedure and stop procedure of the SCUX. For details on register setting, refer to section 37.3, Register Descriptions.

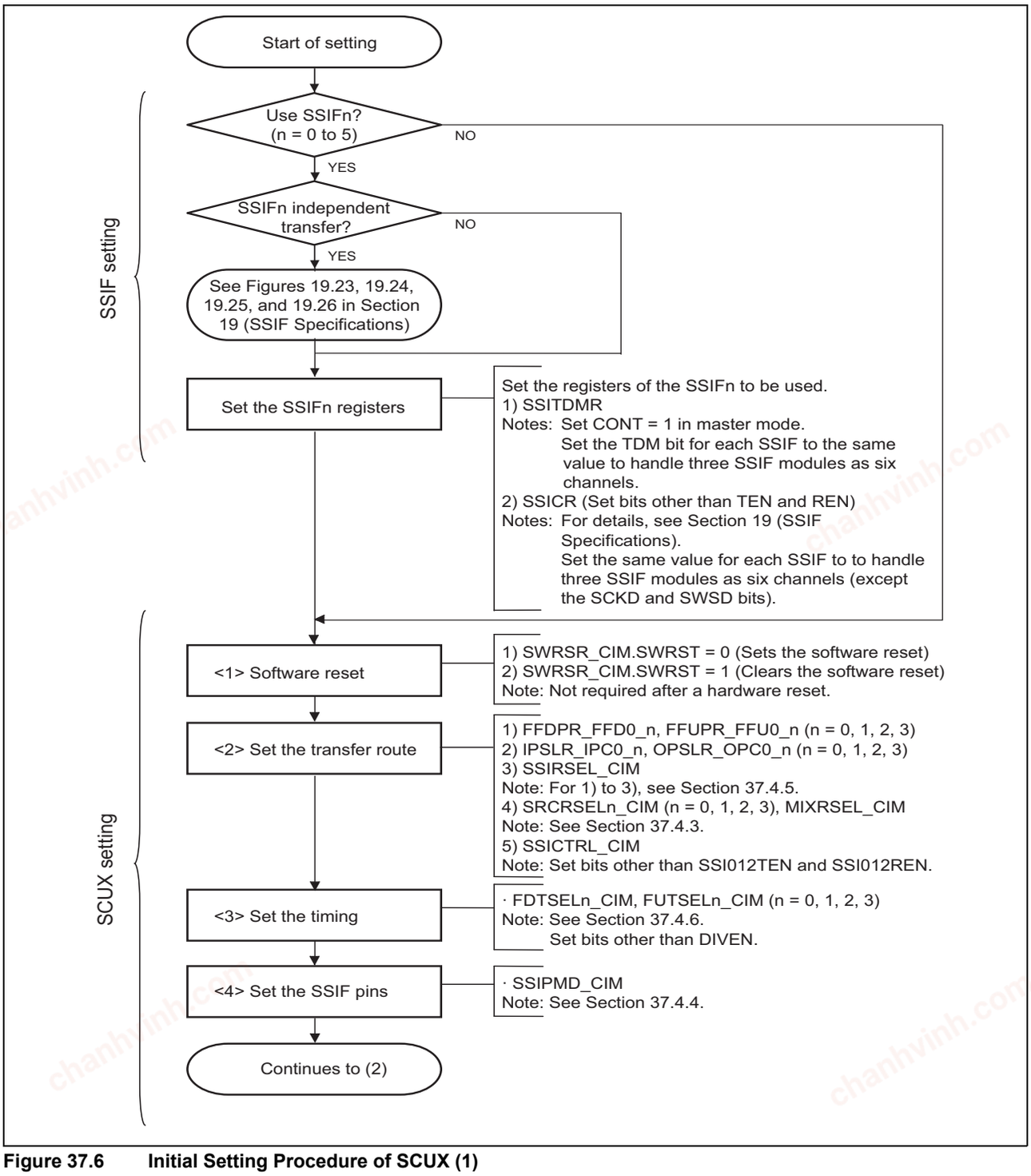


Figure 37.6 Initial Setting Procedure of SCUX (1)

chanhvinh.com

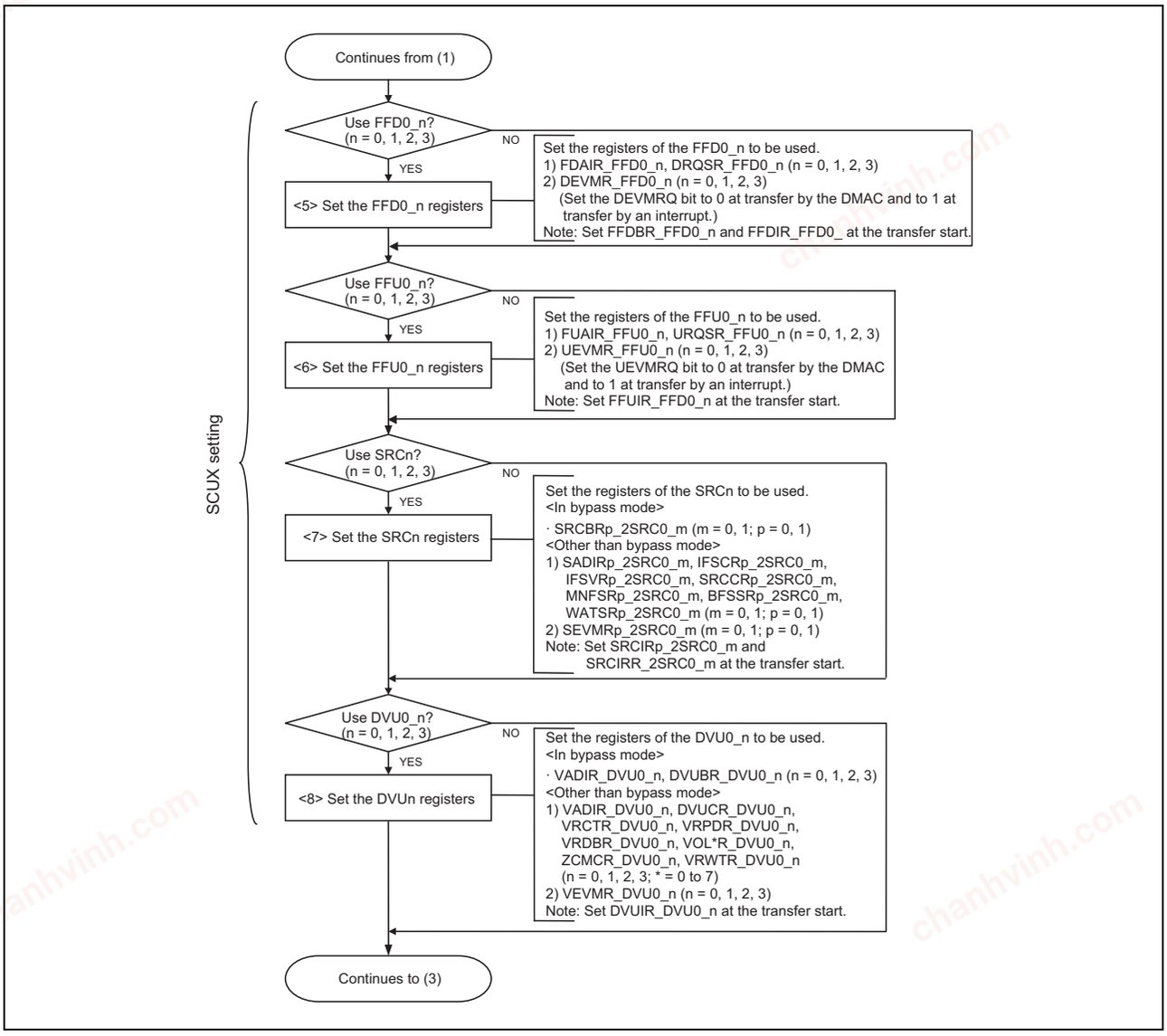


Figure 37.7 Initial Setting Procedure of SCUX (2)

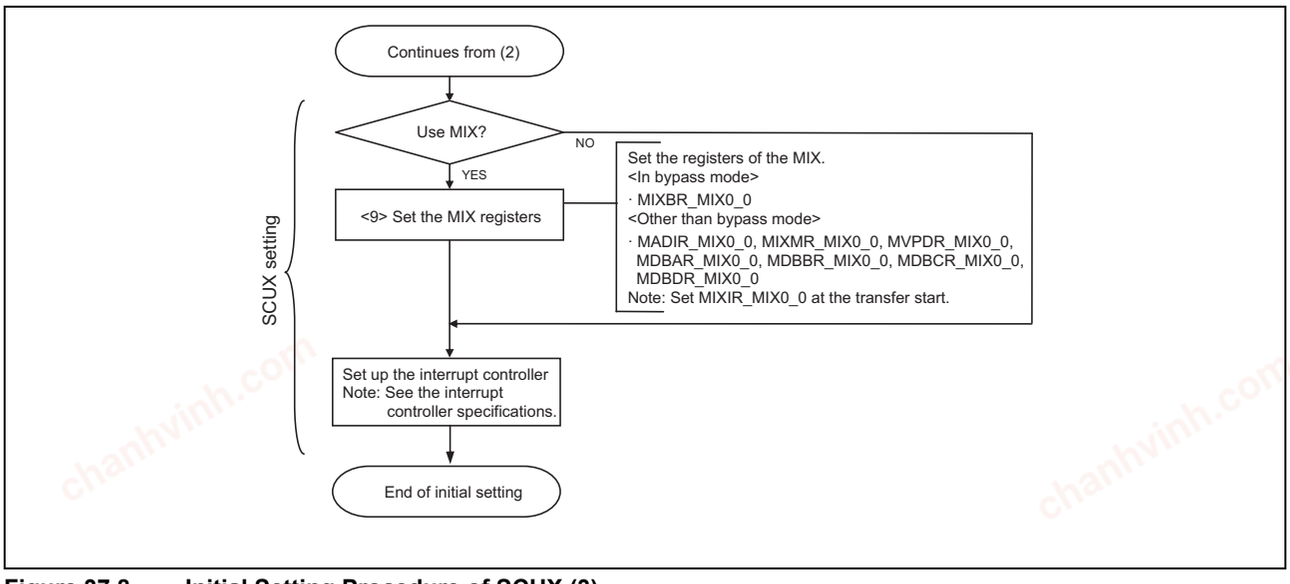


Figure 37.8 Initial Setting Procedure of SCUX (3)

chanhvinh.com

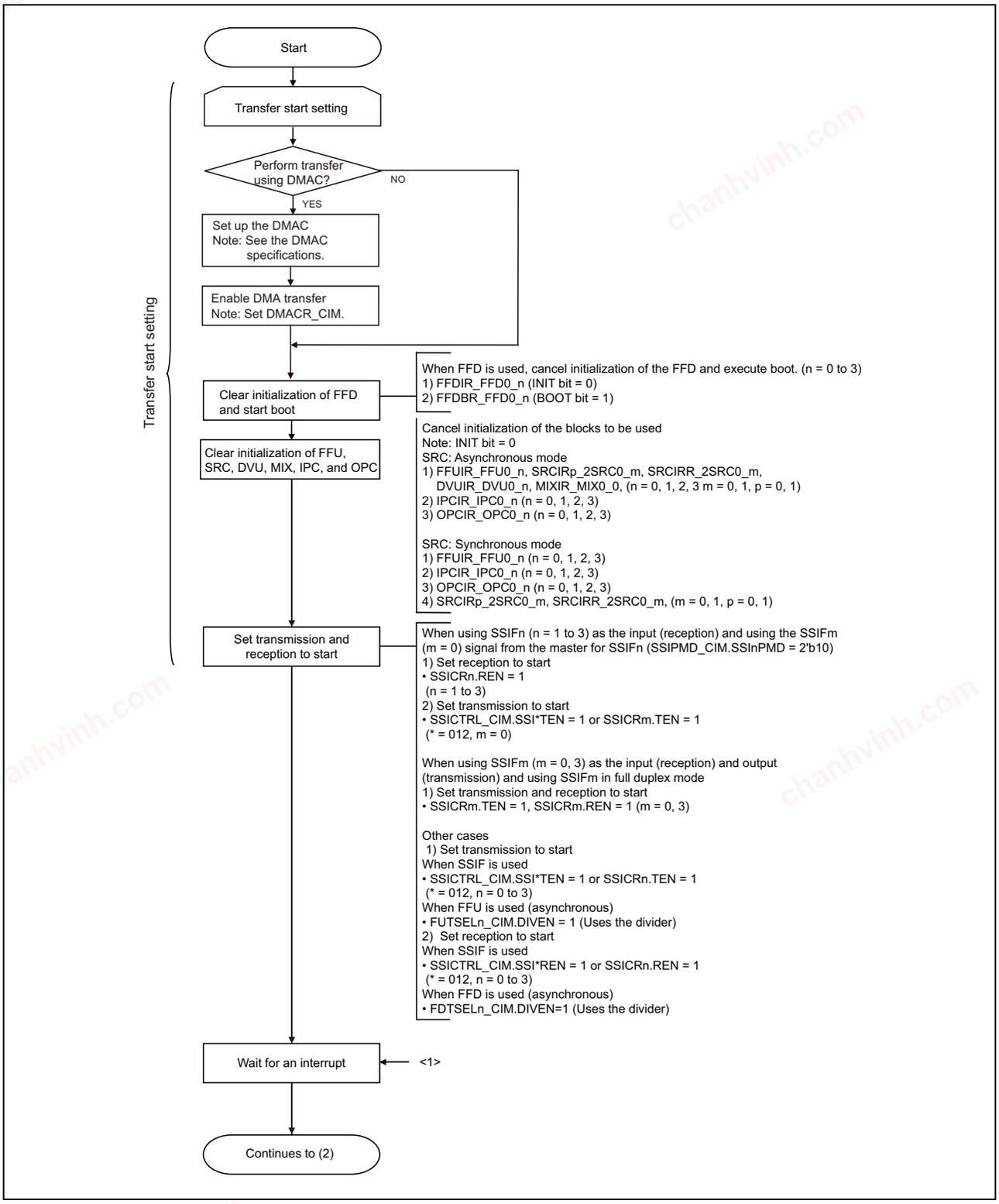


Figure 37.9 Transfer Start Procedure and Stop Procedure of SCUX (1)

chanhvinh.com

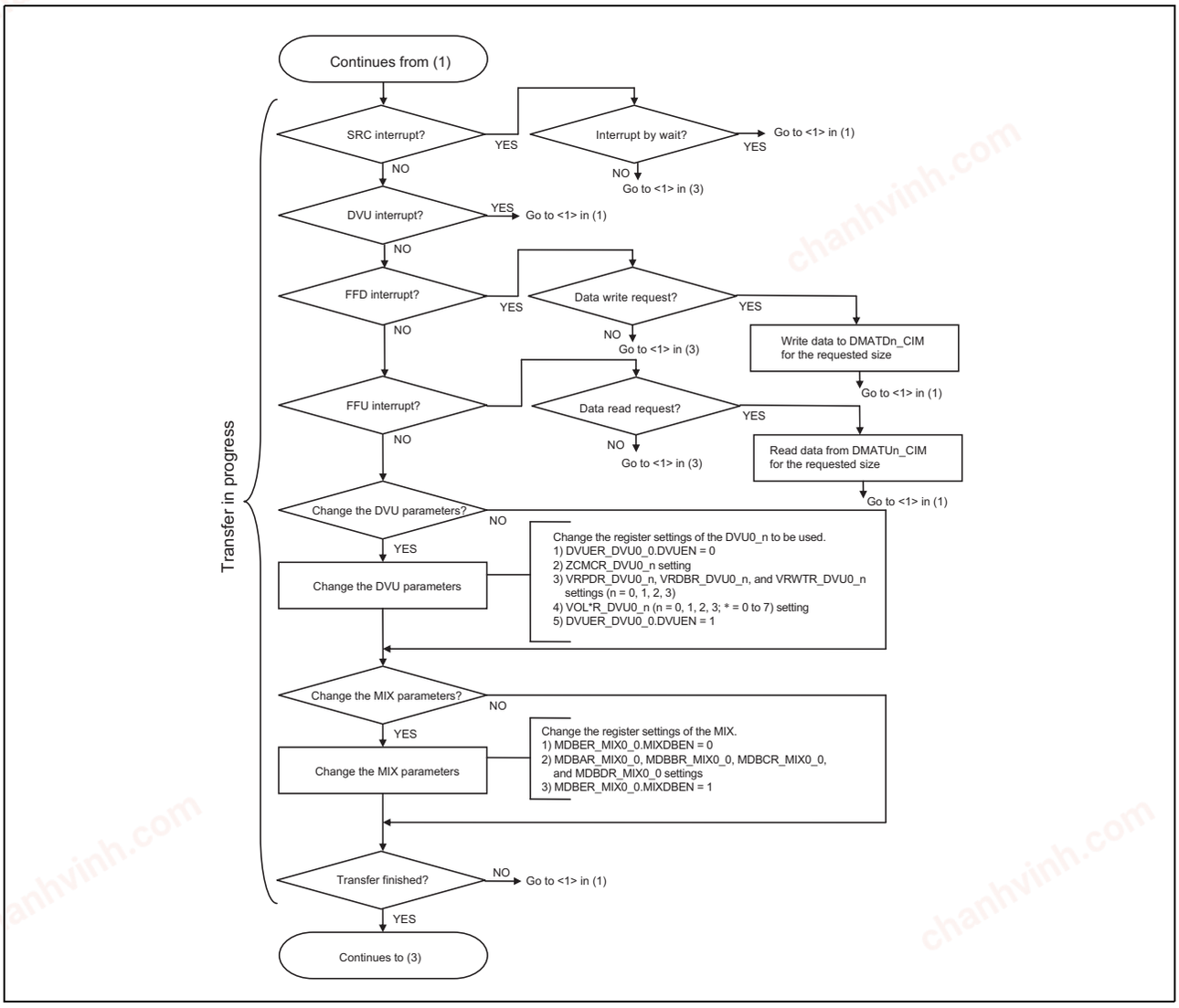
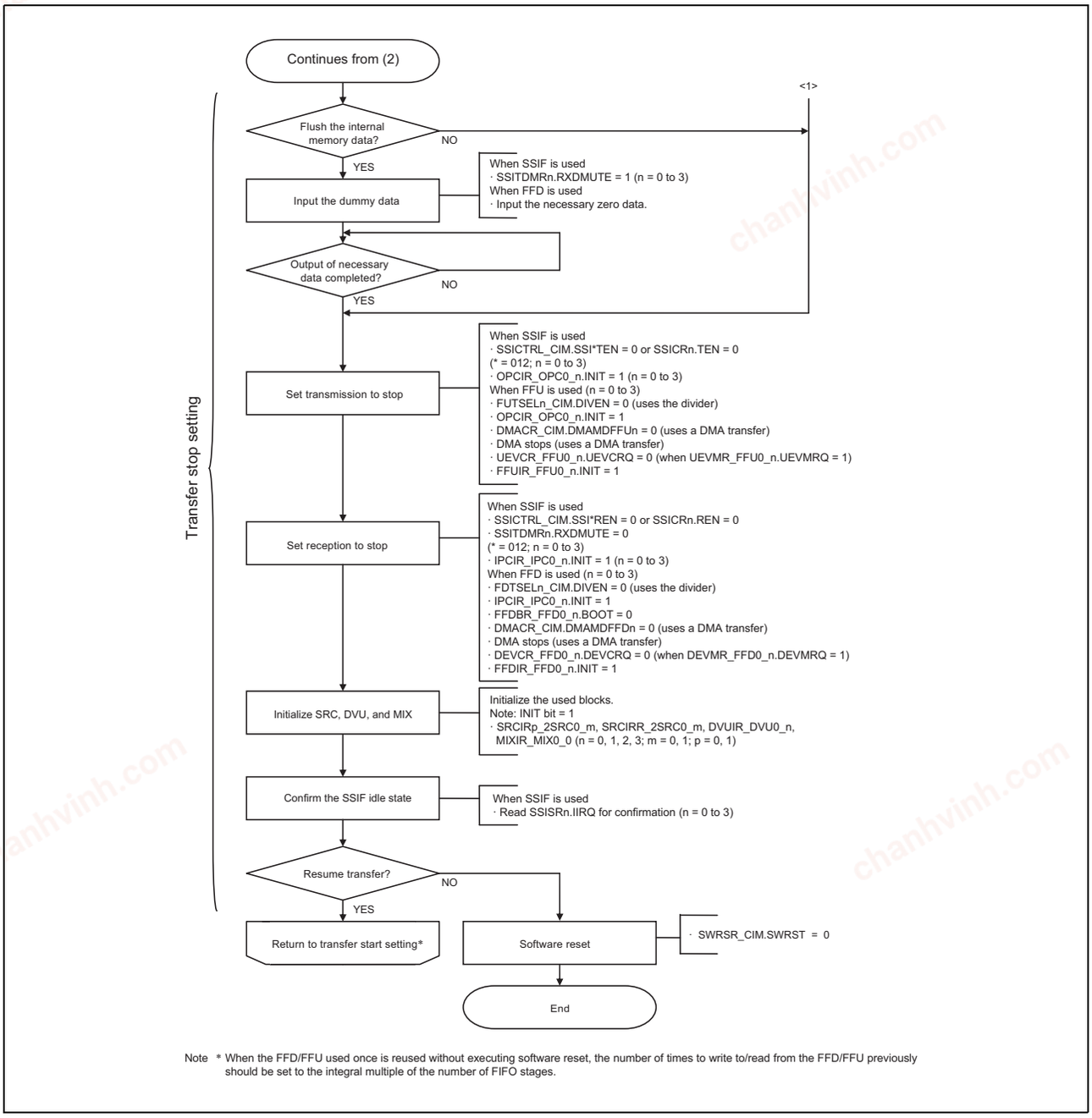


Figure 37.10 Transfer Start Procedure and Stop Procedure of SCUX (2)

chanhvinh.com



Note * When the FFD/FFU used once is reused without executing software reset, the number of times to write to/read from the FFD/FFU previously should be set to the integral multiple of the number of FIFO stages.

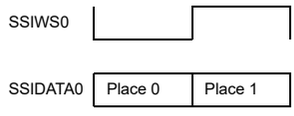
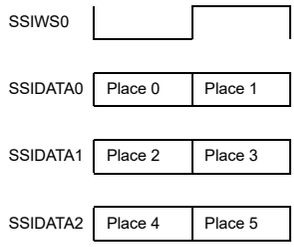
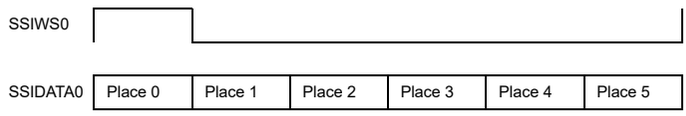
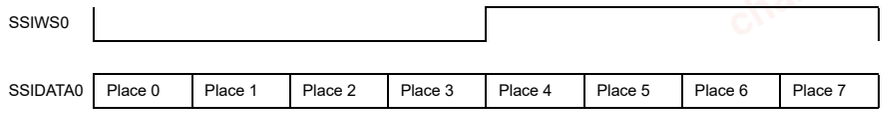
Figure 37.11 Transfer Start Procedure and Stop Procedure of SCUX (3)

chanhvinh.com

37.4.3 Data Rearrangement for Each Channel

Data can be rearranged independently for each channel in the SCUX. Data rearrangement can be performed immediately before input to the SRC and immediately after output from the MIX. Table 37.9 to Table 37.11 show definition of data places. Data rearrangement immediately before the SRC is performed as described in section 37.3.72, SRCn Route Select Register (SRCRSELn_CIM) (n = 0, 1, 2, 3). Data rearrangement immediately after the MIX is performed as described in section 37.3.73, MIX Route Select Register (MIXRSEL_CIM).

Table 37.9 Definition of Data Places (1)

SSIF • Stereo (2 ch)	
SSIF • Stereo × 3 (6 ch)	
SSIF • TDM (6 ch)	
SSIF • Multichannel (8 ch)	

chanhvinh.com

Table 37.10 Definition of Data Places (2)

<table>
<tr><td>FFD0_n
• DMATDn_CIM
32-bit access (2 ch)</td><td>
<table>
<tr><th></th><th>31 8</th><th>7 0</th></tr>
<tr><td>1st write data</td><td>Place 0</td><td>0</td></tr>
<tr><td>2nd write data</td><td>Place 1</td><td>0</td></tr>
<tr><td>3rd write data</td><td>Place 0</td><td>0</td></tr>
<tr><td>4th write data</td><td>Place 1</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>FFD0_n
• DMATDn_CIM
16-bit access (2 ch)</td><td>
<table>
<tr><th></th><th>31 16</th><th>15 0</th></tr>
<tr><td>1st write data</td><td>Place 0</td><td>0</td></tr>
<tr><td>2nd write data</td><td>Place 1</td><td>0</td></tr>
<tr><td>3rd write data</td><td>Place 0</td><td>0</td></tr>
<tr><td>4th write data</td><td>Place 1</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>FFD0_n
• DMATDn_CIM 32-bit
access (8 ch)</td><td>
<table>
<tr><th></th><th>31 8</th><th>7 0</th></tr>
<tr><td>1st write data</td><td>Place 0</td><td>0</td></tr>
<tr><td>2nd write data</td><td>Place 1</td><td>0</td></tr>
<tr><td>3rd write data</td><td>Place 2</td><td>0</td></tr>
<tr><td>4th write data</td><td>Place 3</td><td>0</td></tr>
<tr><td>5th write data</td><td>Place 4</td><td>0</td></tr>
<tr><td>6th write data</td><td>Place 5</td><td>0</td></tr>
<tr><td>7th write data</td><td>Place 6</td><td>0</td></tr>
<tr><td>8th write data</td><td>Place 7</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>FFU0_n
• DMATUn_CIM 32-bit
access (2 ch)</td><td>
<table>
<tr><th></th><th>31 8</th><th>7 0</th></tr>
<tr><td>1st read data</td><td>Place 0</td><td>0</td></tr>
<tr><td>2nd read data</td><td>Place 1</td><td>0</td></tr>
<tr><td>3rd read data</td><td>Place 0</td><td>0</td></tr>
<tr><td>4th read data</td><td>Place 1</td><td>0</td></tr>
</table>
</td></tr>
</table>

chanhvinh.com

Table 37.11 Definition of Data Places (3)

<table>
<tr>
<td>FFU0_n
• DMATUn_CIM 16-bit access (2 ch)</td>
<td>
<table>
<tr><th></th><th>31 16</th><th>15 0</th></tr>
<tr><td>1st read data</td><td>Place 0</td><td>-</td></tr>
<tr><td>2nd read data</td><td>Place 1</td><td>-</td></tr>
<tr><td>3rd read data</td><td>Place 0</td><td>-</td></tr>
<tr><td>4th read data</td><td>Place 1</td><td>-</td></tr>
</table>
</td>
</tr>
<tr>
<td>FFU0_n
• DMATUn_CIM 32-bit access (8 ch)</td>
<td>
<table>
<tr><th></th><th>31 8</th><th>7 0</th></tr>
<tr><td>1st read data</td><td>Place 0</td><td>0</td></tr>
<tr><td>2nd read data</td><td>Place 1</td><td>0</td></tr>
<tr><td>3rd read data</td><td>Place 2</td><td>0</td></tr>
<tr><td>4th read data</td><td>Place 3</td><td>0</td></tr>
<tr><td>5th read data</td><td>Place 4</td><td>0</td></tr>
<tr><td>6th read data</td><td>Place 5</td><td>0</td></tr>
<tr><td>7th read data</td><td>Place 6</td><td>0</td></tr>
<tr><td>8th read data</td><td>Place 7</td><td>0</td></tr>
</table>
</td>
</tr>
</table>

chanhvinh.com

37.4.4 Pin Connection Specifications of SSIF

SSIF0 to SSIF3 can be operated in synchronization with the same SSISCK and SSIWS signals. The AUDIO_CLK input to the SSIF can be selected between external pins AUDIO_CLK and MLB_CLK. These settings are made as described in section 37.3.70, SSI Pin Mode Register (SSIPMD_CIM). Figure 37.12 shows the pin connection specifications of each SSIF module.

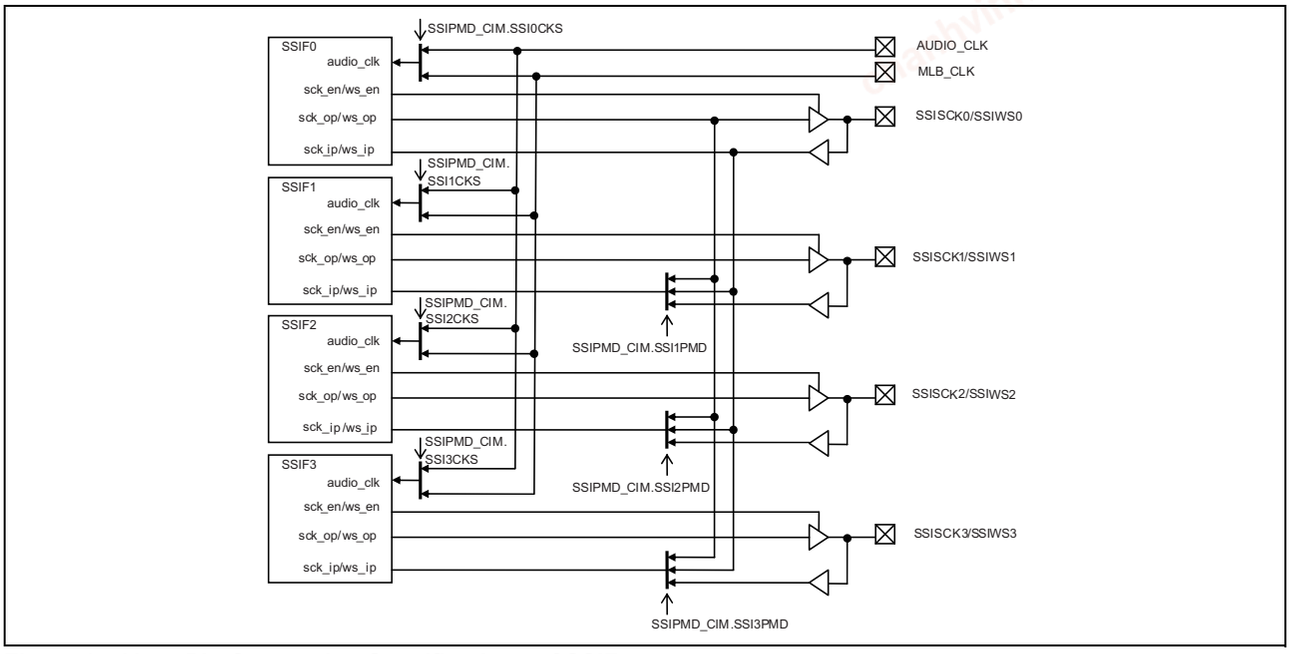


Figure 37.12 Pin Connection Specifications of SSIF

chanhvinh.com

37.4.5 Data Transfer Route

Table 37.12 to Table 37.14 list the data transfer routes that can be selected by the SCUX.
Table 37.15 to Table 37.17 show the necessary register settings for selecting each transfer route.
The 2SRC0_m bypass register p and DVU0_n bypass register should be set appropriately to bypass the SRC and DVU, respectively.

Table 37.12 Data Transfer Routes (1)

Route	No.	SSIF (SRC)	FFD	SRC + DVU	MIX	SSIF (DST)	FFU
SSIF → SRC → SSIF	1	SSIF0	—	SRC0*1 + DVU0_0	—	SSIF0	—
	2	SSIF0	—	SRC0*1 + DVU0_0	—	SSIF3	—
	3	Reserved					
	4	SSIF3	—	SRC1*1 + DVU0_1	—	SSIF3	—
	5	SSIF3	—	SRC1*1 + DVU0_1	—	SSIF0	—
	6	Reserved					
	7	Reserved					
	8	Reserved					
	9	Reserved					
	10	Reserved					
SSIF → SRC → FFU	1	SSIF0	—	SRC0*1	—	—	FFU0_0
	2	SSIF012 (6 ch)	—	SRC0*1	—	—	FFU0_0
	3	SSIF3	—	SRC1*1	—	—	FFU0_1
	4	Reserved					
	5	SSIF1	—	SRC2*1	—	—	FFU0_2
	6	Reserved					
	7	SSIF2	—	SRC3*1	—	—	FFU0_3
	8	Reserved					

Note 1. Selectable only in asynchronous mode. Not selectable in synchronous mode.

chanhvinh.com

Table 37.13 Data Transfer Routes (2)

Route	No.	SSIF (SRC)	FFD	SRC + DVU	MIX	SSIF (DST)	FFU
SSIF → SRC → MIX → SSIF	1	SSIF0	—	SRC0*1 + DVU0_0	MIX	SSIF0	—
	2	SSIF0	—	SRC0*1 + DVU0_0	MIX	SSIF3	—
	3	Reserved					
	4	SSIF3	—	SRC1*1 + DVU0_1	MIX	SSIF3	—
	5	SSIF3	—	SRC1*1 + DVU0_1	MIX	SSIF0	—
	6	SSIF3	—	SRC1*1 + DVU0_1	MIX	SSIF012 (6 ch)	—
	7	SSIF1	—	SRC2*1 + DVU0_2	MIX	SSIF0	—
	8	SSIF1	—	SRC2*1 + DVU0_2	MIX	SSIF3	—
	9	Reserved					
	10	Reserved					
	11	Reserved					
	12	Reserved					
	13	SSIF2	—	SRC3*1 + DVU0_3	MIX	SSIF0	—
	14	SSIF2	—	SRC3*1 + DVU0_3	MIX	SSIF3	—
	15	Reserved					
	16	Reserved					
	17	Reserved					
	18	Reserved					
FFD → SRC → SSIF	1	—	FFD0_0	SRC0*1 + DVU0_0	—	SSIF0	—
	2	—	FFD0_0	SRC0*1 + DVU0_0	—	SSIF012 (6 ch)	—
	3	—	FFD0_0	SRC0*1 + DVU0_0	—	SSIF3	—
	4	Reserved					
	5	—	FFD0_1	SRC1*1 + DVU0_1	—	SSIF3	—
	6	Reserved					
	7	—	FFD0_1	SRC1*1 + DVU0_1	—	SSIF0	—
	8	—	FFD0_1	SRC1*1 + DVU0_1	—	SSIF012 (6 ch)	—
	9	—	FFD0_2	SRC2*1 + DVU0_2	—	SSIF1	—
	10	Reserved					
	11	—	FFD0_3	SRC3*1 + DVU0_3	—	SSIF2	—
	12	Reserved					

Note 1. Selectable only in asynchronous mode. Not selectable in synchronous mode.

chanhvinh.com

Table 37.14 Data Transfer Routes (3)

Route	No.	SSIF (SRC)	FFD	SRC+DVU	MIX	SSIF (DST)	FFU
FFD → SRC → FFU	1	—	FFD0_0	SRC0 (Sync)			FFU0_0
	2	—	FFD0_0	SRC0 (Async)			FFU0_0
	3	—	FFD0_1	SRC1 (Sync)			FFU0_1
	4	—	FFD0_1	SRC1 (Async)			FFU0_1
	5	—	FFD0_2	SRC2 (Sync)			FFU0_2
	6	—	FFD0_2	SRC2 (Async)			FFU0_2
	7	—	FFD0_3	SRC3 (Sync)			FFU0_3
	8	—	FFD0_3	SRC3 (Async)			FFU0_3
FFD → SRC → MIX → SSIF	1	—	FFD0_0	SRC0*1 + DVU0_0	MIX	SSIF0	—
	2	—	FFD0_0	SRC0*1 + DVU0_0	MIX	SSIF012 (6 ch)	—
	3	—	FFD0_0	SRC0*1 + DVU0_0	MIX	SSIF3	—
	4	Reserved					
	5	—	FFD0_1	SRC1*1 + DVU0_1	MIX	SSIF0	—
	6	—	FFD0_1	SRC1*1 + DVU0_1	MIX	SSIF012 (6 ch)	—
	7	—	FFD0_1	SRC1*1 + DVU0_1	MIX	SSIF3	—
	8	Reserved					
	9	—	FFD0_2	SRC2*1 + DVU0_2	MIX	SSIF0	—
	10	—	FFD0_2	SRC2*1 + DVU0_2	MIX	SSIF012 (6 ch)	—
	11	—	FFD0_2	SRC2*1 + DVU0_2	MIX	SSIF3	—
	12	Reserved					
	13	—	FFD0_3	SRC3*1 + DVU0_3	MIX	SSIF0	—
	14	—	FFD0_3	SRC3*1 + DVU0_3	MIX	SSIF012 (6 ch)	—
	15	—	FFD0_3	SRC3*1 + DVU0_3	MIX	SSIF3	—
	16	Reserved					

Note 1. Selectable only in asynchronous mode. Not selectable in synchronous mode.

chanhvinh.com

Table 37.15 Route Select Register Settings (1)

Route	No.	SSIRSEL_CIM								IPSLR_IPC0_n	OPSLR_OPC0_n	FFDPR_FFD0_n	FFUPR_FFU0_n	SSIPMD_CIM
		SISEL3	SISEL2	SISEL1	SISEL0	SOSEL3	SOSEL2	SOSEL1	SOSEL0	IPC_PASS_SEL	OPC_PASS_SEL	PASS	PASS	SSI012EN
SSIF → SRC → SSIF	1	-	-	-	00	-	-	-	00	001	001	00	00	0
	2	-	-	-	00	01	-	-	-	001	001	00	00	0
	3	Reserved												
	4	-	-	00	-	00	-	-	-	001	001	00	00	0
	5	-	-	00	-	-	-	-	01	001	001	00	00	0
	6	Reserved												
	7	Reserved												
	8	Reserved												
	9	Reserved												
	10	Reserved												
SSIF → SRC → FFU	1	-	-	-	00	-	-	-	-	001	011	00	01	0
	2	-	-	-	01	-	-	-	-	001	011	00	01	0
	3	-	-	00	-	-	-	-	-	001	011	00	01	0
	4													
	5	-	00	-	-	-	-	-	-	001	011	00	01	0
	6													
	7	00	-	-	-	-	-	-	-	001	011	00	01	0
	8	Reserved												

chanhvinh.com

Table 37.16 Route Select Register Settings (2)

Route	No.	SSIRSEL_CIM								IPSLR_IPC0_n	OPSLR_OPC0_n	FFDPR_FFD0_n	FFUPR_FFU0_n	SSIPMD_CIM
		SISEL3	SISEL2	SISEL1	SISEL0	SOSEL3	SOSEL2	SOSEL1	SOSEL0	IPC_PASS_SEL	OPC_PASS_SEL	PASS	PASS	SSI012EN
SSIF → SRC → MIX → SSI	1	-	-	-	00	-	-	-	10	001	001	00	00	0
	2	-	-	-	00	10	-	-	-	001	001	00	00	0
	3	Reserved												
	4	-	-	00	-	10	-	-	-	001	001	00	00	0
	5	-	-	00	-	-	-	-	10	001	001	00	00	0
	6	-	-	00	-	-	11	11	10	001	001	00	00	1
	7	-	00	-	-	-	-	-	10	001	001	00	00	0
	8	-	00	-	-	10	-	-	-	001	001	00	00	0
	9	Reserved												
	10	Reserved												
	11	Reserved												
	12	Reserved												
	13	00	-	-	-	-	-	-	10	001	001	00	00	0
	14	00	-	-	-	10	-	-	-	001	001	00	00	0
	15	Reserved												
	16	Reserved												
	17	Reserved												
	18	Reserved												
FFD → SRC → SSIF	1	-	-	-	-	-	-	-	00	011	001	01	00	0
	2	-	-	-	-	-	01	01	00	011	001	01	00	1
	3	-	-	-	-	01	-	-	-	011	001	01	00	0
	4	Reserved												
	5	-	-	-	-	00	-	-	-	011	001	01	00	0
	6	Reserved												
	7	-	-	-	-	-	-	-	01	011	001	01	00	0
	8	-	-	-	-	-	10	10	01	011	001	01	00	1
	9	-	-	-	-	-	-	00	-	011	001	01	00	0
	10	Reserved												
	11	-	-	-	-	-	00	-	-	011	001	01	00	0
	12	Reserved												

chanhvinh.com

Table 37.17 Route Select Register Settings (3)

Route	No.	SSIRSEL_CIM								IPSLR_IPC0_n	OPSLR_OPC0_n	FFDPR_FFD0_n	FFUPR_FFU0_n	SSIPMD_CIM
		SISEL3	SISEL2	SISEL1	SISEL0	SOSEL3	SOSEL2	SOSEL1	SOSEL0	IPC_PASS_SEL	OPC_PASS_SEL	PASS	PASS	SSI012EN
FFD → SRC → FFU	1	-	-	-	-	-	-	-	-	100	100	10	10	0
	2	-	-	-	-	-	-	-	-	011	011	01	01	0
	3	-	-	-	-	-	-	-	-	100	100	10	10	0
	4	-	-	-	-	-	-	-	-	011	011	01	01	0
	5	-	-	-	-	-	-	-	-	100	100	10	10	0
	6	-	-	-	-	-	-	-	-	011	011	01	01	0
	7	-	-	-	-	-	-	-	-	100	100	10	10	0
	8	-	-	-	-	-	-	-	-	011	011	01	01	0
FFD → SRC → MIX → SSIF	1	-	-	-	-	-	-	-	10	011	001	01	00	0
	2	-	-	-	-	-	11	11	10	011	001	01	00	1
	3	-	-	-	-	10	-	-	-	011	001	01	00	0
	4	Reserved												
	5	-	-	-	-	-	-	-	10	011	001	01	00	0
	6	-	-	-	-	-	11	11	10	011	001	01	00	1
	7	-	-	-	-	10	-	-	-	011	001	01	00	0
	8	Reserved												
	9	-	-	-	-	-	-	-	10	011	001	01	00	0
	10					-	11	11	10	011	001	01	00	1
	11					10	-	-	-	011	001	01	00	0
	12	Reserved												
	13					-	-	-	10	011	001	01	00	0
	14					-	11	11	10	011	001	01	00	1
	15					10	-	-	-	011	001	01	00	0
	16	Reserved												

chanhvinh.com

37.4.6 Input Timing Signal and Output Timing Signal

The input timing signal and output timing signal need to be set when using the SRC or MIX. Table 37.18 to Table 37.20 show the input/output timing setting necessary for each transfer route in Table 37.12 to Table 37.14. When direct connection between the SRC and SSIF is selected, the WS signal of the relevant SSIF is automatically selected regardless of the setting of the FFD0_n/FFU0_n timing select register. When direct connection between the MIX and SSIF is selected, set the prescribed value in the FFU0_n timing select register. When connection between the FFD and SRC or connection between the SRC and FFU is selected, a desired value can be set in the FFD0_n/FFU0_n timing select register and timing signals corresponding to the setting are used.

Table 37.18 Input/Output Timing Setting (1)

Route	No.	Input Timing Setting						Output Timing Setting					
		SSICTRL_CIM				FDTSELn_CIM	Input Timing Signal	SSICTRL_CIM				FUTSELn_CIM	Output Timing Signal
		SSI3RX	SSI2RX	SSI1RX	SSI0RX			SSI3TX	SSI2TX	SSI1TX	SSI0TX		
SSIF → SRC → SSIF	1	—	—	—	1	Setting unnecessary	SSIWS0*1	0*2	—	—	1	Setting unnecessary	SSIWS0*1
	2	—	—	—	1	Setting unnecessary	SSIWS0*1	1	—	—	0*2	Setting unnecessary	SSIWS3*1
	3	Reserved											
	4	1	—	—	—	Setting unnecessary	SSIWS3*1	1	—	—	0*2	Setting unnecessary	SSIWS3*1
	5	1	—	—	—	Setting unnecessary	SSIWS3*1	0*2	—	—	1	Setting unnecessary	SSIWS0*1
	6	Reserved											
	7	Reserved											
	8	Reserved											
	9	Reserved											
	10	Reserved											
SSIF → SRC → FFU	1	—	—	—	1	Setting unnecessary	SSIWS0*1	0*2	—	—	0*2	Arbitrary	←
	2	—	—	—	1	Setting unnecessary	SSIWS0*1	0*2	—	—	0*2	Arbitrary	←
	3	1	—	—	—	Setting unnecessary	SSIWS3*1	0*2	—	—	0*2	Arbitrary	←
	4	Reserved											
	5	—	—	1	—	Setting unnecessary	SSIWS1*1	—	—	0	—	Arbitrary	←
	6	Reserved											
	7	—	1	—	—	Setting unnecessary	SSIWS2*1	—	0*2	—	—	Arbitrary	←
	8	Reserved											

Note 1. Automatically selected regardless of the FDTSELn_CIM and FUTSELn_CIM settings.
Note 2. When connection to an SRCm for a given route is not being selected by the SOSELn bits of register SSIRSEL_CIM, SSInTX may be set to 1 (m = 0 to 3; n = 0 to 3).

chanhvinh.com

Table 37.19 Input/Output Timing Setting (2)

Route	No.	Input Timing Setting						Output Timing Setting					
		SSICTRL_CIM				FDTSELn_CIM	Input Timing Signal	SSICTRL_CIM				FUTSELn_CIM	Output Timing Signal
		SSI3RX	SSI2RX	SSI1RX	SSI0RX			SSI3TX	SSI2TX	SSI1TX	SSI0TX		
SSIF → SRC → MIX → SSIF	1	—	—	—	1	Setting unnecessary	SSIWS0*1	0*3	—	—	1	SCKSEL: 1000	SSIWS0
	2	—	—	—	1	Setting unnecessary	SSIWS0*1	1	—	—	0*3	SCKSEL: 1011	SSIWS3
	3	Reserved											
	4	1	—	—	—	Setting unnecessary	SSIWS3*1	1	—	—	0*3	SCKSEL: 1011	SSIWS3
	5	1	—	—	—	Setting unnecessary	SSIWS3*1	0*3	—	—	1	SCKSEL: 1000	SSIWS0
	6	1	—	—	—	Setting unnecessary	SSIWS3*1	0*3	—	—	1	SCKSEL: 1000	SSIWS0
	7	—	—	1	—	Setting unnecessary	SSIWS1*1	0*4	—	0*2	1	SCKSEL: 1000	SSIWS0
	8	—	—	1	—	Setting unnecessary	SSIWS1*1	1	—	0*2	0*4	SCKSEL: 1011	SSIWS3
	9	Reserved											
	10	Reserved											
	11	Reserved											
	12	Reserved											
	13	—	1	—	—	Setting unnecessary	SSIWS2*1	0*4	0*2	—	1	SCKSEL: 1000	SSIWS0
	14	—	1	—	—	Setting unnecessary	SSIWS2*1	1	0*2	—	0*4	SCKSEL: 1011	SSIWS3
	15	Reserved											
	16	Reserved											
	17	Reserved											
	18	Reserved											
FFD → SRC → SSIF	1	—	—	—	0	Arbitrary	←	0*2	—	—	1	Setting unnecessary	SSIWS0*1
	2	—	—	—	0	Arbitrary	←	0*2	—	—	1	Setting unnecessary	SSIWS0*1
	3	—	—	—	0	Arbitrary	←	1	—	—	0*2	Setting unnecessary	SSIWS3*1
	4	Reserved											
	5	0	—	—	—	Arbitrary	←	1	—	—	0*2	Setting unnecessary	SSIWS3*1
	6	Reserved											
	7	0	—	—	—	Arbitrary	←	0*2	—	—	1	Setting unnecessary	SSIWS0*1
	8	0	—	—	—	Arbitrary	←	0*2	—	—	1	Setting unnecessary	SSIWS0*1
	9	—	—	0	—	Arbitrary	←	—	—	1	—	Setting unnecessary	SSIWS1*1
	10	Reserved											
	11	—	0	—	—	Arbitrary	←	—	1	—	—	Setting unnecessary	SSIWS2*1
	12	Reserved											

Note 1. Automatically selected regardless of the FDTSELn_CIM and FUTSELn_CIM settings.
Note 2. When connection to an SRCm for a given route is not being selected by the SOSELn bits of register SSIRSEL_CIM, SSInTX may be set to 1 (m = 0 to 3; n = 0 to 3).
Note 3. When connection to SRCm and MIX for a given route is not being selected by the SOSELn bits of register SSIRSEL_CIM, SSInTX may be set to 1 (m = 0 to 3; n = 0, 3).
Note 4. When connection to a MIX for a given route is not being selected by the SOSELn bits of register SSIRSEL_CIM, SSInTX may be set to 1 (n = 0, 3).

chanhvinh.com

Table 37.20 Input/Output Timing Setting (3)

Route	No.	Input Timing Setting						Output Timing Setting					
		SSICTRL_CIM				FDTSELn_CIM	Input Timing Signal	SSICTRL_CIM				FUTSELn_CIM	Output Timing Signal
		SSI3RX	SSI2RX	SSI1RX	SSI0RX			SSI3TX	SSI2TX	SSI1TX	SSI0TX		
FFD → SRC → FFU	1	—	—	—	—	—	Unnecessary	—	—	—	—	—	Unnecessary
	2	—	—	—	0	Arbitrary	←	0*2	—	—	0*2	Arbitrary	←
	3	—	—	—	—	—	Unnecessary	—	—	—	—	—	Unnecessary
	4	0	—	—	—	Arbitrary	←	0*2	—	—	0*2	Arbitrary	←
	5	—	—	—	—	—	Unnecessary	—	—	—	—	—	Unnecessary
	6	—	—	0	—	Arbitrary	←	—	—	0*2	—	Arbitrary	←
	7	—	—	—	—	—	Unnecessary	—	—	—	—	—	Unnecessary
	8	—	0	—	—	Arbitrary	←	—	0*2	—	—	Arbitrary	←
FFD → SRC → MIX → SSIF	1	—	—	—	0	Arbitrary	←	0*3	—	—	1	SCKSEL: 1000	SSIWS0
	2	—	—	—	0	Arbitrary	←	0*3	—	—	1	SCKSEL: 1000	SSIWS0
	3	—	—	—	0	Arbitrary	←	1	—	—	0*3	SCKSEL: 1011	SSIWS3
	4	Reserved											
	5	0	—	—	—	Arbitrary	←	0*3	—	—	1	SCKSEL: 1000	SSIWS0
	6	0	—	—	—	Arbitrary	←	0*3	—	—	1	SCKSEL: 1000	SSIWS0
	7	0	—	—	—	Arbitrary	←	1	—	—	0*3	SCKSEL: 1011	SSIWS3
	8	Reserved											
	9	—	—	0	—	Arbitrary	←	0*4	—	0*2	1	SCKSEL: 1000	SSIWS0
	10	—	—	0	—	Arbitrary	←	0*4	—	0*2	1	SCKSEL: 1000	SSIWS0
	11	—	—	0	—	Arbitrary	←	1	—	0*2	0*4	SCKSEL: 1011	SSIWS3
	12	Reserved											
	13	—	0	—	—	Arbitrary	←	0*4	0*2	—	1	SCKSEL: 1000	SSIWS0
	14	—	0	—	—	Arbitrary	←	0*4	0*2	—	1	SCKSEL: 1000	SSIWS0
	15	—	0	—	—	Arbitrary	←	1	0*2	—	0*4	SCKSEL: 1011	SSIWS3
	16	Reserved											

Note 1. Automatically selected regardless of the FDTSELn_CIM and FUTSELn_CIM settings.
Note 2. When connection to an SRCm for a given route is not being selected by the SOSELn bits of register SSIRSEL_CIM, SSInTX may be set to 1 (m = 0 to 3; n = 0 to 3).
Note 3. When connection to SRCm and MIX for a given route is not being selected by the SOSELn bits of register SSIRSEL_CIM, SSInTX may be set to 1 (m = 0 to 3; n = 0, 3).
Note 4. When connection to a MIX for a given route is not being selected by the SOSELn bits of register SSIRSEL_CIM, SSInTX may be set to 1 (n = 0, 3).

chanhvinh.com

37.4.7 2SRC (SRC) Block

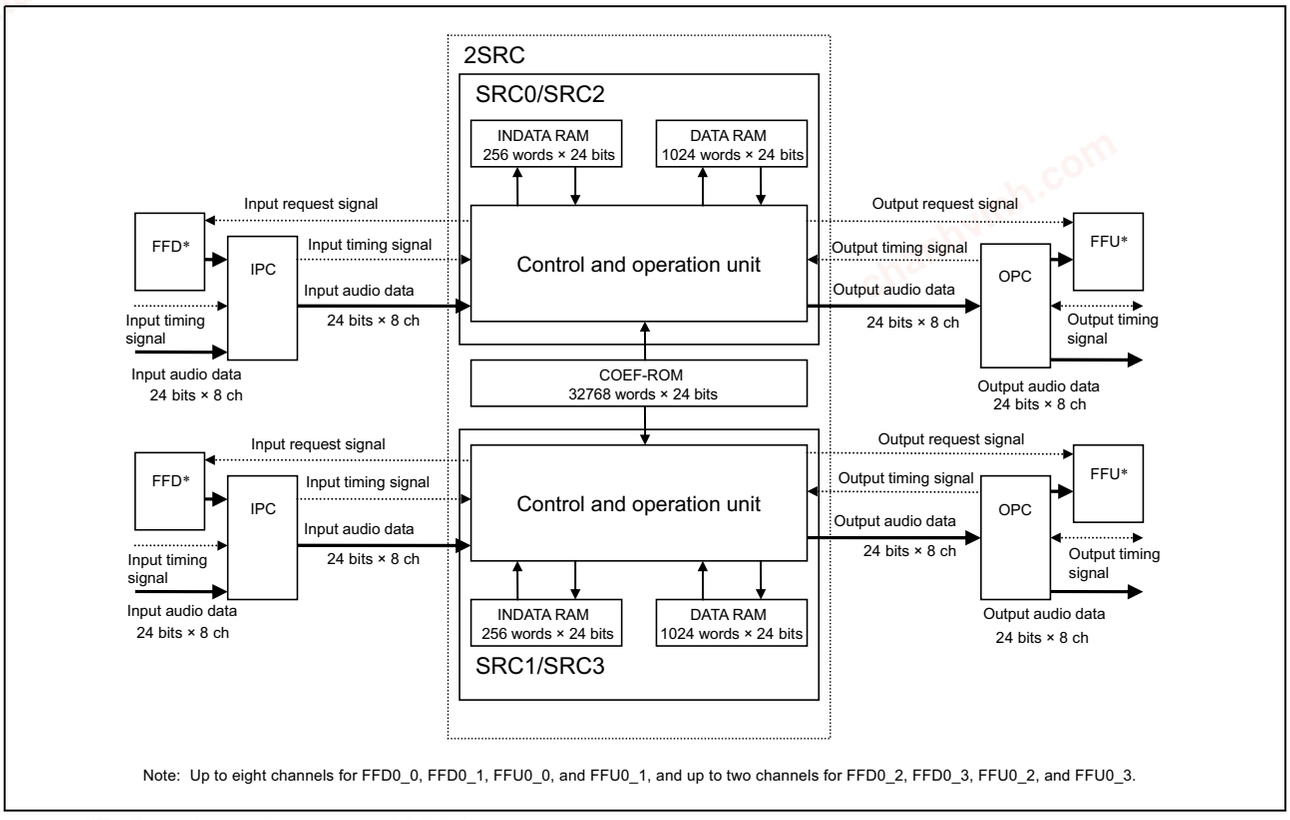
The 2SRC block consists of two SRCs (Sampling Rate Converters) which implement the sampling rate conversion function. Each SRC has two modes allowing either synchronous or asynchronous sampling rate conversion to be performed.

- Asynchronous or synchronous sampling rate conversion is possible*1
- Sampling rate (synchronous mode)*2
  Input [KHz]: 8, 11.025, 12, 16, 22.05, 24, 32, 44.1, 48, 64, 88.2, or 96 is selectable
  Output [KHz]: 8, 16, 24, 32, 44.1, 48, or 96 is selectable
- Sampling rate (asynchronous mode)*2
  Input/output [KHz]: 1 to 96
- The supported number of bits is 16 bits and 24 bits
- Sound quality: −132 dB or less*3
- Supports 1, 2, 4, 6, or 8 channels*4
- DMA transfer with on-chip memory or external memory and direct transfer with the SSIF module are possible

Note 1. The synchronous mode can be selected only when the SCUX is connected to the FFD and FFU modules.
Note 2. The selectable sampling rates depend on the number of used channels and rate ratio.
Note 3. The data format is a 24-bit value.
Note 4. The number of selectable channels depends on the sampling rate and route.

Figure 37.13 shows a block diagram of the 2SRC (SRC) block. Input audio data indicates the data before executing the SRC function and output audio data indicates the data after executing the SRC function.
In synchronous SRC mode, the 2SRC (SRC) block outputs the input request signal to the FFD block and outputs the output request signal to the FFU block. In asynchronous SRC mode, the 2SRC (SRC) block receives the input timing signal and output timing signal from an external module and automatically detects the input sampling rate and output sampling rate from these timing signals.
Each SRC in the 2SRC (SRC) block can simultaneously process audio data for a maximum of eight channels. However, the maximum sampling rate that can be handled depends on the number of channels used. When SRC2 or SRC3 is used by the route from the FFD or the route to the FFU, the maximum number of processable channels is two. The number of channels is set by SADIRp_2SRC0_m.CHNUM. When not in bypass mode (SRCBRp_2SRC0_m.BYPASS = 0), unused channels always output 0 after initialization (SRCIRp_2SRC0_m.INIT = 1) unless the CHNUM setting is changed. In bypass mode, the input audio data is output one cycle later in all channels regardless of the CHNUM setting.
The bit width of output audio data can be set to 16 bits or 24 bits by SADIRp_2SRC0_m.OTBL. For input audio data, audio data with the same bit width as the OTBL bit setting in Table 37.21 should be input.

chanhvinh.com



Note: Up to eight channels for FFD0_0, FFD0_1, FFU0_0, and FFU0_1, and up to two channels for FFD0_2, FFD0_3, FFU0_2, and FFU0_3.

Figure 37.13 Block Diagram of 2SRC Block

Table 37.21 Combinations of Bit Width of Input/Output Audio Data

SADIRp_2SRC0_m.OTBL	Bit Width of Input Audio Data	Bit Width of Output Audio Data
00000 = 24 bits	24 bits	24 bits
01000 = 16 bits	16 bits	16 bits

Table 37.22 shows the functions of the 2SRC (SRC) block.

Table 37.22 2SRC (SRC) Block Functions

Item	Performance					
Type of SRC	Asynchronous SRC/synchronous SRC					
Operation frequency	66 MHz					
Delay mode	Normal				Low delay 1	Low delay 2
Channel number	1 or 2	4	6	8	1 or 2	1 or 2
Sampling rate of input source	8k to 96k [Hz]	8k to 96k [Hz]	8k to 66k [Hz]	8k to 49k [Hz]	8k to 96k [Hz]	8k to 96k [Hz]
Sampling rate of output source	8k to 96k [Hz]	8k to 96k [Hz]	8k to 66k [Hz]	8k to 49k [Hz]	8k to 96k [Hz]	8k to 96k [Hz]
Ratio of input and output sampling rate (FSO/FSI)	16 to 0.125 [time]	16 to 0.25 [time]	16 to 0.375 [time]	16 to 0.5 [time]	16 to 0.5 [time]	16 to 1 [time]
Sound quality	–132 dB					

chanhvinh.com

Table 37.23 shows the latency of the 2SRC (SRC) block.

Table 37.23 2SRC (SRC) Block Latency

Item	Bypass	Normal					Low Delay 1		Low Delay 2	
Channel number	Any	1	2	4	6	8	1	2	1	2
Processing delay	0	641	321	161	102	81	81	81	49	49
Logic delay	1	3	3	3	3	3	3	3	3	3
Output delay	Output delay [sample] = (Processing delay) * (FSO/FSI ratio) + (Logic delay)									

37.4.8 DVU Block

The DVU block is used to control volume and mute. This block can simultaneously process data for a maximum of eight channels.

- Digital volume, volume ramp, and zero cross mute are provided as functions to adjust the volume
- The digital volume is set as a 24-bit fixed-point value within the range from a multiple of 0 to 8 (mute, –120 to 18 dB)
- Volume ramp can be used to perform soft mute, fade in, fade out, and volume change as desired
- The ramp time of volume ramp can be changed and set within the sampling range of 2^0 to 2^{23}
- Zero cross mute turns the sound mute at the zero cross point of audio data
- Direct transfer with the SSIF module and transfer with the mixer are possible

The number of channels is set by VADIR_DVU0_n.CHNUM. For unused channels, the input data that is input to the corresponding channels is output one cycle later. In bypass mode, the input data is output one cycle later in all channels regardless of the CHNUM setting.
Figure 37.14 shows the processing image of the DVU block.

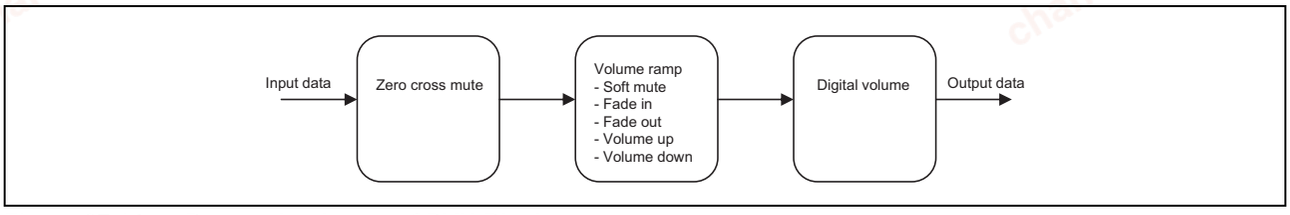


Figure 37.14 Processing Image of DVU Block

chanhvinh.com

Table 37.24 shows the functions of the DVU block.

Table 37.24 DVU Block Functions

Item	Performance
Digital volume	Range: –120 dB to 18 dB (9.5 × 10^{-7}-time to 8-time)
Volume ramp	Volume ramp use for many kinds of operation (soft mute, fade in, fade out, volume change by ramp) Ramp time: 2^0/fso to 2^{23}/fso Example: (1/fso: –128 dB/1 step) (2/fso: –64 dB/1 step) (4/fso: –32 dB/1 step) ..... (128/fso: –1 dB/1 step) (256/fso: –0.5 dB/1 step) (512/fso: –0.25 dB/1 step) (1024/fso: –0.125 dB/1 step) (2048/fso: –0.125 dB/2 steps) (4096/fso: –0.125 dB/4 steps) ..... (8388608/fso: –0.125 dB/8192 steps)
Zero cross mute	Mute the signal at zero cross point

Figure 37.15 shows the image of digital volume operation.

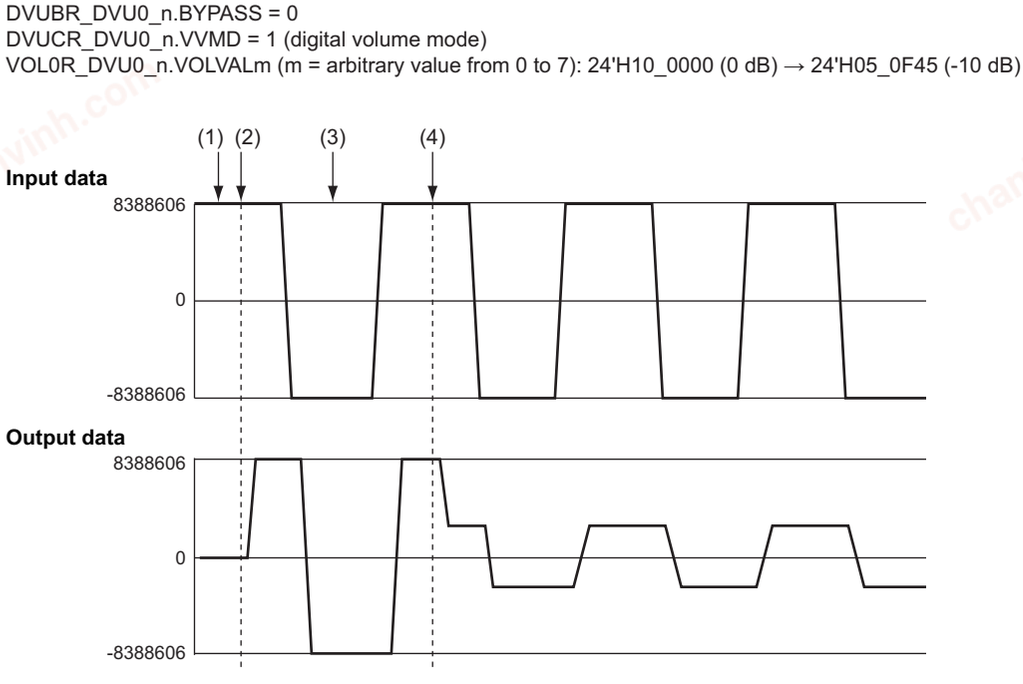


Figure 37.15 Digital Volume Operation

chanhvinh.com

Operation

(1) In the initialized state (DVUIR_DVU0_n.INIT = 1), set the VVMD bit in DVUCR_DVU0_n to 1 (digital volume mode) and the VOLVALm bit in VOL0R_DVU0_n to 24'H10_0000 (0 dB). Until the initialized state is cleared, the DVU does not start operation and its output data is 0.

(2) When the initialized state is cleared (DVUIR_DVU0_n.INIT = 0), the DVU starts operation. Since the VOLVALm bit is set to 0 dB, the output data is the same size as the input data.

(3) Set the DVUEN bit in DVUER_DVU0_n to 0 and the VOLVAL0 bit in VOL0R_DVU0_n to 24'H05_0F45 (-10 dB).

(4) The setting to the VOLVAL0 bit in (3) is enabled by setting the DVUEN bit in DVUER_DVU0_n to 1. The output data is the value obtained by attenuating the input data by -10 dB.

Figure 37.16 shows the image of volume ramp operation.

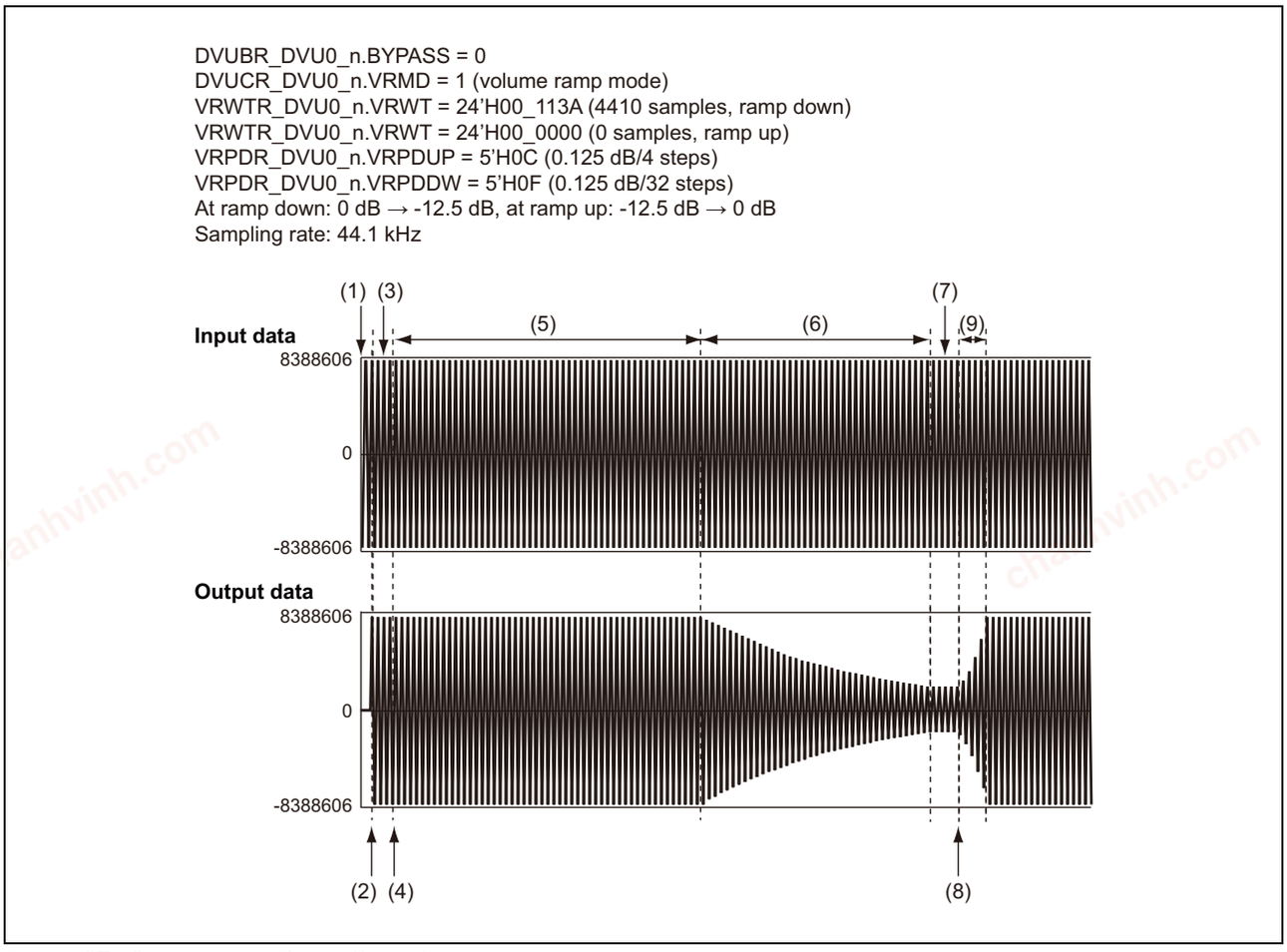


Figure 37.16 Volume Ramp Operation

chanhvinh.com

Operation

(1) In the initialized state (DVUIR_DVU0_n.INIT = 1), set the VRMD bit in DVUCR_DVU0_n to 1 (volume ramp mode) and the VRENm (m: arbitrary value from 0 to 7) bit in VRCTR_DVU0_n to 1. Until the initialized state is cleared, the DVU does not start operation and its output data is 0.

(2) When the initialized state is cleared (DVUIR_DVU0_n.INIT = 0), the DVU starts operation.
Note: Changes to the settings of the VRDB bit of the VRDBR_DVU0_n register made in the INIT state rapidly become applicable on release from the INIT state (independently of the setting of VRWTR_DVU0_n and VRPDR_DVU0_n).

(3) Set the DVUEN bit in DVUER_DVU0_n to 0, the VRPDDW bit in VRPDR_DVU0_n to 5'H0F (0.125 dB/32 steps), the VRDB bit in VRDBR_DVU0_n to 10'H064 (-12.5 dB), and the VRWT bit in VRWTR_DVU0_n to 24'H00_113A (4410 samples).

(4) The setting in (3) is enabled by setting the DVUEN bit in DVUER_DVU0_n to 1.

(5) The VDU waits to start the volume ramp operation for 4410/44100 = 0.1 second because the sampling rate is set to 44.1 kHz with VRWTR_DVU0_n.VRWT = 24'H00_113A (4410 samples).

(6) The VDU starts volume ramp operation after the wait time in (5).
The output data is the value obtained by attenuating the input data by -12.5 dB after 12.5 × 32/0.125 = 3200 steps because the VRPDDW bit in VRPDR_DVU0_n is set to 5'H0F (0.125 dB/32 steps) according to 0d B to -12.5 dB ramp down.

(7) Set the DVUEN bit in DVUER_DVU0_n to 0, the VRPDUP bit in VRPDR_DVU0_n to 5'H0C (0.125 dB/4 steps), the VRDB bit in VRDBR_DVU0_n to 10'H000 (0 dB), and the VRWT bit in VRWTR_DVU0_n to 24'H00_0000 (0 sample).

(8) The setting in (7) is enabled by setting the DVUEN bit in DVUER_DVU0_n to 1.

(9) The VDU starts volume ramp operation without delay because the VRWT bit in VRWTR_DVU0_n is set to 24'H00_0000 (0 sample).
The output data is the same size as the input data after 12.5 × 4/0.125 = 400 steps because the VRPDUP bit in VRPDR_DVU0_n is set to 5'H0C (0.125 dB/4 steps) according to -12.5 dB to 0 dB ramp up.

Table 37.25 shows the latency of the DVU block.

Table 37.25 DVU Block Latency

Item	Bypass	DVU Operation
Output delay	1	2

chanhvinh.com

37.4.9 MIX Block

The MIX block is used to mix (add together) the input audio data from a maximum of four systems into one system. Each system can simultaneously process data for a maximum of eight channels.

- Data of two to four source systems can be mixed (added together) into one system
- The ratio to add the sources can be set
- The ratio can be changed dynamically
- Volume ramp enables mixing to be performed (ramp time is variable)
- Only direct transfer with the SSIF module is possible

The number of channels is set by MADIR_MIX0_0.CHNUM. When not in bypass mode (MIXBR_MIX0_0.BYPASS = 0), unused channels always output 0 after initialization (MIXIR_MIX0_0.INIT = 1) unless the CHNUM setting is changed. In bypass mode, the input data is output one cycle later in all channels regardless of the CHNUM setting. Figure 37.17 shows a block diagram of the MIX block.

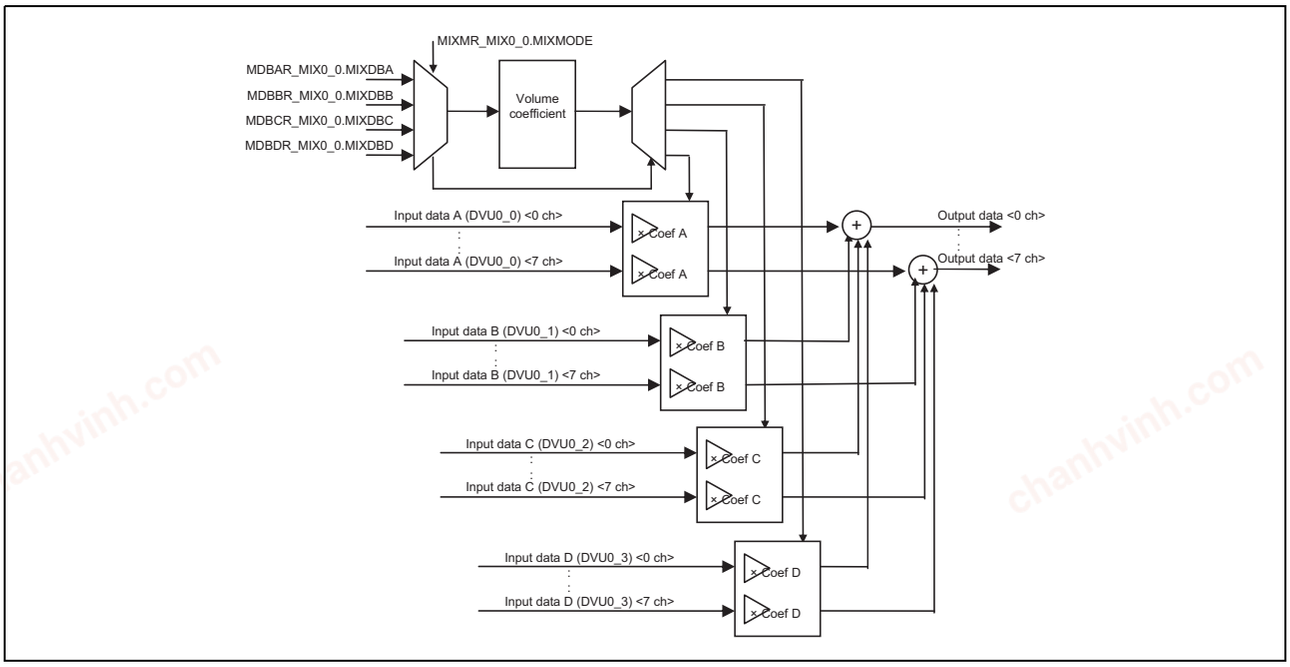


Figure 37.17 Block Diagram of MIX Block

Table 37.26 shows the functions of the MIX block.

Table 37.26 MIX Block Functions

Item	Performance
Volume step mixer	Mix the data of four systems into one system. The volume of each system can be adjusted by step change.
Volume ramp mixer	Mix the data of four systems into one system. The volume of each system can be adjusted by ramp.

Figure 37.18 shows the image of volume step operation for 1-system data. This figure explains about the operation when the volume is changed from 0 dB to –∞ dB. The volume can change the target dB for one step each. In the case of the volume step mixer, the volume of four systems can be adjusted at each step change (one sample).
Figure 37.19 shows the image of volume ramp operation for 1-system data. This figure explains about the operation when the volume is changed from 0 dB to –∞ dB. The volume can be changed by 0.125-dB/step each and it takes 1024 samples to reach –∞ dB in the maximum case for each system. In the case of the volume ramp mixer, the volume of four

chanhvinh.com

systems can be adjusted at each ramp. In addition, the volume ramp mixer has a mechanism to reduce the noise. If the direction of volume is different among the four systems, it operates from the volume-down systems and after that operates on the volume-up systems automatically. In the case of the same direction, it changes the volume at the same timing.

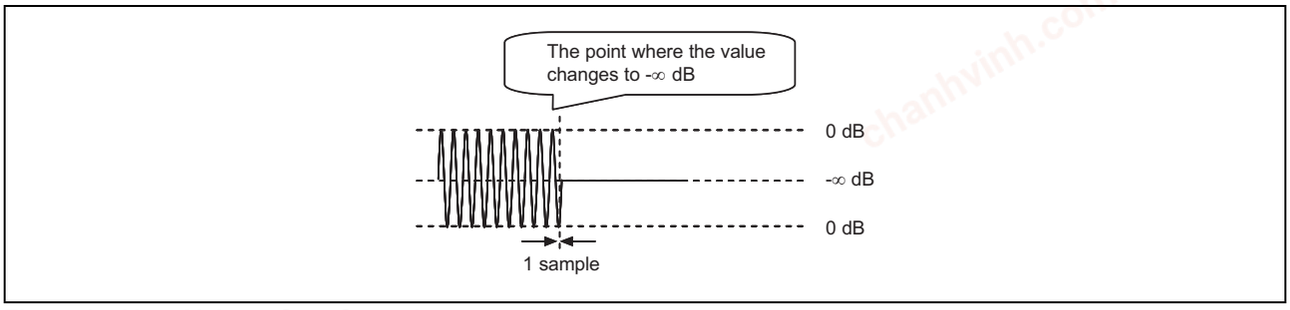


Figure 37.18 Volume Step Operation

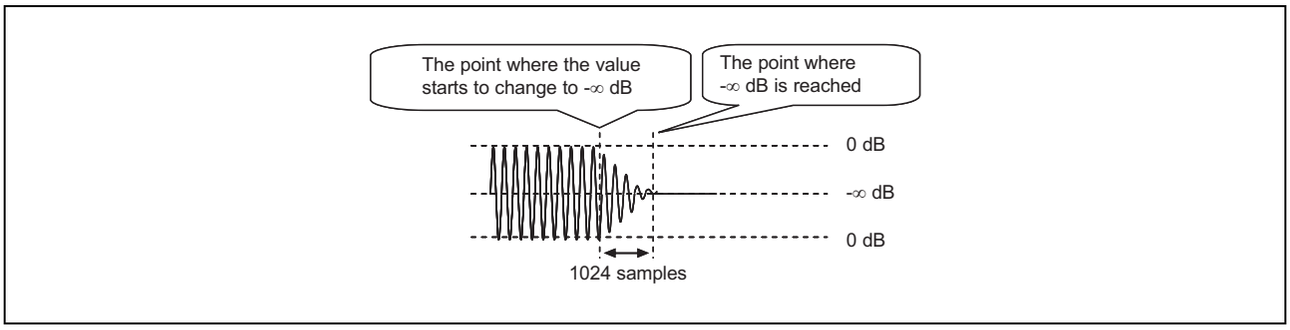


Figure 37.19 Volume Ramp Operation

chanhvinh.com

Figure 37.20 shows the image of volume ramp operation for 4-system data.

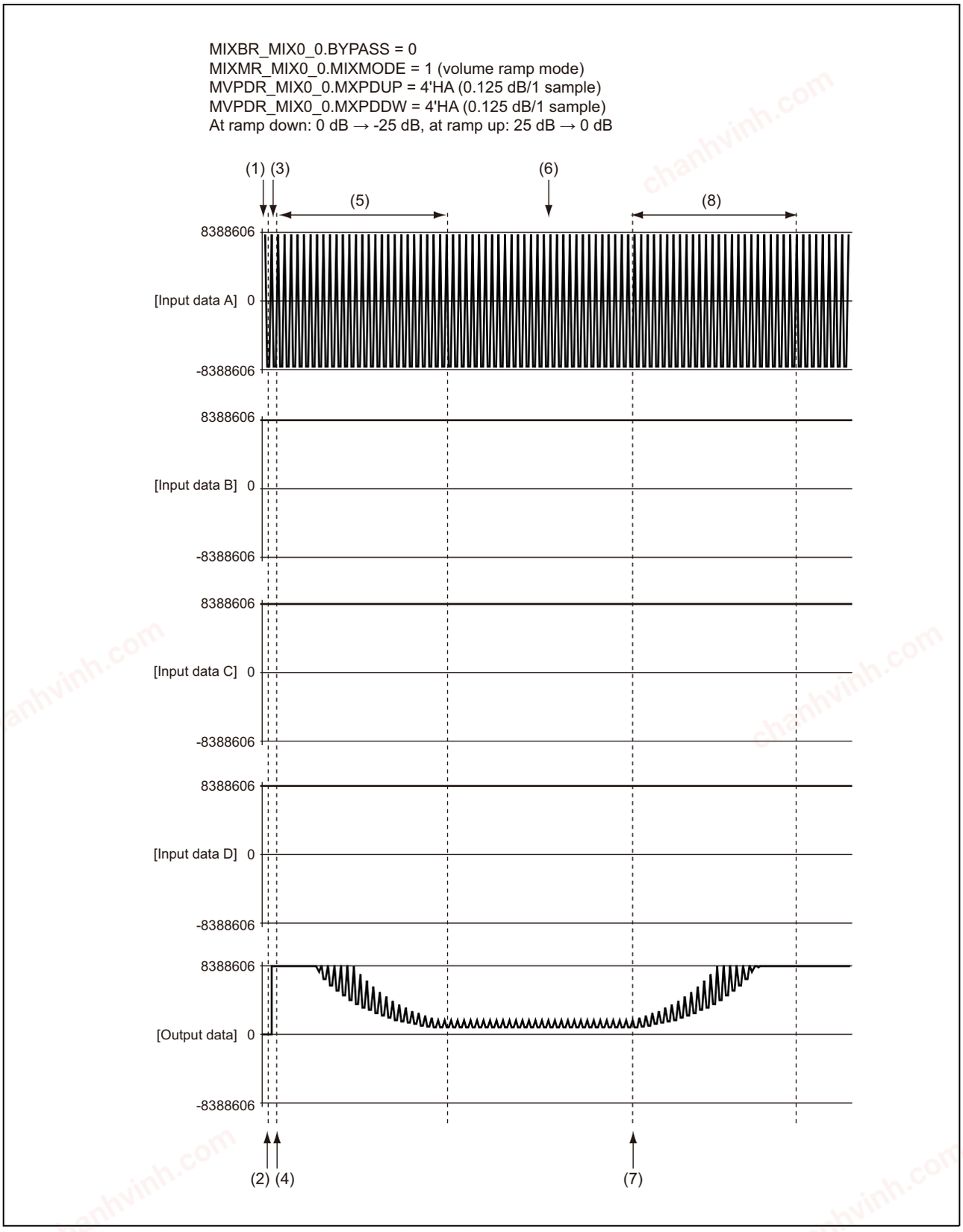


Figure 37.20 Volume Ramp Operation for 4-System Data

chanhvinh.com

Operation

(1) In the initialized state (MIXIR_MIX0_0.INIT = 1), set the MIXMODE bit in MIXMR_MIX0_0 to 1 (volume ramp mode) and the MXPDDW bit in MVPDR_MIX0_0 to 4'HA (0.125 dB/1 sample).
Until the initialized state is cleared, the MIX does not start operation and its output data is 0.

(2) When the initialized state is cleared (MIXIR_MIX0_0.INIT = 0), the MIX starts operation.
Note: Changes to the settings of the MIXDBA bits of the MDBAR_MIX0_0 register, MIXDBB bits of the MDBBR_MIX0_0 register, MIXDBC bits of the MDBCR_MIX0_0 register, and MIXDBD bits of the MDBDR_MIX0_0 register made in the INIT state rapidly become applicable on release from the INIT state (independently of the setting of MVPDR_MIX0_0).

(3) Set as follows:
MDBER_MIX0_0.MIXDBEN = 0
MDBAR_MIX0_0.MIXDBA = 10'H0C8 (-25 dB)
MDBBR_MIX0_0.MIXDBB = 10'H0C8 (-25 dB)
MDBCR_MIX0_0.MIXDBC = 10'H0C8 (-25 dB)
MDBDR_MIX0_0.MIXDBD = 10'H0C8 (-25 dB)

(4) The setting in (3) is enabled by setting the MIXDBEN bit in MDBER_MIX0_0 to 1.

(5) The output data is the value obtained by attenuating and mixing the input data of each system by -25 dB after 25/0.125 = 200 steps because the MXPDDW bit in MVPDR_MIX0_0 is set to 4'HA (0.125 dB/1 sample) according to 0 dB to -25 dB ramp down for each system.

(6) Set as follows:
MDBER_MIX0_0.MIXDBEN = 0
MDBAR_MIX0_0.MIXDBA = 10'H000 (0 dB)
MDBBR_MIX0_0.MIXDBB = 10'H000 (0 dB)
MDBCR_MIX0_0.MIXDBC = 10'H000 (0 dB)
MDBDR_MIX0_0.MIXDBD = 10'H000 (0 dB)

(7) The setting in (6) is enabled by setting the MIXDBEN bit in MDBER_MIX0_0 to 1.

(8) The output data is the value obtained by boosting and mixing the input data of each system by 25 dB after 25/0.125 = 200 steps because the MXPDUP bit in MVPDR_MIX0_0 is set to 4'HA (0.125 dB/1 sample) according to -25 dB to 0 dB ramp up for each system.

Table 37.27 shows the latency of the MIX block.

Table 37.27 MIX Block Latency

Item	Bypass	MIX Operation
Output delay	1	2

chanhvinh.com

37.5 Usage Note

37.5.1 Software Reset

When initializing the internal circuits of this module by clearing the SWRST bit in the SWRSR_CIM register or setting the INIT bit in the IPCIR_IPC0_n, OPCIR_OPC0_n, FFDIR_FFD0_n, FFUIR_FFU0_n, SRCIRp_2SRC0_m, SRCIRR_2SRC0_m, DVUIR_DVU0_n, or MIXIR_MIX0_0 register, be sure to follow the procedures specified in section 37.4.1, Initial Setting Procedure or section 37.4.2, Transfer Start Procedure and Stop Procedure.

chanhvinh.com

38. SD Host Interface

Note: • SD Host/Ancillary Product License Agreement (SD HALA) is required to develop SD host-related products.

38.1 Overview

38.1.1 Features

- SD memory/IO card interface (1-bit/4-bit SD bus)
- SD memory card: SD, SDHC, SDXC
- Transfer mode: Default mode, High-Speed mode
- SD clock (SD_CLK) frequency = $1/2^n$ P1φ (Peripheral clock 1) frequency (n = 1 to 9)
- Error check function: CRC7 (for command/response), CRC16 (for data)
- Three interrupt requests
- DMA transfer request: SD_BUF read/write
- Card detect function
- Write protect support

38.1.2 Block Diagram

Figure 38.1 shows a block diagram of the SD host interface.

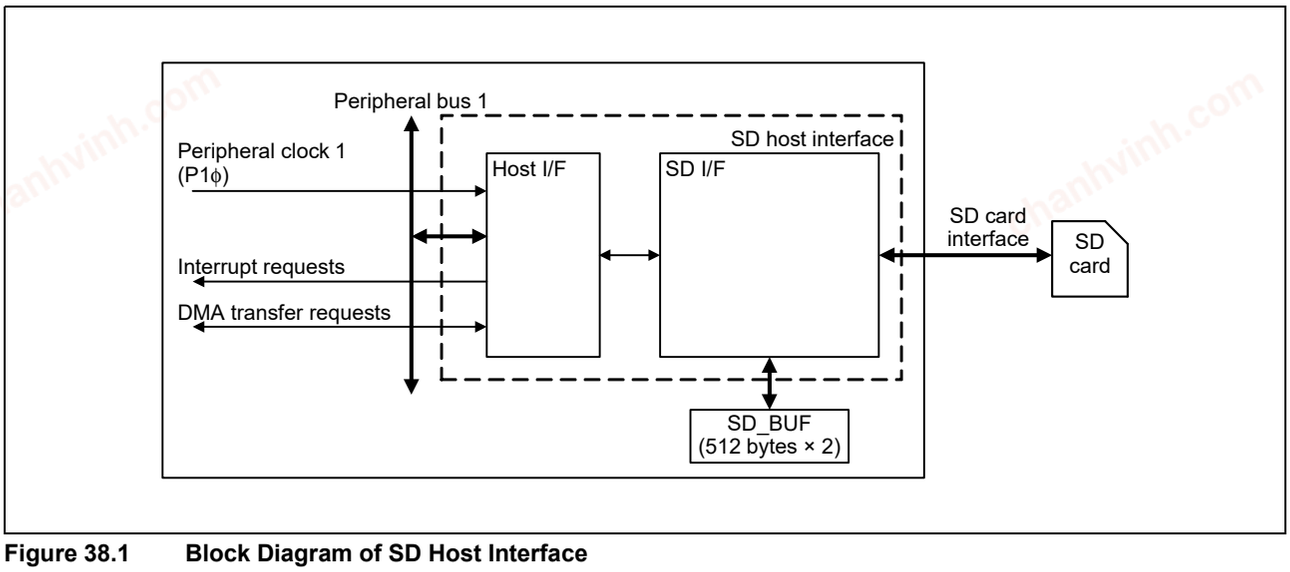


Figure 38.1 Block Diagram of SD Host Interface

chanhvinh.com

38.1.3 Input/Output Pins

Table 38.1 lists the input and output pins used by the SD host interface.

Table 38.1 Pin Configuration

Pin Name	Signal Name	I/O	Function
SD_CLK[1:0]*1	SDCLK	Output	SD clock
SD_CMD[1:0]*1	SDCMD	I/O	Command/response
SD_D0[1:0]*1	SDDAT0	I/O	Data[bit 0]
SD_D1[1:0]*1	SDDAT1	I/O	Data[bit 1]/SDIO interrupt
SD_D2[1:0]*1	SDDAT2	I/O	Data[bit 2]/Read wait
SD_D3[1:0]*1	SDDAT3	I/O	Data[bit 3]
SD_CD[1:0]*1	ISDCD	Input	Card detect*2
SD_WP[1:0]*1	ISDWP	Input	Write protect*2

Note 1. [1:0] indicates the SD Host Interface channel number.
Note 2. When a pin is not in use, fix the corresponding signal to 1.

chanhvinh.com

38.2 Register Descriptions

Table 38.2 summaries the registers of the SD host interface.

Table 38.2 Register Configuration

Channel	Register Name	Abbreviation	Address	Access Size
0	Command type register	SD_CMD	H'E804E000	16
	SD command argument registers	SD_ARG0	H'E804E004	16
		SD_ARG1	H'E804E006	16
	Data stop register	SD_STOP	H'E804E008	16
	Block count register	SD_SECCNT	H'E804E00A	16
	SD card response registers	SD_RSP00	H'E804E00C	16
		SD_RSP01	H'E804E00E	16
		SD_RSP02	H'E804E010	16
		SD_RSP03	H'E804E012	16
		SD_RSP04	H'E804E014	16
		SD_RSP05	H'E804E016	16
		SD_RSP06	H'E804E018	16
		SD_RSP07	H'E804E01A	16
	SD card interrupt flag register 1	SD_INFO1	H'E804E01C	16
	SD card interrupt flag register 2	SD_INFO2	H'E804E01E	16
	SD_INFO1 interrupt mask register	SD_INFO1_MASK	H'E804E020	16
	SD_INFO2 interrupt mask register	SD_INFO2_MASK	H'E804E022	16
	SD clock control register	SD_CLK_CTRL	H'E804E024	16
	Transfer data length register	SD_SIZE	H'E804E026	16
	SD card access control option register	SD_OPTION	H'E804E028	16
	SD error status register 1	SD_ERR_STS1	H'E804E02C	16
	SD error status register 2	SD_ERR_STS2	H'E804E02E	16
	SD buffer read/write register	SD_BUF0	H'E804E030	32
	SDIO mode control register	SDIO_MODE	H'E804E034	16
	SDIO interrupt flag register	SDIO_INFO1	H'E804E036	16
	SDIO_INFO1 interrupt mask register	SDIO_INFO1_MASK	H'E804E038	16
	DMA mode enable register	CC_EXT_MODE	H'E804E0D8	16
	Software reset register	SOFT_RST	H'E804E0E0	16
	Version register	VERSION	H'E804E0E2	16
	Swap control register	EXT_SWAP	H'E804E0F0	16

chanhvinh.com

Channel	Register Name	Abbreviation	Address	Access Size
1	Command type register	SD_CMD	H'E804E800	16
	SD command argument registers	SD_ARG0	H'E804E804	16
		SD_ARG1	H'E804E806	16
	Data stop register	SD_STOP	H'E804E808	16
	Block count register	SD_SECCNT	H'E804E80A	16
	SD card response registers	SD_RSP00	H'E804E80C	16
		SD_RSP01	H'E804E80E	16
		SD_RSP02	H'E804E810	16
		SD_RSP03	H'E804E812	16
		SD_RSP04	H'E804E814	16
		SD_RSP05	H'E804E816	16
		SD_RSP06	H'E804E818	16
		SD_RSP07	H'E804E81A	16
	SD card interrupt flag register 1	SD_INFO1	H'E804E81C	16
	SD card interrupt flag register 2	SD_INFO2	H'E804E81E	16
	SD_INFO1 interrupt mask register	SD_INFO1_MASK	H'E804E820	16
	SD_INFO2 interrupt mask register	SD_INFO2_MASK	H'E804E822	16
	SD clock control register	SD_CLK_CTRL	H'E804E824	16
	Transfer data length register	SD_SIZE	H'E804E826	16
	SD card access control option register	SD_OPTION	H'E804E828	16
	SD error status register 1	SD_ERR_STS1	H'E804E82C	16
	SD error status register 2	SD_ERR_STS2	H'E804E82E	16
	SD buffer read/write register	SD_BUF0	H'E804E830	32
	SDIO mode control register	SDIO_MODE	H'E804E834	16
	SDIO interrupt flag register	SDIO_INFO1	H'E804E836	16
	SDIO_INFO1 interrupt mask register	SDIO_INFO1_MASK	H'E804E838	16
	DMA mode enable register	CC_EXT_MODE	H'E804E8D8	16
	Software reset register	SOFT_RST	H'E804E8E0	16
	Version register	VERSION	H'E804E8E2	16
	Swap control register	EXT_SWAP	H'E804E8F0	16

chanhvinh.com

38.2.1 Command Type Register (SD_CMD)

The command type register (SD_CMD) selects the command type and response type. The command sequence is started by writing to SD_CMD. For details on the SD_CMD setting, refer to section 38.4.10, Example of SD_CMD Register Setting.

Do not write to SD_CMD while the SCLKDIVEN bit in SD_INFO2 is set to 0.

Bit	Bit Name	Initial Value	R/W	Description
15	MD7	0	R/W	Multiple Block Transfer Mode (enabled at multiple block transfer) 00: CMD12 is automatically issued at multiple block transfer. 01: CMD12 is not automatically issued at multiple block transfer. 10: Setting prohibited 11: Setting prohibited
14	MD6	0	R/W	
13	MD5	0	R/W	Single/Multiple Block Transfer (enabled when the command with data is handled) 0: Single block transfer 1: Multiple block transfer
12	MD4	0	R/W	Write/Read Mode (enabled when the command with data is handled) 0: Write (SD host interface → SD card) 1: Read (SD host interface ← SD card)
11	MD3	0	R/W	Data Mode (Command Type) 0: Command without data transfer (bc, bcr, ac) 1: Command with data transfer (adtc)
10	MD2	0	R/W	Mode/Response Type 000: Normal mode The response type and the transfer mode are selected by SD_CMD[7:0], and the SD_CMD[15:11] setting is disabled. 001: Setting prohibited 010: Setting prohibited 011: Extended mode/No response 100: Extended mode/SD card R1, R5, R6, R7 response 101: Extended mode/SD card R1b response 110: Extended mode/SD card R2 response 111: Extended mode/SD card R3, R4 response As some commands cannot be used in normal mode, see section 38.4.10, Example of SD_CMD Register Setting to select mode/response type.
9	MD1	0	R/W	
8	MD0	0	R/W	
7	C1	0	R/W	00: CMD 01: ACMD 10: Setting prohibited 11: Setting prohibited
6	C0	0	R/W	
5	CF45	0	R/W	Command Index These bits specify Command Format[45:40] (command index). [Examples] CMD6: SD_CMD[7:0] = 8'b00_000110 CMD18: SD_CMD[7:0] = 8'b00_010010 ACMD13: SD_CMD[7:0] = 8'b01_001101
4	CF44	0	R/W	
3	CF43	0	R/W	
2	CF42	0	R/W	
1	CF41	0	R/W	
0	CF40	0	R/W	

Note: • SD_CMD cannot be written to when the CBSY bit in SD_INFO2 is 1.

chanhvinh.com

38.2.2 SD Command Argument Registers (SD_ARG)

The SD command argument registers (SD_ARG) set the command argument of SD card. Set the command argument before writing to SD_CMD.

- SD_ARG0

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	CF23 to CF8	All 0	R/W	Set command format[23:8] (argument)

- SD_ARG1

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	CF39 to CF24	All 0	R/W	Set command format[39:24] (argument)

chanhvinh.com

38.2.3 Data STOP Register (SD_STOP)

The data stop register (SD_STOP) is used to enable or disable block counting at multiple block transfer, and to control the issuing of CMD12 within command sequences.

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	All 0	R	Fixed 0
8	SEC	0*1	R/W	Block Count Enable*2 0: Disables SD_SECCNT setting value. 1: Enables SD_SECCNT setting value. Set SEC to 1 at multiple block transfer. When SD_CMD is set as follows to start the command sequence while SEC is set to 1, CMD12 is automatically issued. 1. CMD18 or CMD25 in normal mode (SD_CMD[10:8] = 000) 2. SD_CMD[15:13] = 001 in extended mode (CMD12 is automatically issued, multiple block transfer) When the command sequence is halted because of a communications error or timeout, CMD12 is not automatically issued.
7 to 1	—	All 0	R	Fixed 0
0	STP	0*1	R/W	Stop*3 • When STP is set to 1 during multiple block transfer, CMD12 is issued to halt the transfer through the SD host interface. However, if a command sequence is halted because of a communications error or timeout, CMD12 is not issued. Although continued buffer access is possible even after STP has been set to 1, the buffer access error bit (ERR5 or ERR4) in SD_INFO2 will be set accordingly. • When STP has been set to 1 during transfer for single block write, the access end flag is set when SD_BUF becomes empty, and CMD12 is not issued. If SD_BUF does contain data, the access end flag is set on completion of reception of the busy state without CMD12 having been issued. • When STP has been set to 1 during transfer for single block read, the access end flag is set immediately after setting of the STP bit and CMD12 is not issued. • When STP is set to 1 during reception of the busy state after an R1b response, the access end flag is set on completion of reception of the busy state without CMD12 having been issued. • When STP is set to 1 after a command sequence has been completed, CMD12 is not issued and the access end flag is not set.

Note 1. The initial value is applied at a reset and when the SDRST bit in SOFT_RST is 0.
Note 2. Do not change the value of this bit when the SCLKDIVEN bit in SD_INFO2 is set to 0.
Note 3. Do not change the value of this bit from 1 to 0 when the SCLKDIVEN bit in SD_INFO2 is set to 0.

chanhvinh.com

38.2.4 Block Count Register (SD_SECCNT)

The block count register (SD_SECCNT) specifies the number of transfer blocks at multiple block transfer.

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	CNT15 to CNT0	All 0	R/W	Number of Transfer Blocks*1 When 0x0001 is set, the number of transfer blocks is 1. : When 0xFFFF is set, the number of transfer blocks is 65535. However, when 0x0000 is set, the number of transfer blocks is 65536.

Note 1. Do not change the value of this bit when the SCLKDIVEN bit in SD_INFO2 is set to 0.

38.2.5 SD Card Response Registers (SD_RSP)

The SD card response registers (SD_RSP) store the response from the SD card.

- SD_RSP00

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	R23 to R8	All 0	R	Store the response from the SD card

- SD_RSP01

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	R39 to R24	All 0	R	Store the response from the SD card

- SD_RSP02

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	R55 to R40	All 0	R	Store the response from the SD card

- SD_RSP03

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	R71 to R56	All 0	R	Store the response from the SD card

- SD_RSP04

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	R87 to R72	All 0	R	Store the response from the SD card

- SD_RSP05

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	R103 to R88	All 0	R	Store the response from the SD card

chanhvinh.com

- SD_RSP06

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	R119 to R104	All 0	R	Store the response from the SD card

- SD_RSP07

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Fixed 0
7 to 0	R127 to R120	All 0	R	Store the response from the SD card

Table 38.3 lists the response types and corresponding SD_RSP registers.

Table 38.3 Response Types and Corresponding SD_RSP Registers

Response Types	SD_RSP Registers
R1, R1b[39:8]	SD_RSP01 and SD_RSP00
R2[127:8]	SD_RSP07 to SD_RSP00
R3[39:8]	SD_RSP01 and SD_RSP00
R4[39:8]	SD_RSP01 and SD_RSP00
R5[39:8]	SD_RSP01 and SD_RSP00
R6[39:8]	SD_RSP01 and SD_RSP00
R7[39:8]	SD_RSP01 and SD_RSP00

chanhvinh.com

38.2.6 SD Card Interrupt Flag Register 1 (SD_INFO1)

The SD card interrupt flag register 1 (SD_INFO1) indicates the response end and access end in the command sequence. This register also indicates the card detect/write protect state.

For CMD12 and CMD52 (SDIO abort) at multiple block transfer, INFO0 is not set but only INFO2 is set.

Even if the command sequence is halted because of a communications error or timeout, INFO0 or INFO2 is set.

INFO10, INFO9, and INFO8 change depending on the SD_D3 state after a reset is released and continue to change in 4-bit transfer mode.

To clear a flag, write 0 to the bit to be cleared and 1 to the other bits.

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Fixed 0
10	INFO10	Unknown	R	Indicates the SD_D3 state. 1: SD_D3 is set to 1. 0: SD_D3 is set to 0.
9	INFO9	0	R/W*1	SD_D3 set [Setting condition] After change in SD_D3 from 0 to 1, two cycles of P1φ has elapsed with SD_D3 held 1. [Clearing condition] When 0 is written to INFO9
8	INFO8	0	R/W*1	SD_D3 clear [Setting condition] After change in SD_D3 from 1 to 0, two cycles of P1φ has elapsed with SD_D3 held 0. [Clearing condition] When 0 is written to INFO8
7	INFO7	Unknown	R	Write Protect Indicates the SD_WP state. 1: SD_WP is set to 0. 0: SD_WP is set to 1.
6	—	0	R	Fixed 0
5	INFO5	Unknown	R	Indicates the SD_CD state. 1: Indicates that Ncycle has elapsed with SD_CD held 0. 0: Indicates that Ncycle has elapsed with SD_CD held 1. Ncycle is set by bits 3 to 0 in SD_OPTION.
4	INFO4	0	R/W*1	SD_CD Card Insertion [Setting condition] After change in SD_CD from 1 to 0, Ncycle has elapsed with SD_CD held 0. [Clearing condition] When 0 is written to INFO4 Ncycle is set by bits 3 to 0 in SD_OPTION.
3	INFO3	0	R/W*1	SD_CD Card Removal [Setting condition] After change in SD_CD from 0 to 1, Ncycle has elapsed with SD_CD held 1. [Clearing condition] When 0 is written to INFO3 Ncycle is set by bits 3 to 0 in SD_OPTION.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
2	INFO2	0*2	R/W*1	Access End [Setting conditions] • When read access to the buffer is completed in the case of transfer for single block read • When read access to the buffer for the last block of data is completed in the case of transfer for multiple block read • When read access to the buffer and reception of the response to CMD12 are completed in the case of transfer for multiple block read with automatic issuing of CMD12 • When reception of the busy state after reception of the CRC status is completed in the case of transfer for single block write • When reception of the busy state after reception of the CRC status of the last block of data is completed in the case of transfer for multiple block write • When reception of the response busy state for CMD12 is completed in the case of transfer for multiple block write with automatic issuing of CMD12 • When reception of the response to CMD12 that was issued by setting the STP bit to1 is completed in the case of transfer for multiple block read • When reception of the response busy state for CMD12 that was issued by setting the STP bit to1 is completed in the case of transfer for multiple block write • When reception of the response to CMD52 that was issued by setting the IOABT bit to1 is completed in the case of transfer for multiple block read • When reception of the response to CMD52 that was issued by setting the IOABT bit to1 is completed in the case of transfer for multiple block write In addition to the above conditions, this bit is set when a command sequence is halted because of a communications error or timeout. [Clearing condition] When 0 is written to INFO2
1	—	0	R	Fixed 0
0	INFO0	0*2	R/W*1	Response End [Setting conditions] • When reception of the response is completed • When transmission of a command without response is completed • When reception of the busy state after R1b response is completed • When reception of the response to CMD52 that was issued by setting the C52PUB bit to1 is completed in the case of transfer for multiple block read • When reception of the response to CMD52 that was issued by setting the C52PUB bit to1 is completed in the case of transfer for multiple block write In addition to the above conditions, this bit is set when a command sequence is halted because of a communications error or timeout. [Clearing condition] When 0 is written to INFO0

Note 1. It is effective only if 0 is written.
Note 2. The initial value is applied at a reset and when the SDRST bit in SOFT_RST is 0.

chanhvinh.com

38.2.7 SD Card Interrupt Flag Register 2 (SD_INFO2)

The SD card interrupt flag register 2 (SD_INFO2) indicates the access status of the SD buffer (SD_BUF) and SD card. To clear a flag, write 0 to the bit to be cleared and 1 to the other bits.

Bit	Bit Name	Initial Value	R/W	Description
15	ILA	0*2	R/W*1	Illegal Access Error [Setting conditions] • When data is written to SD_CMD within a command sequence • When SD_CMD[10:8] = 3'b011 (no response) and SD_CMD[11] = 1'b1 (command with data transfer) are set in SD_CMD • When SD_CMD[11] = 1'b1 (command with data transfer) and SD_CMD[7:0] = 8'b00001100 (CMD12) are set in SD_CMD [Clearing condition] When 0 is written to ILA
14	CBSY	0*2	R	Command Type Register Busy 1: The command sequence is being executed. 0: The command sequence is ended.
13	SCLKDIVEN	1*2	R	SD Bus Busy 1: SD buses (CMD and DAT) are not busy. 0: SD buses (CMD and DAT) are busy. When a command sequence is started by writing to SD_CMD, CBSY is set to 1. At the same time, SCLKDIVEN is set to 0. After CBSY has been set to 0 at the end of the command sequence, SCLKDIVEN is set to 1 when 8 cycles of SD_CLK elapses.
12	—	0	R	Fixed 0
11	—	0*2	R/W*1	Reserved. The write value should always be 1.
10	—	0	R	Fixed 0
9	BWE	0*2	R/W*1	SD_BUF Write Enable 1: Data can be written in SD_BUF0. 0: Data cannot be written in SD_BUF0. [Setting conditions] • When SD_BUF is empty at single block transfer • When either bank 1 or bank 2 of SD_BUF is empty at multiple block transfer [Clearing condition] • When 0 is written to BWE When data is written to SD_BUF0 by the CPU, clear BWE and then write amount of data specified by SD_SIZE*3. Note that this bit is not set when the SD_BUF read/write DMA transfer is enabled by setting the DMASDRW bit in CC_EXT_ MODE to 1.
8	BRE	0*2	R/W*1	SD_BUF Read Enable 1: Data can be read from SD_BUF0. 0: Data cannot be read from SD_BUF0. [Setting conditions] • When data set in SD_SIZE is stored in SD_BUF at single block transfer • When data set in SD_SIZE is stored in either bank 1 or bank 2 of SD_BUF at multiple block transfer [Clearing condition] • When 0 is written to BRE When data is read from SD_BUF0 by the CPU, clear BRE and then read amount of data specified by SD_SIZE*3. Note that this bit is not set when the SD_BUF read/write DMA transfer is enabled by setting the DMASDRW bit in CC_EXT_ MODE to 1. Even if a CRC error or an END error occurs while block data is read, data is stored in SD_BUF and BRE is set.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
7	DAT0	Unknown	R	SD_D0 Indicates the SD_D0 state 1: SD_D0 is set to 1. 0: SD_D0 is set to 0. If the data timeout (ERR3) is set but the response timeout (ERR6) is not set after the Erase command has been issued, the end of the Erase sequence (DAT0 = 1) is confirmed by polling DAT0. If a communications error or timeout occurs during a write sequence, the DAT0 bit may retain the value 0.
6	ERR6	0*2	R/W*1	Response Timeout [Setting condition] When a response is not received though a longer time than 640 cycles of SD_CLK has elapsed [Clearing condition] When 0 is written to ERR6 The command sequence is halted by the response timeout.*4
5	ERR5	0*2	R/W*1	SD_BUF Illegal Read Access [Setting conditions] • When SD_BUF is empty while SD_BUF0 is read • When data with a CRC error or END error is read from SD_BUF0 [Clearing condition] When 0 is written to ERR5
4	ERR4	0*2	R/W*1	SD_BUF Illegal Write Access [Setting conditions] • When data is written to SD_BUF0 while it is not in the data read/write command state • When data is written to SD_BUF0 while SD_BUF is full. • When data is written to SD_BUF0 while an error occurs in the CRC status or CRC status length. • When data is written to SD_BUF0 while a busy state after the CRC status continues for longer than Ncycle [Clearing condition] When 0 is written to ERR4 Ncycle is set by bits 7 to 4 in SD_OPTION.
3	ERR3	0*2	R/W*1	Data Timeout (except response timeout) [Setting conditions] • After R1b response, the busy state (SD_D0 = 0) continues for longer than Ncycle. • After CRC status, the busy state (SD_D0 = 0) continues for longer than Ncycle. • After write data, the CRC status is not received though Ncycle has elapsed. • After read command, read data is not received though a longer time than Ncycle has elapsed. • After CMD12 has been issued within a command sequence, the busy state (SD_D0 = 0) for longer than Ncycle continues. • After the reception of read data, read data for the next block are not received though a longer time than Ncycle has elapsed. • After release of the read wait state, read data for the next block are not received though a longer time than Ncycle has elapsed. [Clearing condition] • When 0 is written to ERR3 Ncycle is set by bits 7 to 4 in SD_OPTION. The command sequence is halted by the data timeout.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
2	ERR2	0*2	R/W*1	END Error [Setting conditions] • When an error occurs in the response length (and the end bit has not been detected) • When an error occurs in the read data length (and the end bit has not been detected among the valid bits) • When an error occurs in the CRC status length (and the end bit has not been detected) [Clearing condition] When 0 is written to ERR2 The command sequence is halted by the End error.*4
1	ERR1	0*2	R/W*1	CRC Error [Setting conditions] • When an error occurs in the CRC status • When a CRC error occurs in the read data • When a CRC error occurs in the response [Clearing condition] When 0 is written to ERR1 The command sequence is halted by the CRC error.*4
0	ERR0	0*2	R/W*1	CMD Error [Setting condition] When the command index of the transmitted command differed from the command index of the received response [Clearing condition] When 0 is written to ERR0 The command sequence is halted by the CMD error.*4

Note 1. It is effective only if 0 is written.
Note 2. The initial value is applied at a reset and when the SDRST bit in SOFT_RST is 0.
Note 3. The single byte or three bytes from the fraction of a full 32-bit unit are regarded as excess data due to an odd value for the number of bytes setting in SD_SIZE, or the two bytes from the fraction of a full 32-bit unit are regarded as excess data due if the value for the number of bytes setting in SD_SIZE is even but is not on a four-byte boundary.
Note 4. After the C52PUB bit in SDIO_MODE has been set to 1, if a communications error or timeout for response occurs in response to the CMD52 that is issued, since the command sequence has not been completed, complete the sequence with error processing as in usage examples in Figure 38.17 under section 38.4.7, IO_RW_EXTENDED Command (CMD53/Multiple Block Read) or in Figure 38.20 under section 38.4.8, IO_RW_EXTENDED Command (CMD53/Multiple Block Write).

38.2.8 SD_INFO1 Interrupt Mask Register (SD_INFO1_MASK)

The SD_INFO1 interrupt mask register (SD_INFO1_MASK) enables or disables the SD_INFO1 interrupt. When 0 is set in SD_INFO1_MASK while the corresponding flag in SD_INFO1 is set, an interrupt occurs.

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Fixed 0
9	IMASK9	1	R/W	INFO9 interrupt masked
8	IMASK8	1	R/W	INFO8 interrupt masked
7 to 5	—	All 0	R	Fixed 0
4	IMASK4	1	R/W	INFO4 interrupt masked
3	IMASK3	1	R/W	INFO3 interrupt masked
2	IMASK2	1	R/W	INFO2 interrupt masked
1	—	0	R	Fixed 0
0	IMASK0	1	R/W	INFO0 interrupt masked

chanhvinh.com

38.2.9 SD_INFO2 Interrupt Mask Register (SD_INFO2_MASK)

The SD_INFO2 interrupt mask register (SD_INFO2_MASK) enables or disables the SD_INFO2 interrupt. When 0 is set in SD_INFO2_MASK while the corresponding flag in SD_INFO2 is set, an interrupt occurs.

Bit	Bit Name	Initial Value	R/W	Description
15	IMASK	1	R/W	ILA interrupt masked
14 to 12	—	All 0	R	Fixed 0
11	—	1	R/W	Reserved. The write value should always be 1.
10	—	0	R	Fixed 0
9	BMASK1	1	R/W	BWE interrupt masked
8	BMASK0	1	R/W	BRE interrupt masked
7	—	0	R	Fixed 0
6	EMASK6	1	R/W	ERR6 interrupt masked
5	EMASK5	1	R/W	ERR5 interrupt masked
4	EMASK4	1	R/W	ERR4 interrupt masked
3	EMASK3	1	R/W	ERR3 interrupt masked
2	EMASK2	1	R/W	ERR2 interrupt masked
1	EMASK1	1	R/W	ERR1 interrupt masked
0	EMASK0	1	R/W	ERR0 interrupt masked

chanhvinh.com

38.2.10 SD Clock Control Register (SD_CLK_CTRL)

The SD clock control register (SD_CLK_CTRL) controls the SD clock (SD_CLK) output and sets the frequency. Set SCLKEN to 1 before writing to SD_CMD to issue a command.
Do not write to SD_CLK_CTRL while the SCLKDIVEN bit in SD_INFO2 is set to 0.

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Fixed 0
9	SDCLKOFFEN	0	R/W	SD Clock (SD_CLK) Output Automatic Control Enable 0: Automatic control for SD clock (SD_CLK) output is disabled. 1: Automatic control for SD clock (SD_CLK) output is enabled. This function of automatic control for SD clock (SD_CLK) output causes SD_CLK output only within a command sequence. The timing with which SD_CLK output starts and stops is as follows. SD_CLK output starts after writing to SD_CMD. SD_CLK output stops when 8 cycles of SD_CLK have elapsed after the end of the command sequence. In addition, automatic control for SD clock (SD_CLK) output is enabled when SCLKEN in SD_CLK_CTRL is 1.
8	SCLKEN	0*1	R/W*2	SD Clock (SD_CLK) Output Control Enable 0: SD clock (SD_CLK) output is disabled. The SD_CLK signal is fixed 0. 1: SD clock (SD_CLK) output is enabled.
7	DIV7	0	R/W*2	SD Clock (SD_CLK) 10000000: P1φ/512 01000000: P1φ/256 00100000: P1φ/128 00010000: P1φ/64 00001000: P1φ/32 00000100: P1φ/16 00000010: P1φ/8 00000001: P1φ/4 00000000: P1φ/2 Other settings are prohibited.
6	DIV6	0	R/W*2	
5	DIV5	1	R/W*2	
4	DIV4	0	R/W*2	
3	DIV3	0	R/W*2	
2	DIV2	0	R/W*2	
1	DIV1	0	R/W*2	
0	DIV0	0	R/W*2	

Note 1. This initial value is applied at a reset and when the SDRST bit in SOFT_RST is 0.
Note 2. Writing is impossible when the CBSY bit in SD_INFO2 is 1.

38.2.11 Transfer Data Length Register (SD_SIZE)

The transfer data length register (SD_SIZE) specifies the transfer data size.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Fixed 0
11, 10	—	All 0	R	Reserved
9 to 0	LEN9 to LEN0	1000000000	R/W	Transfer Data Size*1 These bits specify a size between 1 and 512 bytes for the transfer of single blocks. In cases of multiple block transfer with automatic issuing of CMD12 (CMD18 and CMD25), the only specifiable transfer data size is 512 bytes. Furthermore, in cases of multiple block transfer without automatic issuing of CMD12, as well as 512 bytes, 32, 64, 128, and 256 bytes are specifiable. However, in the reading of 32, 64, 128, and 256 bytes for the transfer of multiple blocks, this is restricted to cases of multiple block transfer by CMD53. Additionally, if a command accompanies data transfer, do not set these bits to 0.

Note 1. Do not change the values of these bits when the SCLKDIVEN bit in SD_INFO2 is 0.

chanhvinh.com

38.2.12 SD Card Access Control Option Register (SD_OPTION)

The SD card access control option register (SD_OPTION) sets the bus width and timeout counter.

Bit	Bit Name	Initial Value	R/W	Description
15	WIDTH	0*1	R/W	Bus Width*2 0: 4-bit width 1: 1-bit width
14	—	1	R	Fixed 1
13 to 8	—	All 0	R	Fixed 0
7	TOP27	1*1	R/W	Timeout Counter*2 1111: Setting prohibited 1110: SD_CLK × 2^{27} 1101: SD_CLK × 2^{26} : 0001: SD_CLK × 2^{14} 0000: SD_CLK × 2^{13}
6	TOP26	1*1	R/W	
5	TOP25	1*1	R/W	
4	TOP24	0*1	R/W	
3	CTOP24	1*1	R/W	Card Detect Time Counter 1111: Setting prohibited 1110: P1φ × 2^{24} 1101: P1φ × 2^{23} : 0001: P1φ × 2^{11} 0000: P1φ × 2^{10}
2	CTOP23	1*1	R/W	
1	CTOP22	1*1	R/W	
0	CTOP21	0*1	R/W	

Note 1. The initial value is applied at a reset and when the SDRST bit in SOFT_RST is 0.
Note 2. Do not change the values of these bits when the SCLKDIVEN bit in SD_INFO2 is 0.

chanhvinh.com

38.2.13 SD Error Status Register 1 (SD_ERR_STS1)

The SD error status register 1 (SD_ERR_STS1) indicates the CRC status, CRC error, End error, and CMD error.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Fixed 0
14	E14	0*1	R	These bits store the CRC status. (normal: 010)
13	E13	1*1	R	
12	E12	0*1	R	
11	E11	0*1	R	Set to 1 when an error occurs in the CRC status
10	E10	0*1	R	Set to 1 when a CRC error occurs in the read data
9	E9	0*1	R	Set to 1 when a CRC error occurs in the response to a command*2 issued within a command sequence. In cases where CMD12 is issued by setting a command index in SD_CMD, this is indicated in E8.
8	E8	0*1	R	Set to 1 when a CRC error occurs in a response (other than a response to a command*2 issued within a command sequence)
7, 6	—	All 0	R	Fixed 0
5	E5	0*1	R	Set to 1 when an error occurs in the CRC status length (and the end bit has not been detected)
4	E4	0*1	R	Set to 1 when an error occurs in the read data length (and the end bit has not been detected among the valid bits)
3	E3	0*1	R	Set to 1 when an error occurs in the response length to a command*2 issued within a command sequence. In cases where CMD12 is issued by setting a command index in SD_CMD, this is indicated in E2.
2	E2	0*1	R	Set to 1 when an error occurs in the response length (other than a response to a command*2 issued within a command sequence)
1	E1	0*1	R	Set to 1 when an error occurs in the command index of the response to a command*2 issued within a command sequence. In cases where CMD12 is issued by setting a command index in SD_CMD, this is Indicated in E0.
0	E0	0*1	R	Set to 1 when an error occurs in the command index of a response (other than a response to a command*2 issued within a command sequence).

Note 1. The initial value is applied at a reset and when the SDRST bit in SOFT_RST is 0.
Note 2. "Command issued within a command sequence" refers to CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

chanhvinh.com

38.2.14 SD Error Status Register 2 (SD_ERR_STS2)

The SD error status register 2 (SD_ERR_STS2) indicates the timeout state. Ncycle is set by bits 7 to 4 in SD_OPTION.

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	All 0	R	Fixed 0
6	E6	0*1	R	Set to 1 when the busy state continues for longer than Ncycle after the CRC status
5	E5	0*1	R	Set to 1 when the CRC status is not received though a longer time than Ncycle has elapsed after data writing
4	E4	0*1	R	Set to 1 when read data is not received though a longer time than Ncycle has elapsed after read command. Set to 1 when read data for the next block are not received though a longer time than Ncycle has elapsed after the reception of read data. Set to 1 when read data for the next block are not received though a longer time than Ncycle has elapsed after release of the read wait state.
3	E3	0*1	R	Set to 1 when the busy state for longer than Ncycle continues after CMD12 has been issued within a command sequence. In cases where CMD12 is issued by setting a command index in SD_CMD, this is indicated in E2.
2	E2	0*1	R	Set to 1 when the busy state for longer than Ncycle continues after R1b response
1	E1	0*1	R	Set to 1 when the response to a command*2 issued within a command sequence is not received though a longer time than 640 cycles of SD_CLK has elapsed. In cases where CMD12 is issued by setting a command index in SD_CMD, this is indicated in E0.
0	E0	0*1	R	Set to 1 when the response (other than a response to a command*2 issued within a command sequence) is not received though a longer time than 640 cycles of SD_CLK has elapsed.

Note 1. The initial value is applied at a reset and when the SDRST bit in SOFT_RST is 0.
Note 2. "Command issued within a command sequence" refers to CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

38.2.15 SD Buffer Read/Write Register (SD_BUF0)

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BUF31 to BUF0	Unknown	R/W	When writing to the SD card, the write data is written to this register. When reading from the SD card, the read data is read from this register. This register is internally connected to two 512-byte (128-word) buffers.

chanhvinh.com

38.2.16 SDIO Mode Control Register (SDIO_MODE)

The SDIO mode control register (SDIO_MODE) controls the CMD52 issuance and the read wait state at multiple block transfer, and the reception of SDIO interrupt. C52PUB and IOABT should not be set to 1 simultaneously.

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Fixed 0
9	C52PUB	0	R/W	SDIO None Abort*2 • When C52PUB is set to 1 in the CMD53 (multiple block) write sequence, CMD52 is automatically issued if SD_BUF becomes empty and between the current one-block and next-block. C52PUB is automatically cleared to 0 after reception of the response to CMD52 is completed. Additionally, if C52PUB is set to 1 while the last block is being transferred, CMD52 is not issued. In this case, C52PUB is automatically cleared to 0 after the access end flag has been set to 1. • When C52PUB and RWREQ are set to 1 in the CMD53 (multiple block) read sequence, the block transfer enters the read wait state between the current one-block and next-block and CMD52 is automatically issued. C52PUB is automatically cleared to 0 after reception of the response to CMD52 is completed. Additionally, if C52PUB is set to 1 while the last block is being transferred, CMD52 is not issued. In this case, C52PUB is automatically cleared to 0 after the access end flag has been set to 1. • If C52PUB is set to 1 in the CMD53 (multiple block) read sequence, be sure to set RWREQ to 1 as well as C52PUB. • Set SD_ARG before setting C52PUB to 1.
8	IOABT	0	R/W	SDIO Abort*2 • When IOABT is set to 1 in the CMD53 (multiple block) sequence, the CMD53 sequence is halted and CMD52 is issued. However, if a command sequence is halted because of a communications error or timeout, CMD52 is not issued. Although continued buffer access is possible even after IOABT has been set to 1, the buffer access error bit (ERR5 or ERR4) in SD_INFO2 will be set accordingly. Set SD_ARG before setting IOABT to 1. • When IOABT has been set to 1 during transfer for single block write, the access end flag is set when SD_BUF becomes empty, and CMD52 is not issued. If SD_BUF does contain data, the access end flag is set on completion of reception of the busy state without CMD52 having been issued. • When IOABT has been set to 1 during transfer for single block read, the access end flag is set immediately after setting of IOABT and CMD52 is not issued. • When IOABT is set to 1 during reception of the busy state after an R1b response, the access end flag is set on completion of reception of the busy state without CMD52 having been issued. • When IOABT is set to 1 after a command sequence has been completed, CMD52 is not issued and the access end flag is not set.
7 to 3	—	All 0	R	Fixed 0

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
2	RWREQ	0	R/W	Read Wait Request When RWREQ is set to 1 in the CMD53 (multiple block) read sequence, the block transfer enters the read wait state (SD_D2 changes from 1 to 0) between the current one-block and next-block. [Read wait state releasing] • The read wait state is released, when RWREQ is cleared to 0 in the read wait state.*3 • When IOABT is set to 1 in the read wait state, RWREQ is automatically cleared to 0 after CMD52 has been issued, and then the read wait state is released. • When C52PUB and RWREQ are set to 1 simultaneously in the CMD53 (multiple block) read sequence, the read wait state is not automatically released. Therefore, after the CMD52 response is received, clear RWREQ. (Be sure to set RWREQ and C52PUB simultaneously.) When RWREQ is set to 1 while the last block in the CMD53 (multiple block) read sequence is transferred, the read wait state is not entered and RWREQ is automatically cleared to 0 by setting access end.
1	—	0	R	Fixed 0
0	IOMOD	0	R/W	SDIO Mode*1 1: Enables the SD host interface to receive SDIO interrupt from the SDIO card. 0: Disables the SD host interface to receive SDIO interrupt from the SDIO card.

Note 1. Do not change the value of this bit when the SCLKDIVEN bit in SD_INFO2 is set to 0.
Note 2. Do not change the values of these bits from 1 to 0 when the SCLKDIVEN bit in SD_INFO2 is set to 0.
Note 3. Clear this bit after reception of the response to CMD52 is completed.

chanhvinh.com

38.2.17 SDIO Interrupt Flag Register (SDIO_INFO1)

The SDIO interrupt flag register (SDIO_INFO1) indicates the status regarding to the SDIO card access. To clear a flag, write 0 to the bit to be cleared and 1 to the other bits.

Bit	Bit Name	Initial Value	R/W	Description
15	EXWT	0*2	R/W*1	[Setting condition] While the last block in the CMD53 (multiple block) read sequence is transferred, RWREQ in SDIO_MODE is set to 1. [Clearing condition] When 0 is written to EXWT
14	EXPUB52	0*2	R/W*1	[Setting conditions] • While the last block in the CMD53 (multiple block) sequence is transferred, C52PUB in SDIO_MODE is set to 1. • While C52PUB is set to 1 in the CMD53 (multiple block) write sequence, the last block is transferred. [Clearing condition] When 0 is written to EXPUB52
13 to 3	—	All 0	R	Fixed 0
2, 1	—	0*2	R/W*1	Reserved The write value should always be 1.
0	IOIRQ	0*2	R/W*1	SD IO Interrupt [Setting condition] When SDIO interrupt from an SDIO card is received while IOMOD in SDIO_MODE is set to 1 [Clearing condition] When 0 is written to IOIRQ*3

Note 1. It is effective only if 0 is written.
Note 2. The initial value is applied at a reset and when the SDRST bit in SOFT_RST is 0.
Note 3. Before clearing this bit, access the SDIO card to negate the SDIO interrupt signal from the SDIO card (after CMD25 has been issued and the response received). If the interrupt signal is not negated, this bit may be set again.

chanhvinh.com

38.2.18 SDIO_INFO1 Interrupt Mask Register (SDIO_INFO1_MASK)

The SDIO_INFO1 interrupt mask register (SDIO_INFO1_MASK) enables or disables the SD_INFO1 interrupt. When 0 is set in SDIO_INFO1_MASK while the corresponding flag in SD_INFO1 is set, an interrupt occurs.

Bit	Bit Name	Initial Value	R/W	Description
15	MEXWT	1	R/W	EXWT interrupt masked
14	MEXPUB52	1	R/W	EXPUB52 interrupt masked
13 to 3	—	All 0	R	Fixed 0
2, 1	—	All 1	R/W	Reserved The write value should always be 1.
0	IOMSK	1	R/W	IOIRQ interrupt masked

38.2.19 DMA Mode Enable Register (CC_EXT_MODE)

The DMA mode enable register (CC_EXT_MODE) enables the DMA transfer.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Fixed 0
12	—	1	R	Reserved
11, 10	—	All 0	R	Fixed 0
9, 8	—	All 0	R	Reserved
7 to 5	—	All 0	R	Fixed 0
4	—	1	R	Reserved
3, 2	—	All 0	R	Reserved
1	DMASDRW	0	R/W	SD_BUF Read/Write DMA Transfer*1 1: The SD_BUF read/write DMA transfer is enabled. 0: The SD_BUF read/write DMA transfer is disabled. At the SD_BUF read/write DMA transfer, set DMASDRW to 1 before setting SD_CMD.
0	—	0	R	Reserved

Note 1. Do not change the values of these bits when the SCLKDIVEN bit in SD_INFO2 is set to 0.

chanhvinh.com

38.2.20 Software Reset Register (SOFT_RST)

The software reset register (SOFT_RST) sets a software reset. Make sure to release the reset before using the SD host interface.

Bit	Bit Name	Initial Value	R/W	Description
15 to 3	—	All 0	R	Fixed 0
2	—	1	R	Reserved
1	—	1	R	Reserved
0	SDRST	0	R/W	Software Reset of SD I/F Unit 0: Reset 1: Reset released

38.2.21 Version Register (VERSION)

The version register (VERSION) indicates the version of the SD host interface.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	1000	R	Reserved
11 to 8	UR3 to UR0	4'h2	R	Version of Renesas IP
7 to 0	IP7 to IP0	8'h0B	R	Version of adopted IP

chanhvinh.com

38.2.22 Swap Control Register (EXT_SWAP)

The swap control register (EXT_SWAP) can replace data when SD_BUF0 is accessed.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R/W	Reserved The write value should always be 0.
14, 13	—	All 0	R	Reserved
12, 11	—	All 0	R/W	Reserved The write value should always be 0.
10, 9	—	All 0	R	Fixed 0
8	DMASEL	0	R/W	DMA Transfer Size Select Selects the transfer unit for SD_BUF read/write DMA transfer. Set this bit in combination with the transfer size set in the DMA Channel Configuration register. 0: 4-byte (longword) unit 1: 64-byte (longword × 16) unit Note: • For DMA transfer in 64-byte units, set address H'E804E000 (channel 0) / H'E804E800 (channel 1) as the destination or source of DMA transfer instead of the SD buffer read/write register (SD_BUF0). Moreover, if a communications error or timeout occurs during the 64-byte unit DMA transfer, this bit should be set to 0 and re-setting again before using DMA transfer. If a software reset occurs during the 64-byte unit DMA transfer, this bit should be set to 0 and re-setting again before using DMA transfer.
7	SDBRSWAP	0	R/W	SD_BUF0 Swap Read*1 When reading from SD_BUF0, data stored in SD_BUF0 can be replaced and then read.*2 0: The current data is read without replacement. 1: Data replacement for reading proceeds in bytes.
6	SDBWSWAP	0	R/W	SD_BUF0 Swap Write*1 When writing to SD_BUF0, data to be written can be replaced and then stored in SD_BUF0.*2 0: The current data is written without replacement. 1: Data replacement for writing proceeds in bytes.
5	—	0	R	Fixed 0
4, 3	—	All 0	R/W	Reserved The write value should always be 0.
2	—	0	R	Fixed 0
1	—	0	R/W	Reserved The write value should always be 0.
0	—	0	R	Fixed 0

Note 1. Do not change the values of these bits when the SCLKDIVEN bit in SD_INFO2 is set to 0.
Note 2. Details on SD_BUF0 access with data replacement or non-replacement are indicated in (1) SD Data Format in section 38.3.1, SD I/F.

chanhvinh.com

38.3 Operation

38.3.1 SD I/F

(1) SD Data Format

When data is read from the SD card, the procedure is as follows.

1. The SD host interface receives data from the SD card via the SD_D signal. (SD_D signal: see Figure 38.2 and Figure 38.3.)
2. The receive data is stored in SD_BUF of the SD host interface. (SD_BUF store data: see Figure 38.4)
3. The data stored in SD_BUF is read by SD_BUF0. (Reading from SD_BUF0: see Figure 38.5)

When data is written to the SD card, the above procedure will be reversed.
When accessing SD_BUF0, caution should be taken for the transfer order in SD_D and the store order in SD_BUF. In addition, data stored in SD_BUF0 can be replaced in bytes with the EXT_SWAP. (See Figure 38.5)

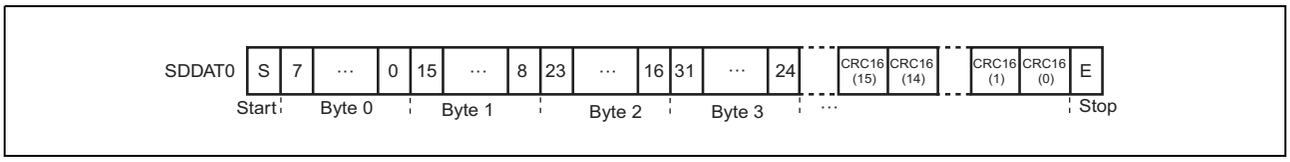


Figure 38.2 SD_D0 in 1-Bit Width Mode

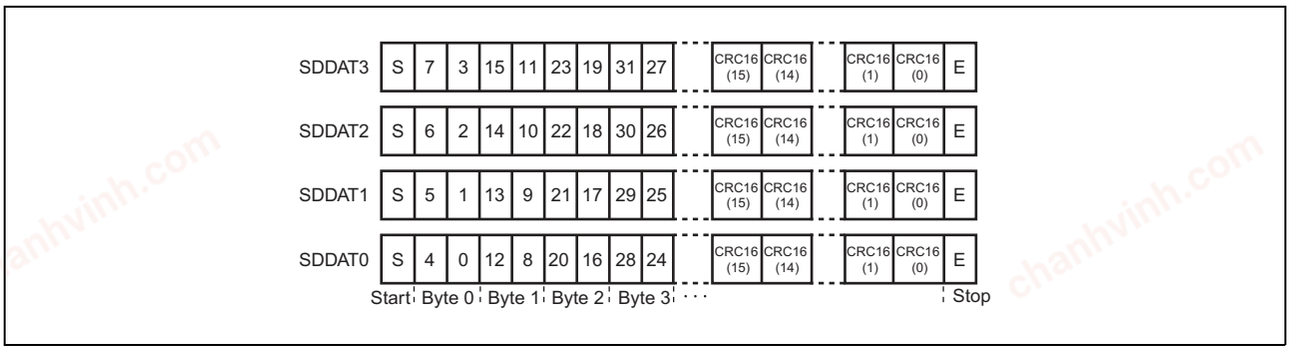


Figure 38.3 SD_D3 to SD_D0 in 4-Bit Width Mode

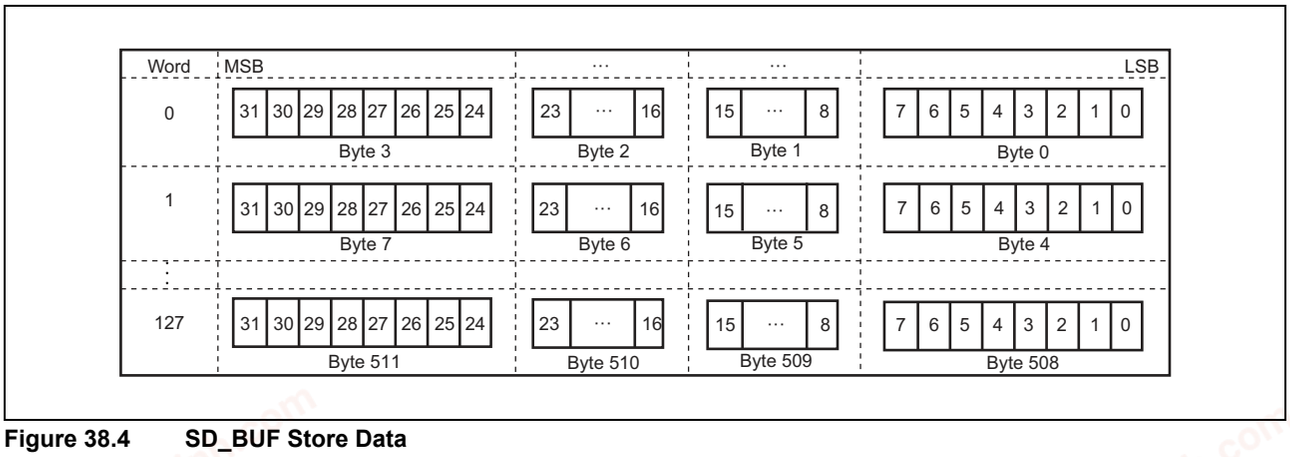


Figure 38.4 SD_BUF Store Data

chanhvinh.com

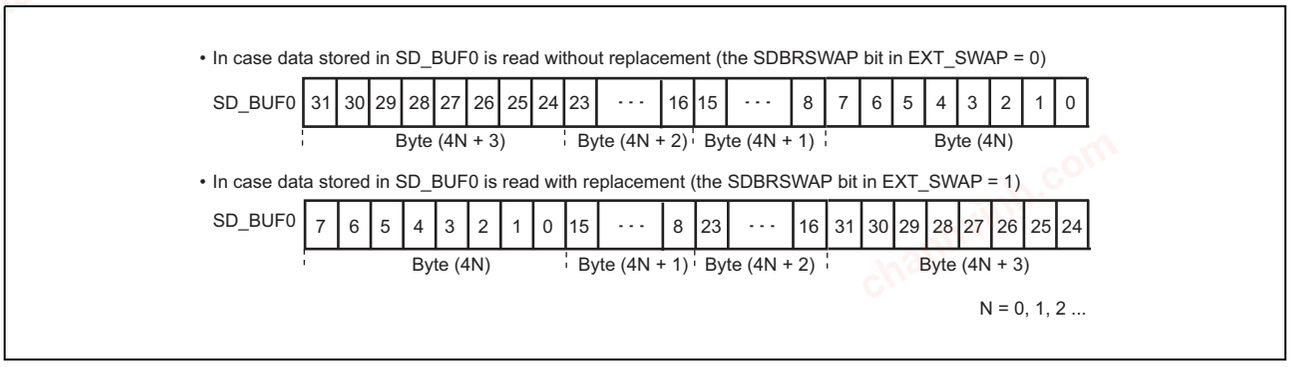


Figure 38.5 Reading from SD_BUF0

chanhvinh.com

38.3.2 Card Detect/Write Protect

(1) Card Detect

The SD host interface has two types of card detect functions as described in the following.

- Card detect with SD_CD
  Figure 38.6 shows the timing chart of card detect using SD_CD. SD_CD is connected to the card socket and pulled up on the host device. The resistance of the pull-up resistor is decided by the specification of the SD host device.

[Card insertion]
SD_CD is pulled down when a card is inserted. At this time, if SD_CD has been pulled down for the Ncycle period (set in SD_OPTION), INFO4 in SD_INFO1 is set to 1. (It is cleared to by writing 0.)
[Card removal]
SD_CD is pulled up when a card is removed. At this time, if SD_CD has been pulled up for the Ncycle period (set in SD_OPTION), INFO3 in SD_INFO1 is set to 1. (It is cleared to by writing 0.)

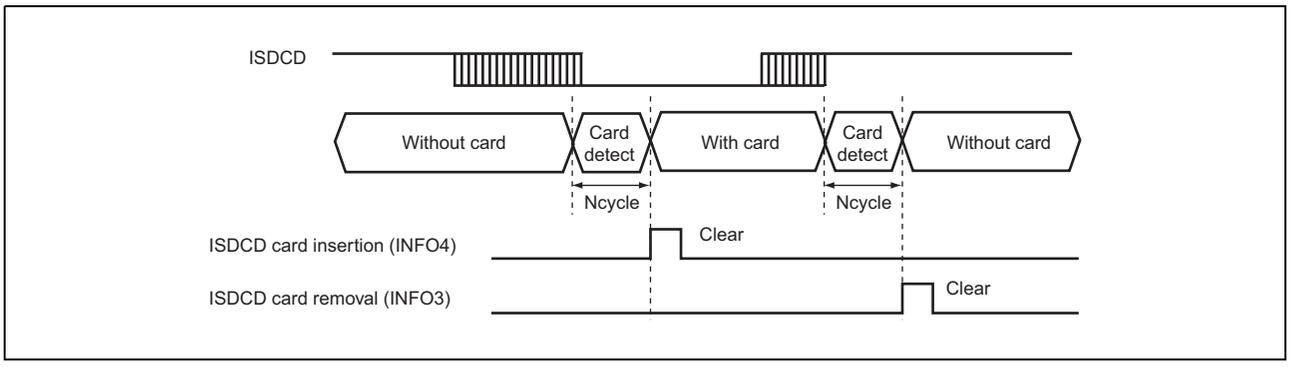


Figure 38.6 Example of Card Detect with SD_CD

- SD card detect with SD_D3
  Figure 38.7 shows the timing chart when the SD card is detected with SD_D3. In addition, SD_D3 is pulled down by the host device, and the resistance value for pulling down is decided by the specification of SD host device.

[Card insertion]
When an SD card is inserted, SD_D3 is pulled up. Accordingly, INFO9 in SD_INFO1 is set to 1. (It is cleared to by writing 0.)
[Card removal]
When an SD card is removed, SD_D3 is pulled down. Accordingly, INFO8 in SD_INFO1 is set to 1. (It is cleared to by writing 0.)

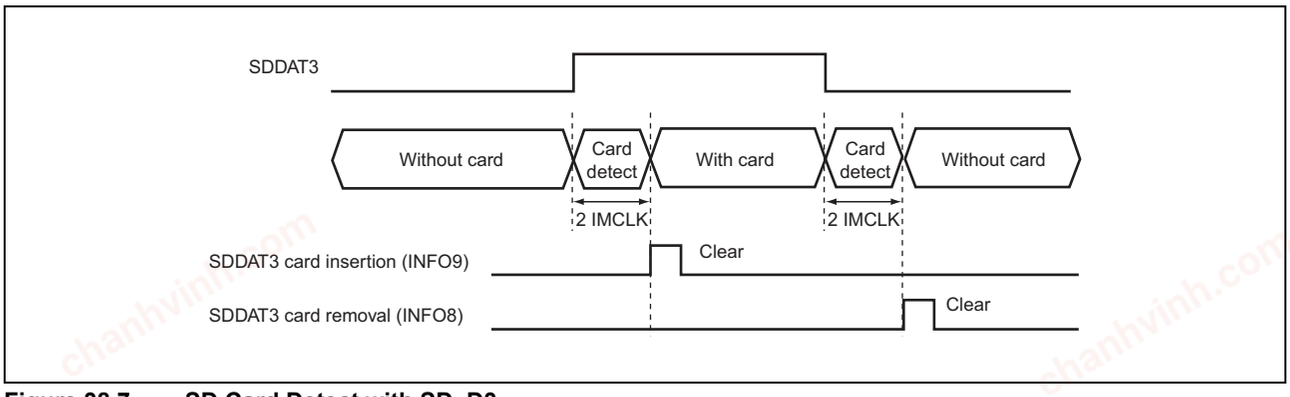


Figure 38.7 SD Card Detect with SD_D3

chanhvinh.com

- Power-Down mode at Card removal
  SD Host Interface module is halted by the MSTP123 bit to MSTP120 bit in Standby Control Register 12 (STBCR12). If these bits of each channel in STBCR12 are set to 10, low power consumption is achieved at Card removal. See section 42, Power-Down Modes.

Bit	Bit Name	Initial Value	R/W	Description
3	MSTP123	1	R/W	SD Host Interface 0 Module Stop 00: SD Host Interface 0 Module runs. 01: Setting prohibited 10: Only card detect block in SD Host Interface 0 Module runs. 11: Clock supply to SD Host Interface 0 Module is halted.
2	MSTP122	1		
1	MSTP121	1	R/W	SD Host Interface 1 Module Stop 00: SD Host Interface 1 Module runs. 01: Setting prohibited 10: Only card detect block in SD Host Interface 1 Module runs. 11: Clock supply to SD Host Interface 1 Module is halted.
0	MSTP120	1		

(2) Write Protect

The SD host interface has two types of write protect functions.

- Write protect with SD_WP
  SD_WP is connected to the card socket, and pulled up or pulled down by the card insertion. The selection of pulling up or pulling down and the resistance value is decided by the specification of SD host device. As the SD_WP state is reflected to INFO7 in SD_INFO1, the write protect is decided after the SD card is inserted.
- Write protect with command
  The card's internal write protection and the card lock/unlock operation are realized by the command.

chanhvinh.com

38.3.3 Interrupt Request and DMA Transfer Request

(1) Interrupt Request

The SD host interface has three interrupt requests. Table 38.4 shows the relationship between the interrupt flag registers and the interrupt mask registers. When a bit in an interrupt mask register is set to 0, an interrupt occurs by setting the corresponding bit in the interrupt flag register to 1.
To clear a flag, write 0 to the bit to be cleared and 1 to the other bits.

Table 38.4 Interrupt Request

	Interrupt Flag Register		Interrupt Mask Register	
Interrupt Request	**Register Name**	**Bit Name**	**Register Name**	**Bit Name**
Card access interrupt (SDHI0)	SD_INFO1	INFO2	SD_INFO1_MASK	IMASK2
		INFO0		IMASK0
	SD_INFO2	ILA	SD_INFO2_MASK	IMASK
		BWE		BMASK1
		BRE		BMASK0
		ERR6		EMASK6
		ERR5		EMASK5
		ERR4		EMASK4
		ERR3		EMASK3
		ERR2		EMASK2
		ERR1		EMASK1
		ERR0		EMASK0
SDIO access interrupt (SDHI1)	SDIO_INFO1	EXWT	SDIO_INFO1_MASK	MEXWT
		EXPUB52		MEXPUB52
		IOIRQ		IOMSK
Card detect interrupt (SDHI3)	SD_INFO1	INFO9	SD_INFO1_MASK	IMASK9
		INFO8		IMASK8
		INFO4		IMASK4
		INFO3		IMASK3

chanhvinh.com

(2) DMA Transfer Request

The SD host interface has two types of DMA transfer requests. The DMA transfer requests are described in the following passages.

- SD_BUF write DMA transfer request
  - When the buffer is empty while the DMASDRW bit in CC_EXT_MODE is set to 1, the SD_BUF write DMA transfer request is asserted.
  - The SD_BUF write DMA transfer request is negated when the last data in one block (= the transfer data size set in SD_SIZE) is transferred. The SD_BUF write DMA transfer request is also negated by clearing the SDRST bit in SOFT_RST to 0 or setting the STP bit in SD_STOP to 1. Note that if a communications error or timeout occurs at the DMA transfer, the SD_BUF write DMA transfer request is not negated.
  - The number of DMA transfers should be n x one block. (n = integer, one block = the transfer data size set in SD_SIZE)
  - When the IOABT bit in SDIO_MODE is set to 1, the SD_BUF write DMA transfer request is negated.
  - The DMA transfer request is also negated by clearing the DMASDRW bit to 0. However, note that the DMA transfer request is asserted again when the DMASDRW bit is set to 1 before writing to SD_CMD.
- SD_BUF read DMA transfer request
  - When the buffer is full while the DMASDRW bit in CC_EXT_MODE is set to 1, the SD_BUF read DMA transfer request is asserted.
  - The SD_BUF read DMA transfer request is negated when the last data in one block (= the transfer data size set in SD_SIZE) is transferred. The SD_BUF read DMA transfer request is also negated by clearing the SDRST bit in SOFT_RST to 0 or setting the STP bit in SD_STOP to 1. Note that if a communications error or timeout occurs at the DMA transfer, the SD_BUF read DMA transfer request is not negated.
  - The number of DMA transfers should be n x one block. (n = integer, one block = the transfer data size set in SD_SIZE)
  - When the IOABT bit in SDIO_MODE is set to 1, the SD_BUF read DMA transfer request is negated.
  - The DMA transfer request is also negated by clearing the DMASDRW bit to 0. However, note that the DMA transfer request is asserted again when the DMASDRW bit is set to 1 before writing to SD_CMD.
  - The number of cycles at one DMA transfer will be more than P1φ × 10. If SD_CLK is high frequency and SD_BUF is not empty, SD_CLK is stopped until SD_BUF is empty.

chanhvinh.com

38.3.4 Communications Errors and Timeouts

- Communications Errors and Timeouts
  Table 38.5 and Table 38.6 show the relationships between the SD card interrupt flag register and SD error status register for communications errors and timeouts, respectively. When a bit in the SD card interrupt flag register is set to 1, the corresponding bit in the SD error status register is set to 1. The values of the SD error status register are cleared by writing to SD_CMD or writing 0 to the SDRST bit in SOFT_RST.

Table 38.5 Communications Errors

	SD Card Interrupt Flag Register		SD Error Status Register		
Communication Error	**Register Name**	**Bit Name**	**Register Name**	**Bit Name**	**Description**
END error	SD_INFO2	ERR2	SD_ERR_STS1	E5	When an error occurs in the CRC status length
				E4	When an error occurs in read data length
				E3	When an error occurs in the response length to a command issued within a command sequence
				E2	When an error occurs in the response length (other than a response to a command issued within a command sequence)
CRC error		ERR1		E11	When an error occurs in the CRC status
				E10	When a CRC error occurs in the read data
				E9	When a CRC error occurs in the response to a command issued within a command sequence
				E8	When a CRC error occurs in the response (other than a response to a command issued within a command sequence)
CMD error		ERR0		E1	The command index of the transmitted command differed from the command index of the received response (for a command issued within a command sequence).
				E0	The command index of the transmitted command differed from the command index of the received response (for a command issued other than within a command sequence)

chanhvinh.com

Table 38.6 Timeouts

Timeout	SD Card Interrupt Flag Register		SD Error Status Register		Description
	Register Name	Bit Name	Register Name	Bit Name	
Response timeout	SD_INFO2	ERR6	SD_ERR_STS2	E1	When the response to a command issued within a command sequence is not received though a longer time than 640 cycles of SD_CLK has elapsed
				E0	When the response (other than a response to a command issued within a command sequence) is not received though a longer time than 640 cycles of SD_CLK has elapsed
Data timeout (other than response timeout)		ERR3		E6	When the busy state (SD_D0 = 0) continues for longer than Ncycle after the CRC status
				E5	When the CRC status is not received though a longer time than Ncycle has elapsed after data writing
				E4	When read data is not received though a longer time than Ncycle has elapsed after read command
					When read data for the next block are not received though a longer time than Ncycle has elapsed after the reception of read data
					When read data for the next block are not received though a longer time than Ncycle has elapsed after release of the read wait state
				E3	When the busy state (SD_D0 = 0) for longer than Ncycle continues after CMD12 has been issued within a command sequence.
				E2	When the busy state (SD_D0 = 0) for longer than Ncycle continues after R1b response

chanhvinh.com

38.4 Usage Example

38.4.1 Command without Data Transfer

(1) Flowchart

Figure 38.8 and Figure 38.9 show flowchart examples.

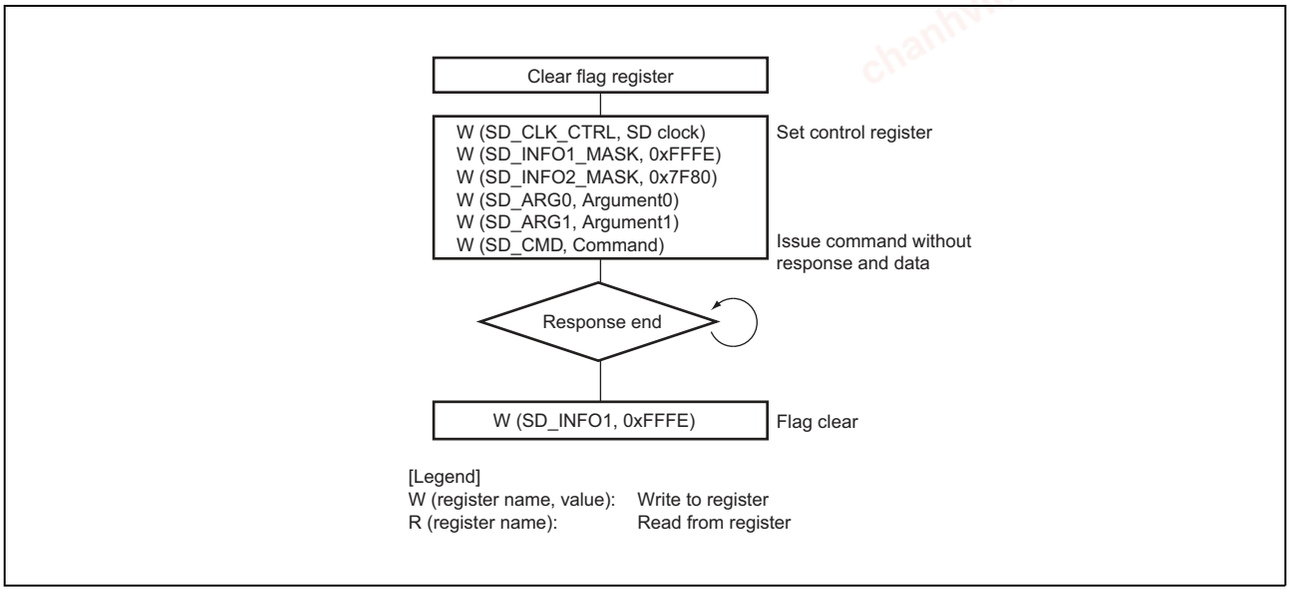


Figure 38.8 Flow Example of Command Without Response and Data

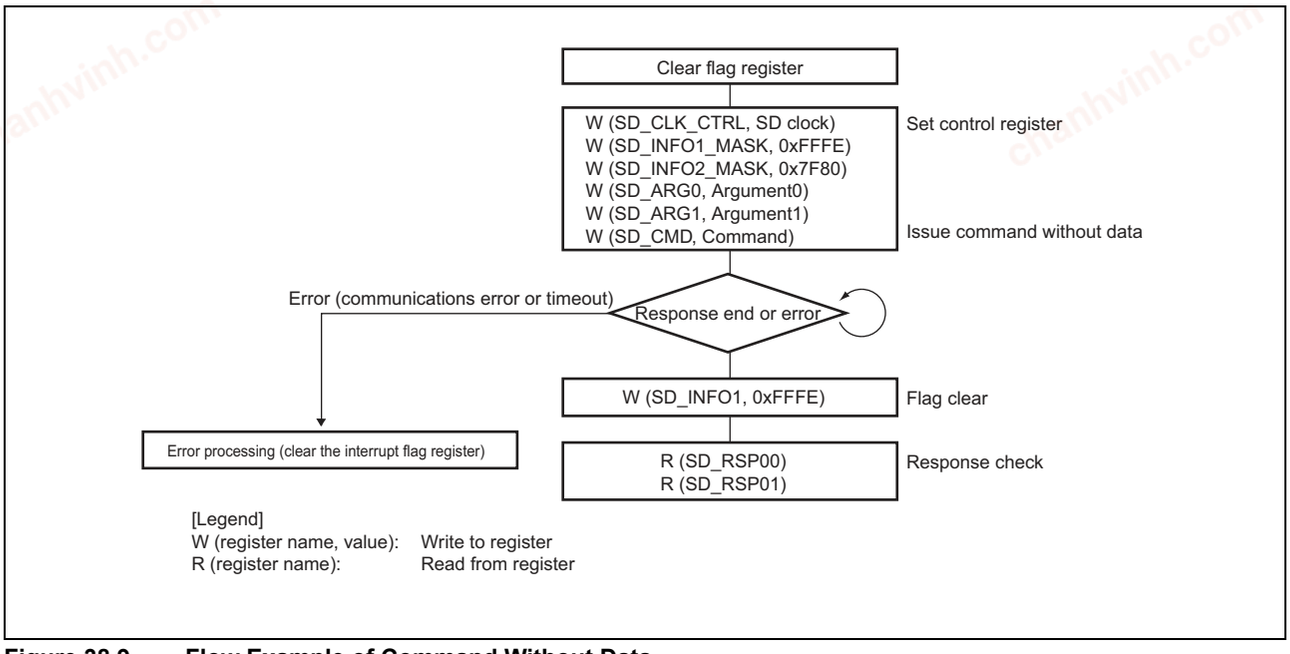


Figure 38.9 Flow Example of Command Without Data

chanhvinh.com

(2) Operation for Command without Data Transfer: The operation of the command without data transfer is described in the following.

- Command without response and data
  - Flag register clear
    First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)
  - Control register set
    Set the SD clock (SD_CLK), interrupt mask, and so on. (SD_CLK_CTRL, SD_INFO1_MASK, and SD_INFO2_MASK)
  - Command issue
    Set CMD Argument in SD_ARG0 and SD_ARG1 and write to SD_CMD.
    Accordingly, CMD is issued, and the operation is started.
  - Flag clear
    When transmission of a command is completed, INFO0 (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO0 to 0.

- Command without data
  - Flag register clear
    First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)
  - Control register set
    Set the SD clock (SD_CLK), interrupt mask, and so on. (SD_CLK_CTRL, SD_INFO1_MASK, and SD_INFO2_MASK)
  - Command issue
    Set CMD Argument in SD_ARG0 and SD_ARG1 and write to the SD_CMD.
    Accordingly, CMD is issued, and the operation is started.
  - Flag clear
    When a response is received, INFO0 (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO0 to 0.
  - Read a response from SD_RSP00 and SD_RSP01.

Furthermore, perform error processing (clear the interrupt flag register) if an error occurs (a communications error or timeout).

chanhvinh.com

38.4.2 Single Block Read

(1) Flowchart

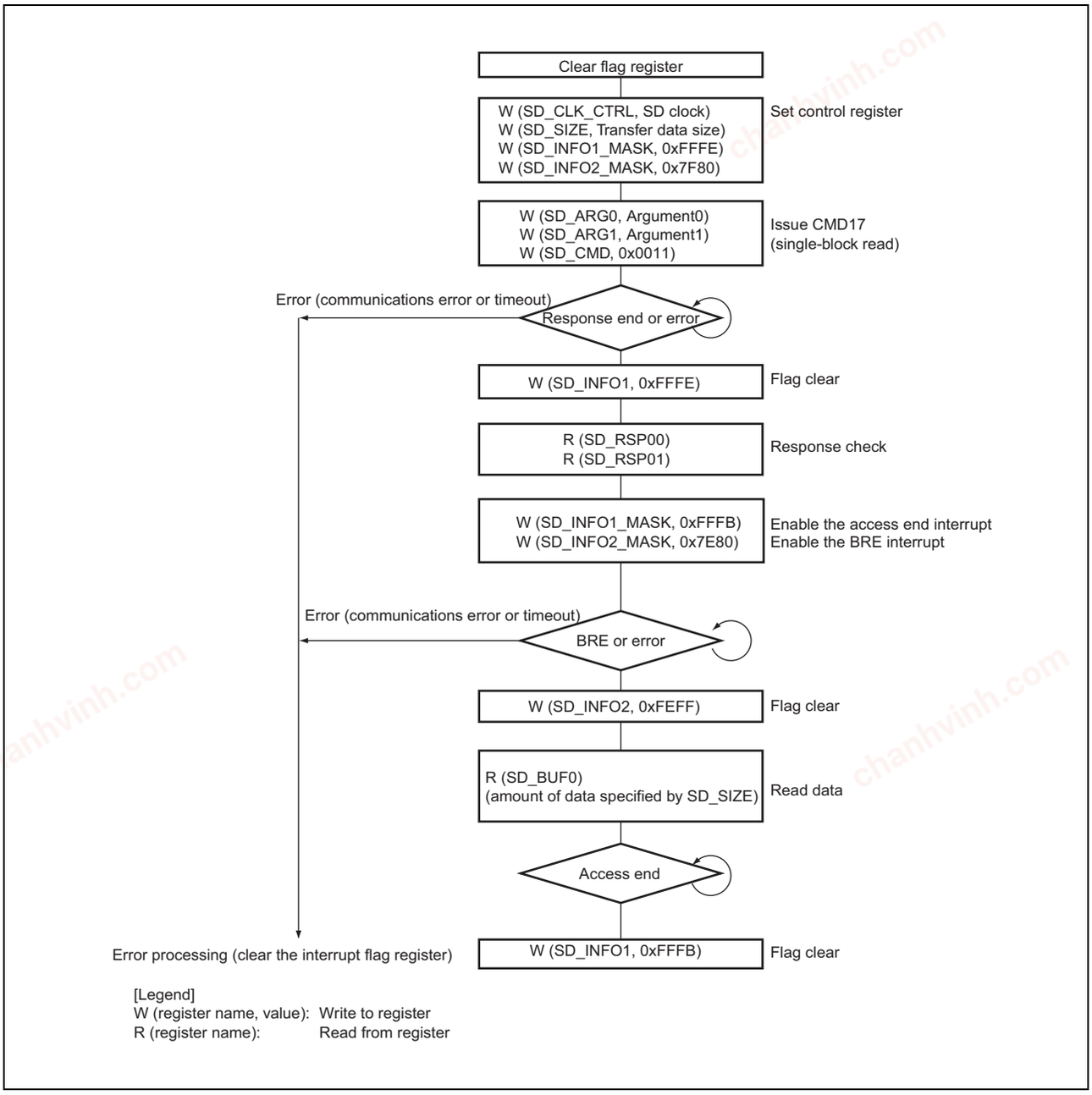


Figure 38.10 Single Block Read Flowchart Example

chanhvinh.com

(2) Operation for Single Block Read: The operation of the single block read is described in the following.

– Flag register clear
First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)

– Control register set
Set the SD clock (SD_CLK), transfer data size, interrupt mask, and so on. (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK)

– Command issue (CMD17)
Set CMD17 Argument in SD_ARG0 and SD_ARG1 and write 0x0011 to SD_CMD.
Accordingly, CMD17 is issued, and the single block read operation is started.

– Response check
On receiving the response, INFO0 (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO0 to 0 and read the response from SD_RSP00 and SD_RSP01.
If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP or the IOABT bit in SDIO_MODE to 1. Furthermore, this causes CMD12 and CMD52 to not be issued.
If the INFO2 bit (access end) in SD_INFO has been set, halting the command sequence will also lead to the generation of an interrupt.

– Data receive from SD card and data read
Write 0xFFFB to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0x7E80 to SD_INFO2_MASK to enable the BRE interrupt. When the data receive from the SD card is completed, the BRE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BRE bit to 0 and read the amount of data specified by SD_SIZE from SD_BUF0.
However, a communications error or timeout may be generated if data are being received while reading of SD_BUF0 is in progress.

– Operation complete
When the data read from SD_BUF0 is completed, INFO2 (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO2 to 0 to end the single block read operation.

Furthermore, perform error processing (clear the interrupt flag register) if an error occurs (a communications error or timeout).

chanhvinh.com

38.4.3 Single Block Write

(1) Flowchart

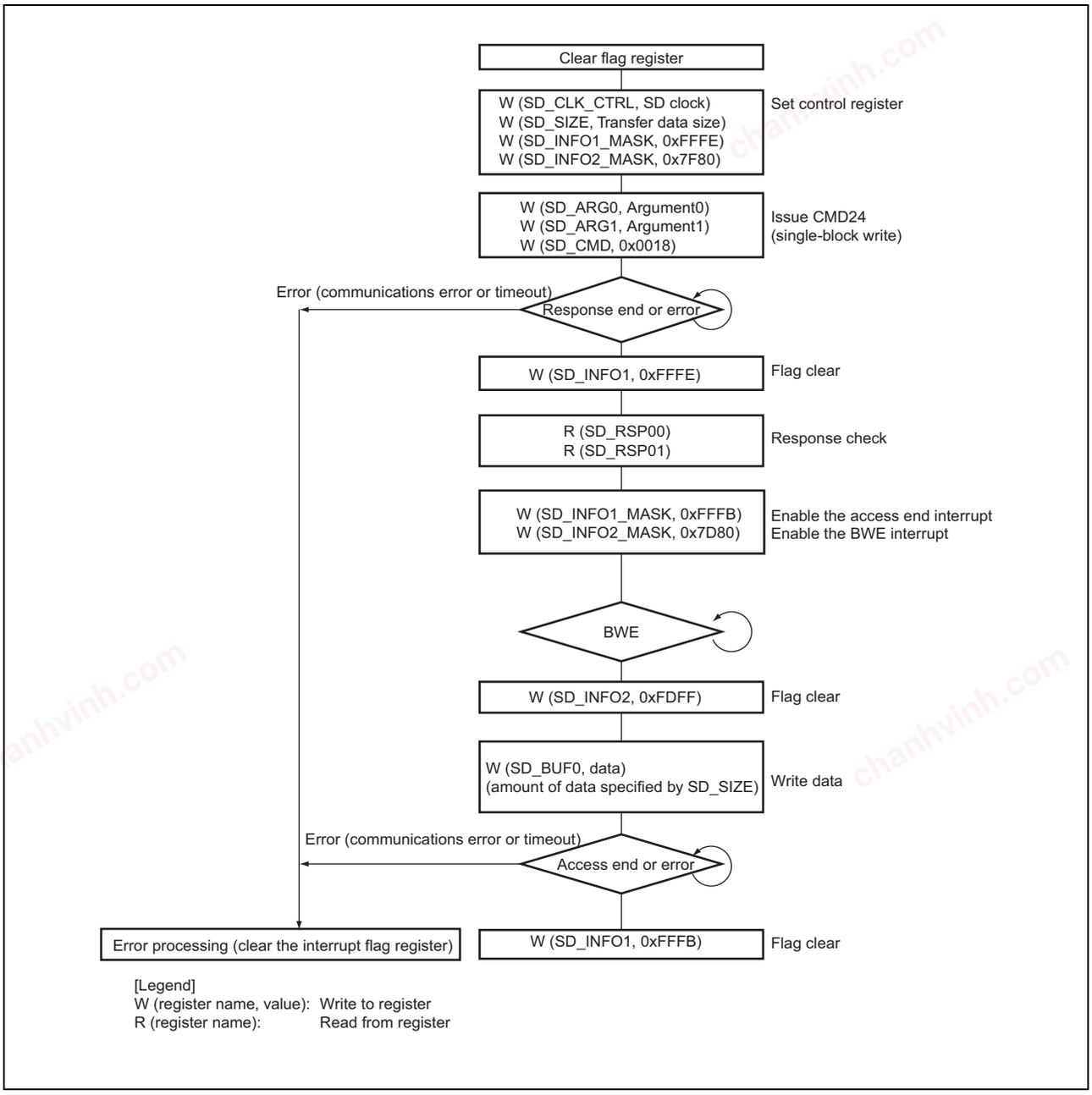


Figure 38.11 Single Block Write Flowchart Example

chanhvinh.com

(2) Operation for Single Block Write: The operation of the single block write is described in the following.

– Flag register clear
First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)

– Control register set
Set the SD clock (SD_CLK), transfer data size, interrupt mask, and so on. (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK)

– Command issue (CMD24)
Set CMD24 Argument in SD_ARG0 and SD_ARG1 and write 0x0018 to SD_CMD.
Accordingly, CMD24 is issued, and the single block write operation is started.

– Response check
On receiving the response, INFO0 (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO0 to 0 and read the response from SD_RSP00 and SD_RSP01.
If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP or the IOABT bit in SDIO_MODE to 1. Furthermore, this causes CMD12 and CMD52 to not be issued.
If the INFO2 bit (access end) in SD_INFO has been set, halting the command sequence will also lead to the generation of an interrupt.

– Data write and data transmit to SD card
Write 0xFFFB to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0x7D80 to SD_INFO2_MASK to enable the BWE interrupt. When SD_BUF0 is ready for the data to be written, the BWE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BWE bit to 0 and write the amount of data specified by SD_SIZE to SD_BUF0. When the data write to SD_BUF0 is completed, data is transmitted to the SD card. Then, the CRC status and busy state are received from the SD card.
However, a communications error or timeout may be generated if data are being transmitted after writing to SD_BUF0.

– Operation complete
When the CRC status and busy state are received from the SD card, INFO2 (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the INFO2 bit to 0 to end the single block write operation.

Furthermore, perform error processing (clear the interrupt flag register) if an error occurs (a communications error or timeout).

chanhvinh.com

38.4.4 Multiple Block Read

(1) Flowchart

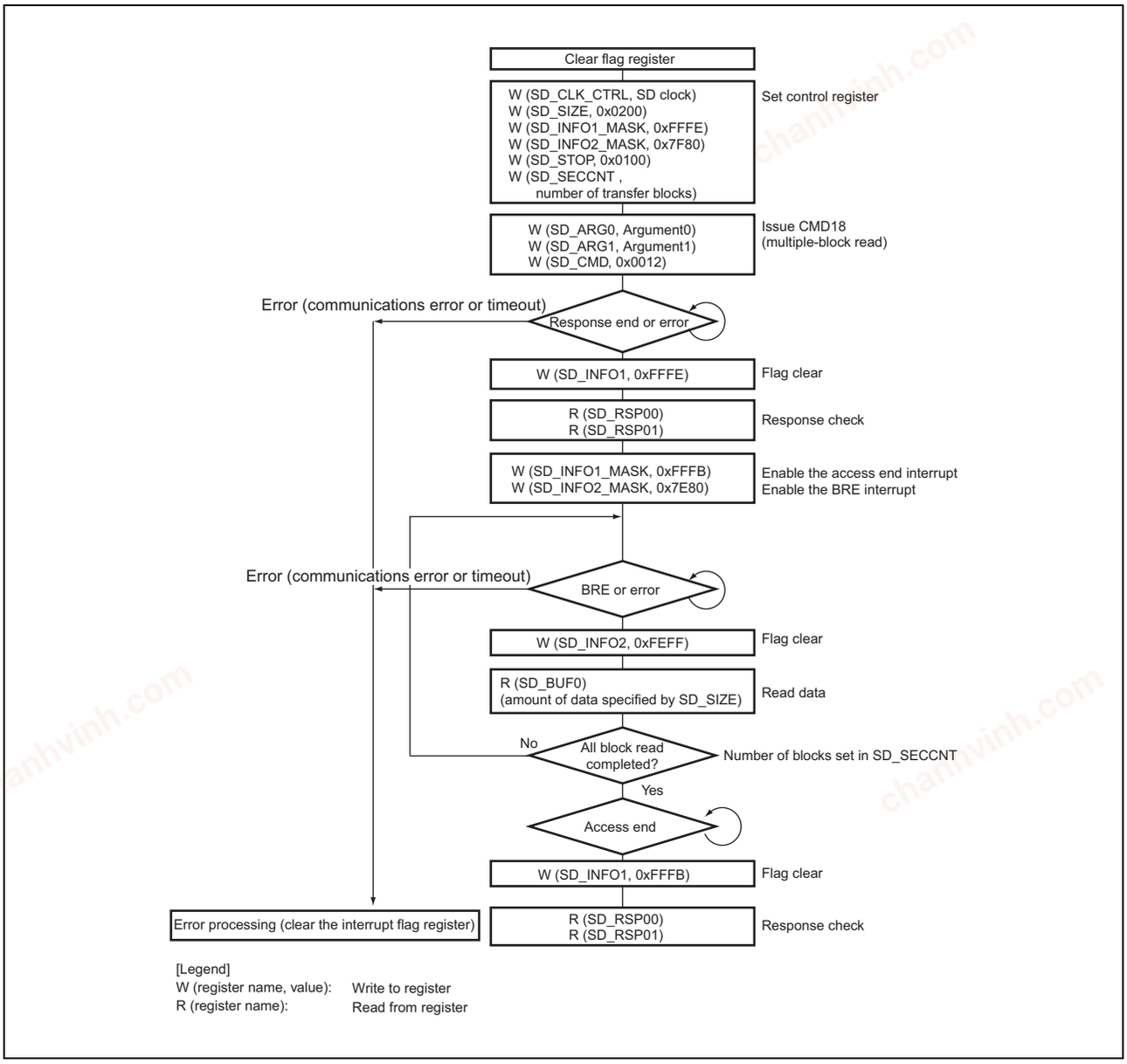


Figure 38.12 Multiple Block Read Flowchart Example

chanhvinh.com

(2) Operation for Multiple Block Read: The operation of the multiple block read is described in the following.

– Flag register clear
First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)

– Control register set
Set the SD clock (SD_CLK), transfer data size, interrupt mask, and so on. (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK)
Set SEC in SD_STOP to 1, and set the number of transfer blocks in SD_SECCNT.

– Command issue (CMD18)
Set CMD18 Argument in SD_ARG0 and SD_ARG1 and write 0x0012 to SD_CMD.
Accordingly, CMD18 is issued, and the multiple block read operation is started.

– Response check
On receiving the response, INFO0 (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO0 to 0 and read the response from SD_RSP00 and SD_RSP01.
If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP to 1. Setting the STP bit to 1 also causes CMD12 to be issued and the response received. When this happens, the arguments for the CMD12 command are set in SD_ARG0 and SD_ARG1. If the command sequence is halted because the access end interrupt has been enabled, an interrupt will be generated by setting of the INFO2 bit (access end) bit in SD_INFO1 to 1 when reception of the response has been completed.
Clear the INFO2 bit to 0 and read the response.

– Data receive from SD card and data read
Write 0xFFFB to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0x7E80 to SD_INFO2_MASK to enable the BRE interrupt. When one-block data receive from the SD card is completed, the BRE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BRE bit to 0 and read the amount of data specified by SD_SIZE from SD_BUF0. Doing this repeats transfer of the number of blocks set in SD_SECCNT. However, a communications error or timeout may be generated if data are being received while reading of SD_BUF0 is in progress. When reception of the last block of data is completed, CMD12 is automatically issued and the response is received.
At this point, the arguments of CMD12 are reflected as the arguments of CMD18.

– Operation complete
When all-block data read and the CMD12 response receive are completed, INFO2 (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO2 to 0 to read the response. This is the end of multiple block read operation.

Furthermore, perform error processing (clear the interrupt flag register) if an error occurs (a communications error or timeout).

chanhvinh.com

38.4.5 Multiple Block Write

(1) Flowchart

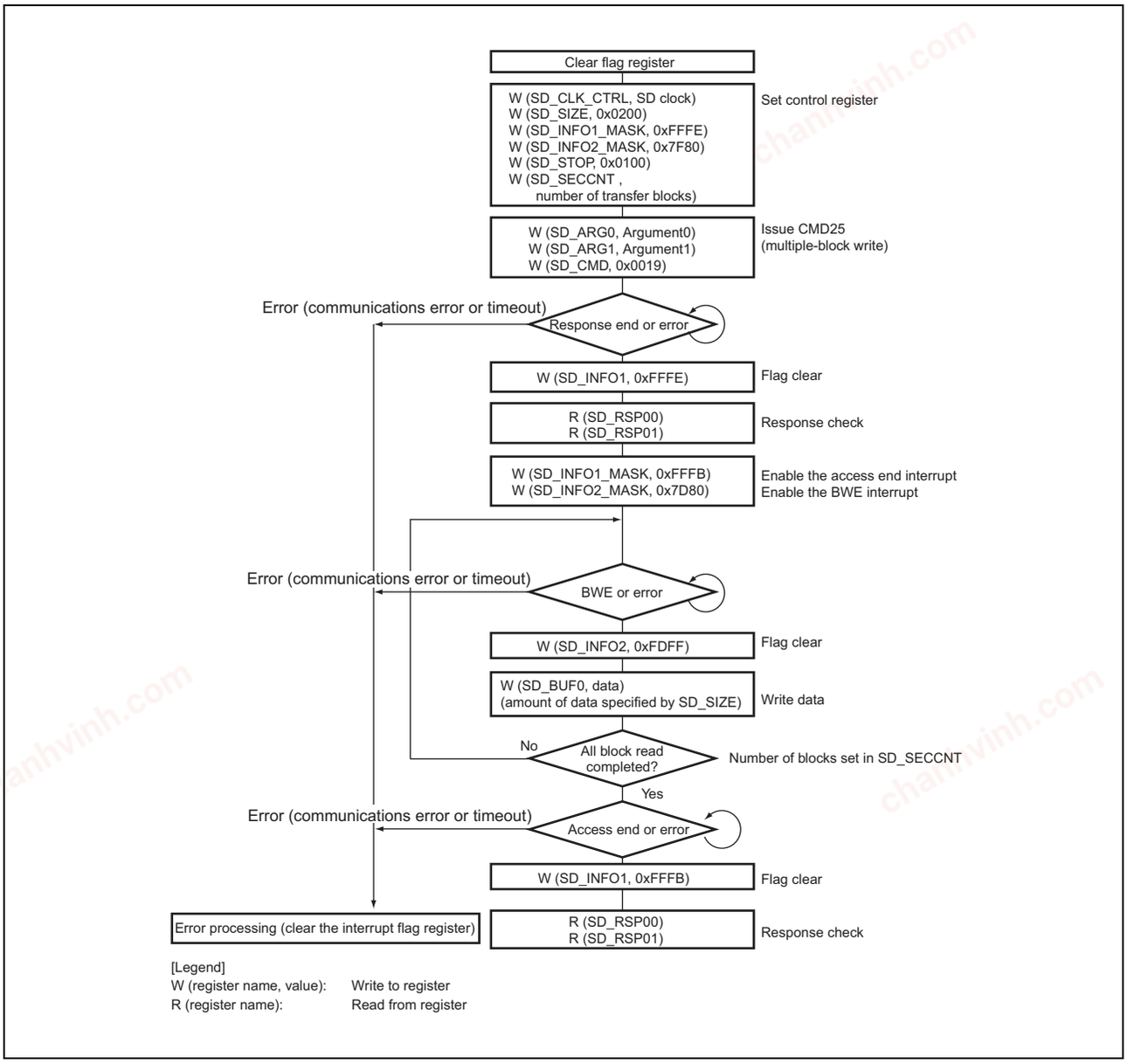


Figure 38.13 Multiple Block Write Flowchart Example

chanhvinh.com

(2) Operation for Multiple Block Write: The operation of the multiple block write is described in the following.

– Flag register clear
First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)

– Control register set
Set the SD clock (SD_CLK), transfer data size, interrupt mask, and so on. (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK)
Set the SEC bit in SD_STOP to 1, and set the number of transfer blocks in SD_SECCNT.

– Command issue (CMD25)
Set CMD25 Argument in SD_ARG0 and SD_ARG1 and write 0x0019 to SD_CMD.
Accordingly, CMD25 is issued, and the multiple block write operation is started.

– Response check
On receiving the response, the INFO0 bit (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the INFO0 bit to 0 and read the response from SD_RSP00 and SD_RSP01.
If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP to 1. Setting the STP bit to 1 also causes CMD12 to be issued and the response received. When this happens, the arguments for the CMD12 command are set in SD_ARG0 and SD_ARG1. If the command sequence is halted because the access end interrupt has been enabled, an interrupt will be generated by setting of the INFO2 bit (access end) bit in SD_INFO1 to 1 when reception of the response has been completed.
Clear the INFO2 bit to 0 and read the response.

– Data write and data transmit to SD card
Write 0xFFFB to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0x7D80 to SD_INFO2_MASK to enable the BWE interrupt. When SD_BUF0 is ready for the data to be written, the BWE bit in the SD_INFO2 resister is set to 1 to generate an interrupt. Clear the BWE bit to 0 and write the amount of data specified by SD_SIZE to SD_BUF0. When the data write to SD_BUF0 is completed, data is transmitted to the SD card. Then, the CRC status and busy state are received from the SD card. Doing this repeats transfer of the number of blocks set in SD_SECCNT. However, a communications error or timeout may be generated if data are being received while writing to SD_BUF0 is in progress. When reception of the last block of data is completed, CMD12 is automatically issued and the response is received.
At this point, the arguments of CMD12 are reflected as the arguments of CMD25.

– Operation complete
When all-block data transmit and the CRC status receive are completed, the INFO2 bit (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the INFO2 bit to 0 to read the response. This is the end of multiple block write operation.

Furthermore, perform error processing (clear the interrupt flag register) if an error occurs (a communications error or timeout).

chanhvinh.com

38.4.6 IO_RW_DIRECT Command (CMD52)

(1) Flowchart

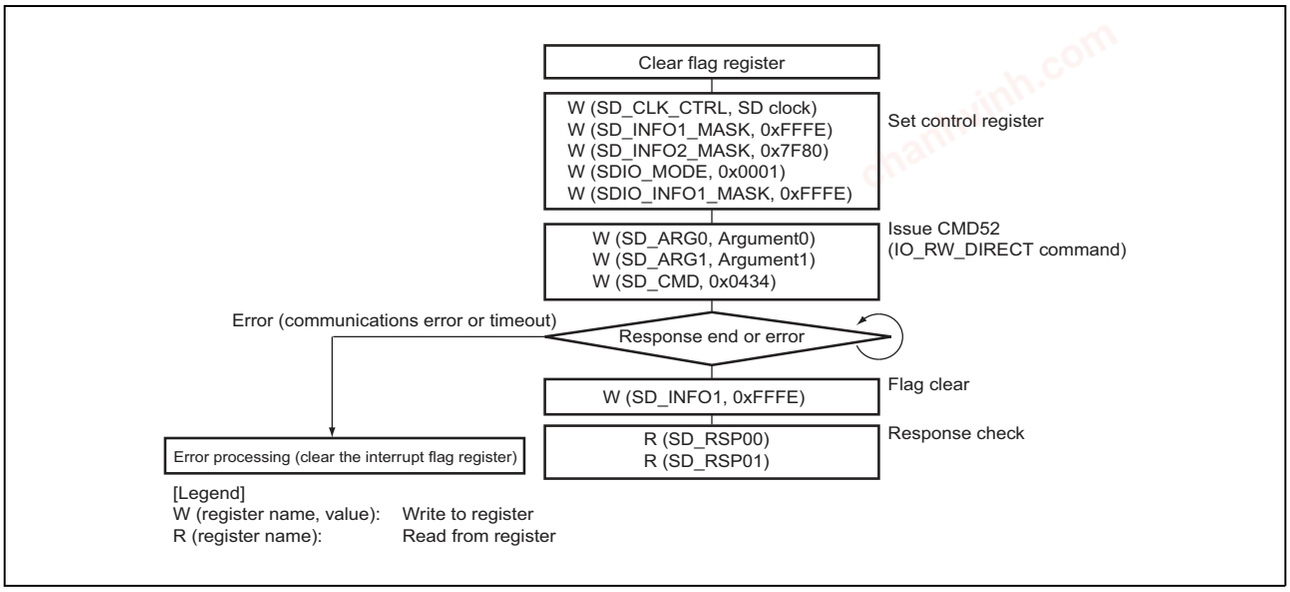


Figure 38.14 IO_RW_DIRECT Command (CMD52) Flowchart Example

chanhvinh.com

38.4.7 IO_RW_EXTENDED Command (CMD53/Multiple Block Read)

(1) Flowchart

Figure 38.15 shows a flowchart example for CMD53 (multiple block read).

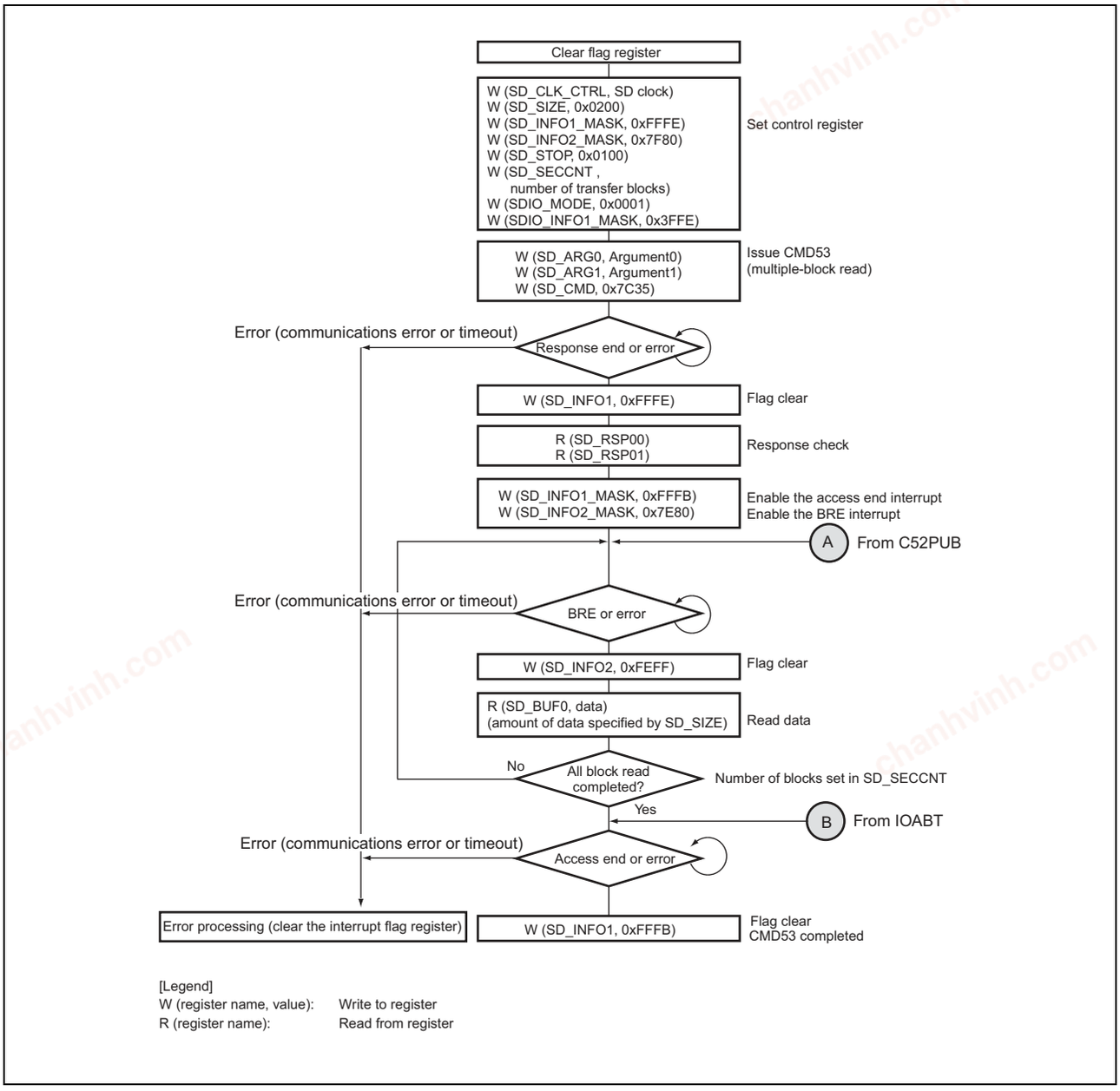


Figure 38.15 CMD53 (Multiple Block Read) Flowchart Example

chanhvinh.com

Figure 38.16 shows a flowchart example when CMD52 (SDIO abort) is issued at CMD53 (multiple block read).

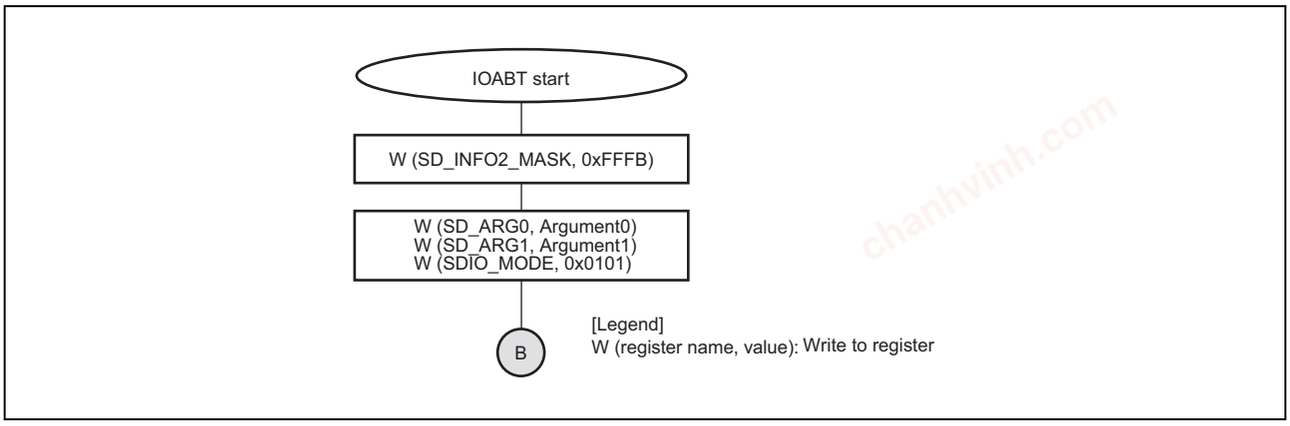


Figure 38.16 Flowchart Example when CMD52 (SDIO Abort) is Issued at CMD53 (Multiple Block Read)

Figure 38.17 shows a flowchart example when CMD52 (SDIO none abort) is issued at CMD53 (multiple block read) while the SD host interface is in the read wait state.

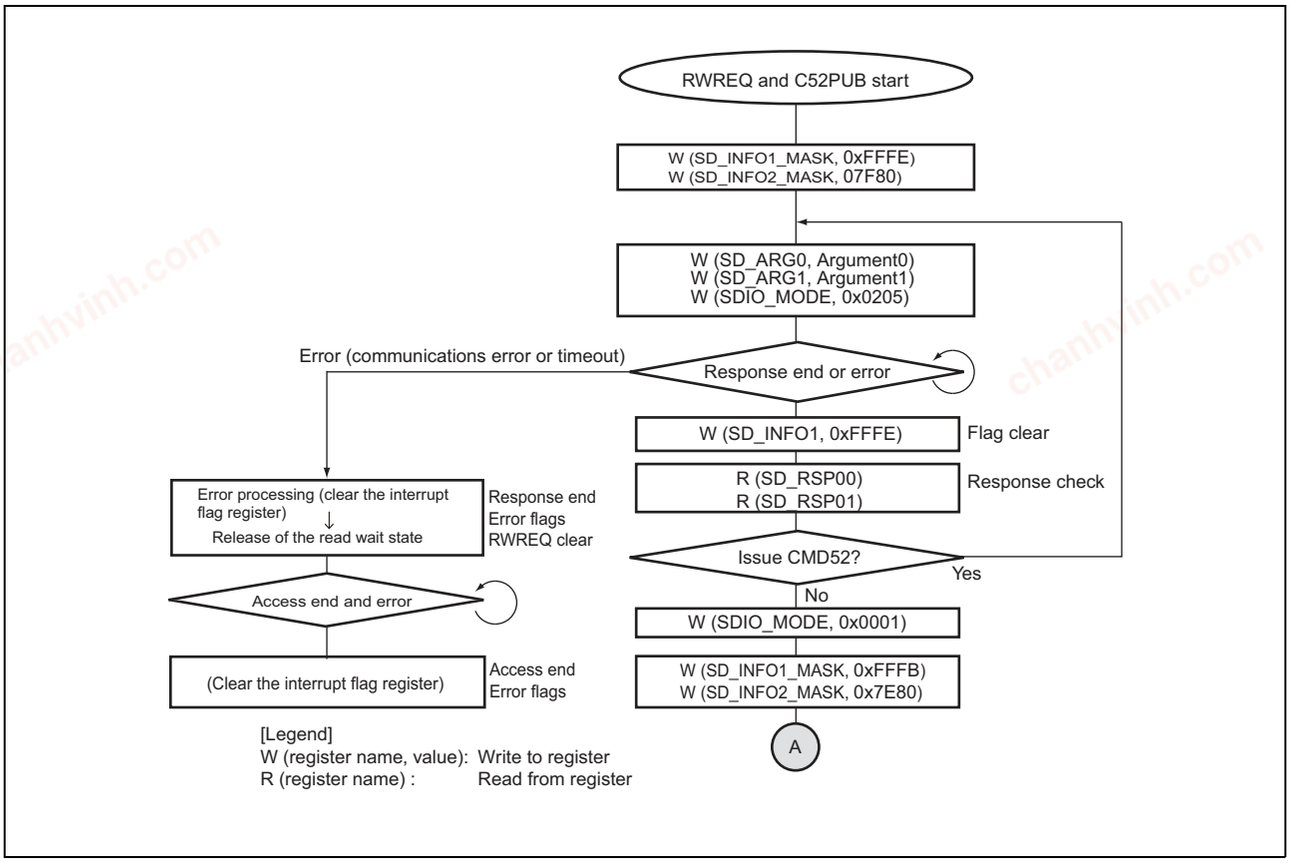


Figure 38.17 Flowchart Example when CMD52 (SDIO None Abort) is Issued at CMD53 (Multiple Block Read) while the SD Host Interface is in the Read Wait State

chanhvinh.com

38.4.8 IO_RW_EXTENDED Command (CMD53/Multiple Block Write)

(1) Flowchart

Figure 38.18 shows a flowchart example for CMD53 (multiple block write).

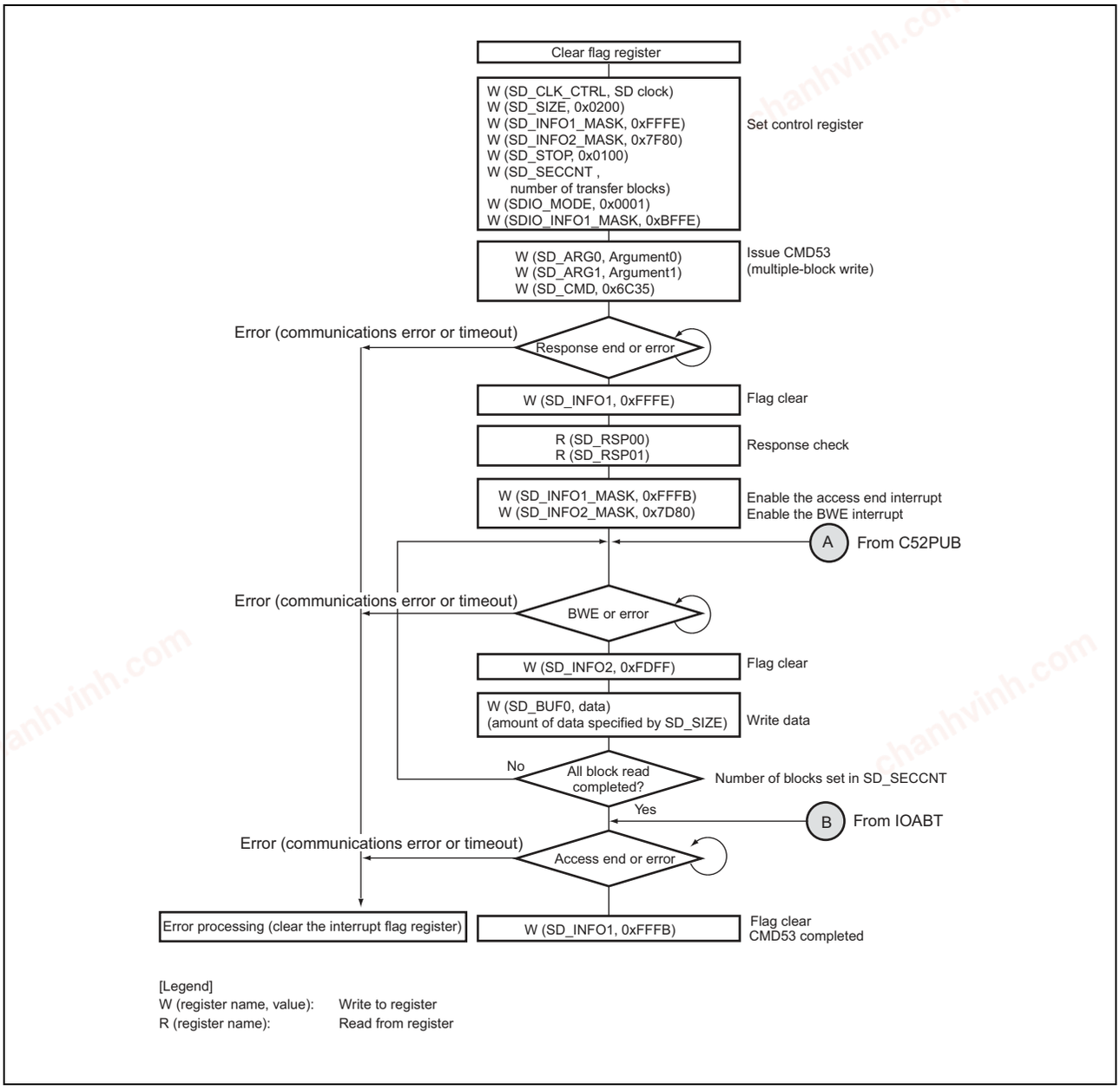


Figure 38.18 CMD53 (Multiple Block Write) Flowchart Example

chanhvinh.com

Figure 38.19 shows a flowchart example when CMD52 (SDIO abort) is issued at CMD53 (multiple block write).

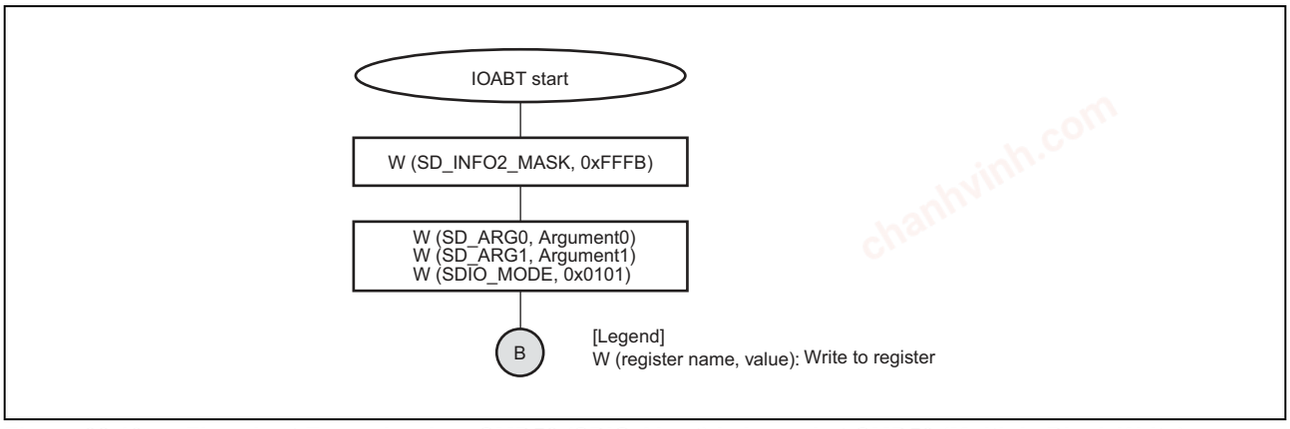


Figure 38.19 Flowchart Example when CMD52 (SDIO Abort) is Issued at CMD53 (Multiple Block Write)

Figure 38.20 shows a flowchart example when CMD52 (SDIO none abort) is issued at CMD53 (multiple block write).

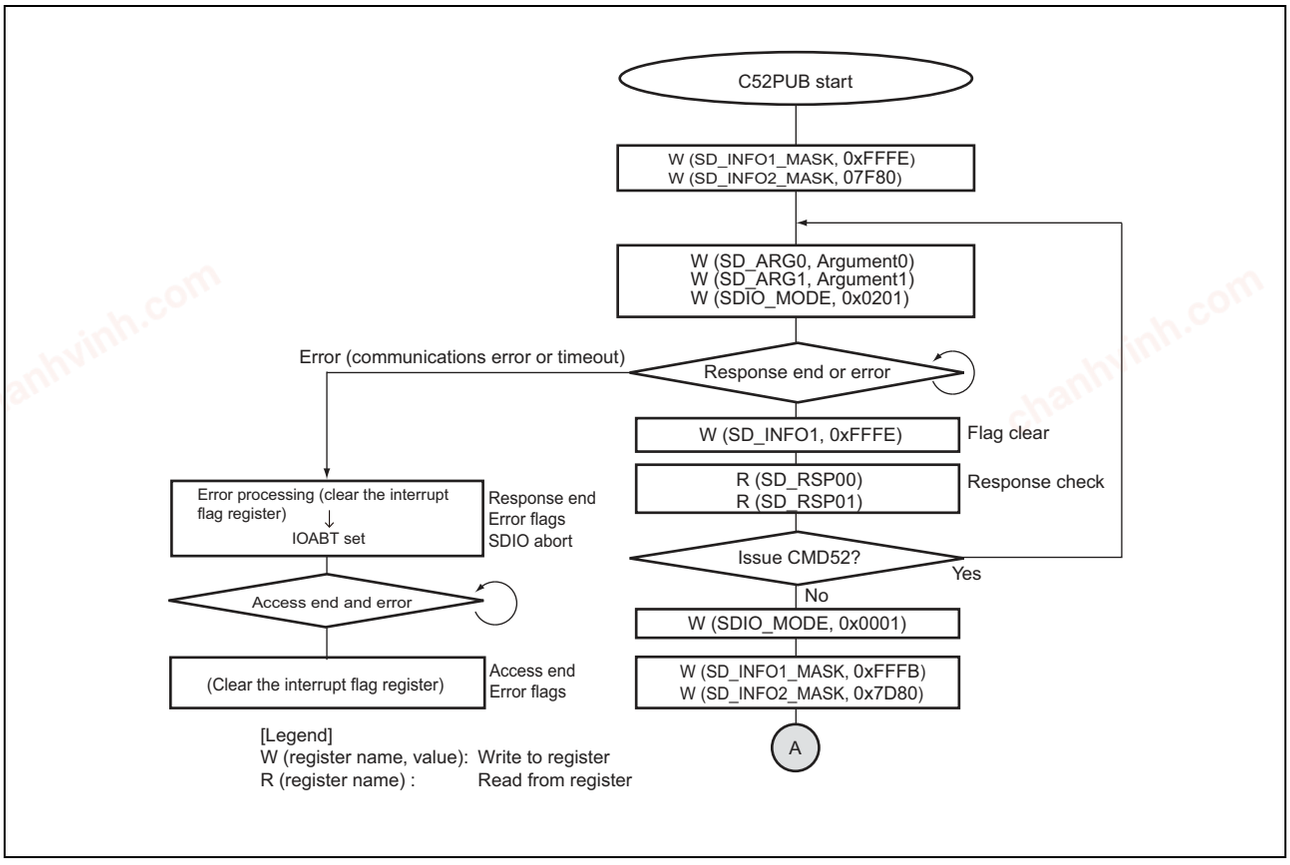


Figure 38.20 Flowchart Example when CMD52 (SDIO None Abort) is Issued at CMD53 (Multiple Block Write)

chanhvinh.com

38.4.9 DMA Transfer

(1) SD_BUF DMA Transfer

Figure 38.21 shows a flowchart example for SD_BUF DMA read when CMD18 (multiple block read) is issued.

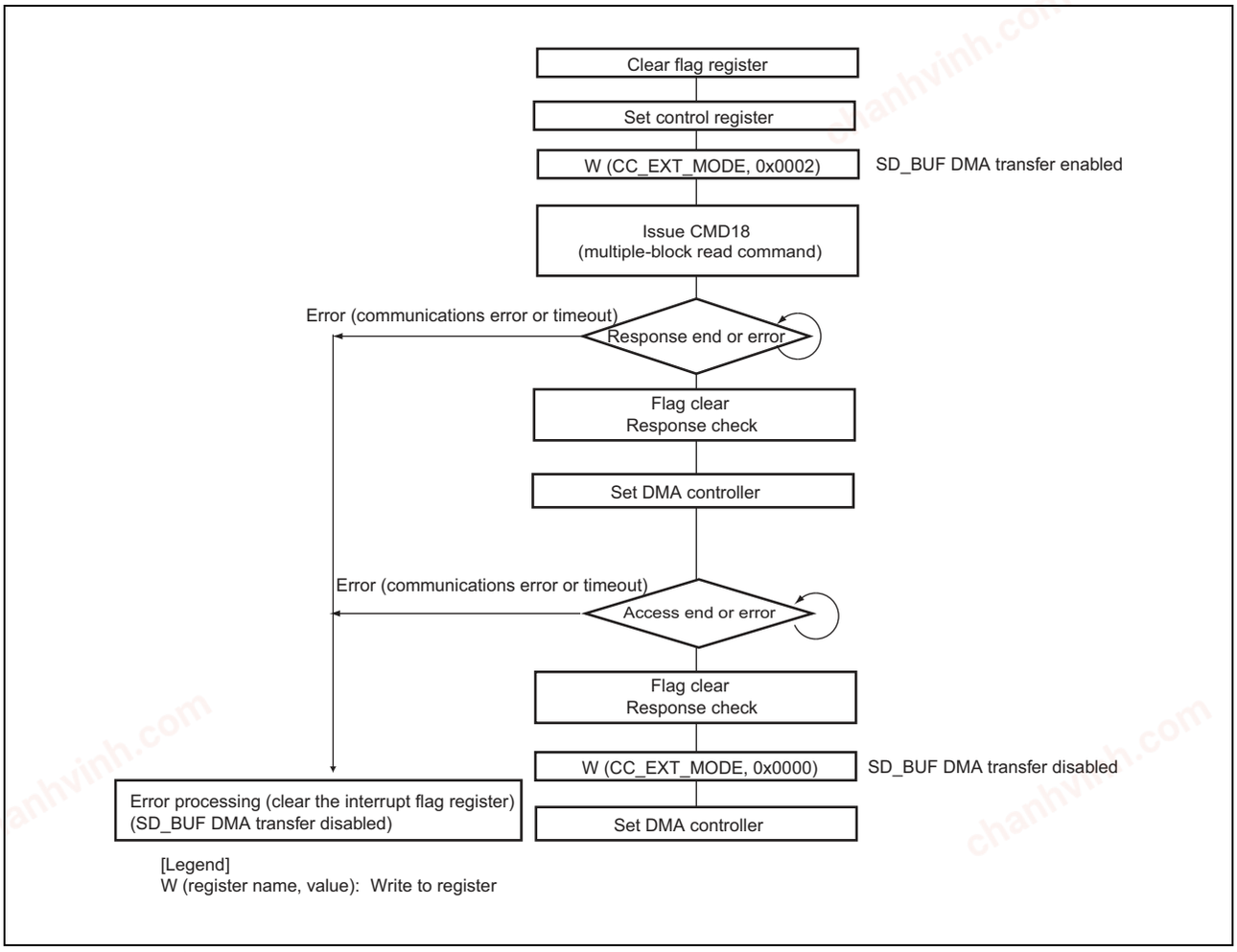


Figure 38.21 SD_BUF DMA Read Flowchart Example

chanhvinh.com

Figure 38.22 shows a flowchart example for SD_BUF DMA write when CMD25 (multiple block write) is issued.

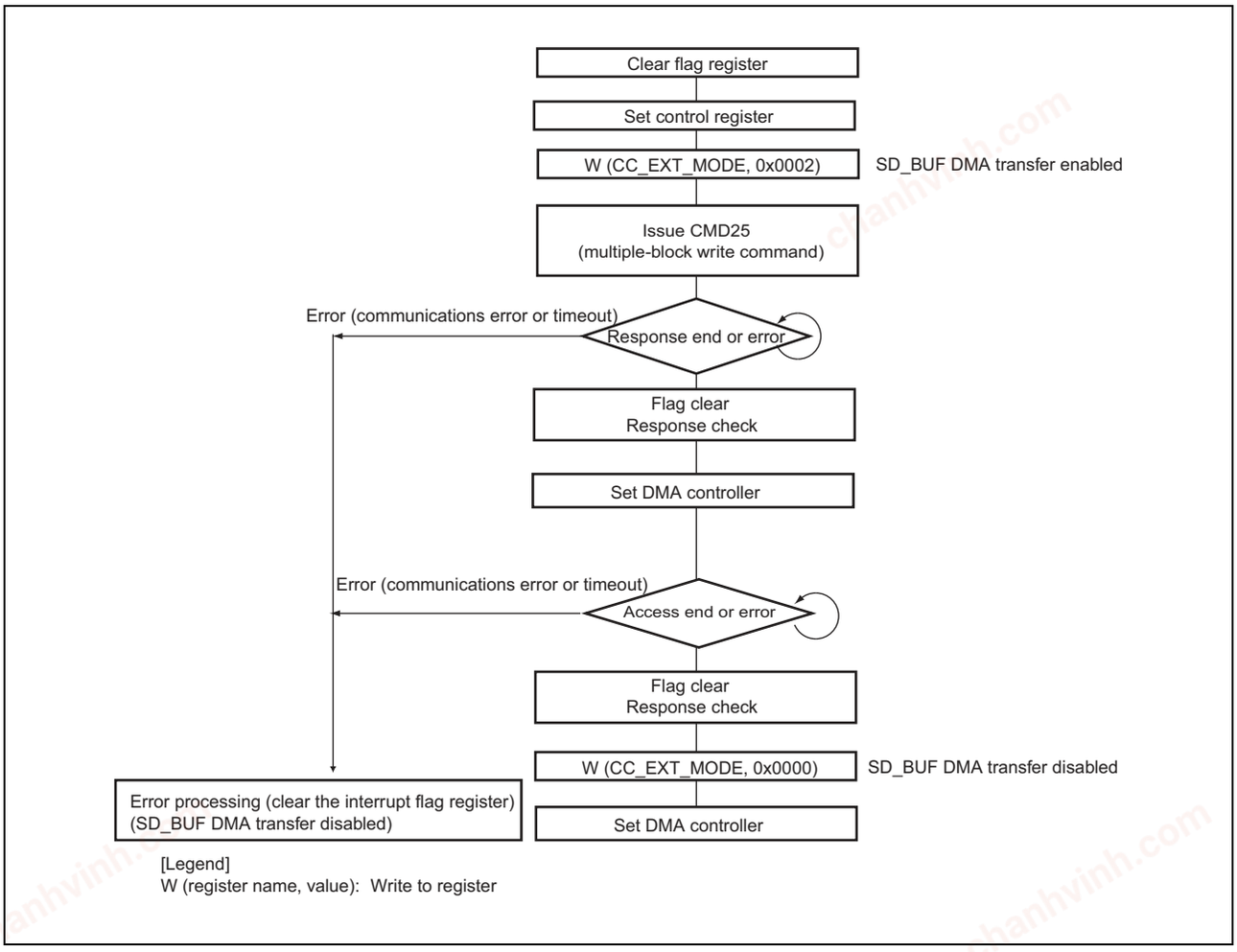


Figure 38.22 SD_BUF DMA Write Flowchart Example

chanhvinh.com

38.4.10 Example of SD_CMD Register Setting

Table 38.7 lists the example of SD_CMD register setting.

Table 38.7 Example of SD_CMD Register Setting

Type	Command	Example of SD_CMD Register Setting	Remark
CMD	CMD0	0x0000	
	CMD2	0x0002	
	CMD3	0x0003	
	CMD4	0x0004	
	CMD5	0x0705	
	CMD6	0x1C06	
	CMD7	0x0007	When the card is placed in the deselected state, the response timeout flag will be set since there is no response.
	CMD8	0x0408	
	CMD9	0x0009	
	CMD10	0x000A	
	CMD12	0x000C	
	CMD13	0x000D	
	CMD15	0x000F	
	CMD16	0x0010	
	CMD17	0x0011	
	CMD18	0x0012	
	CMD24	0x0018	
	CMD25	0x0019	
	CMD27	0x001B	
	CMD28	0x001C	
	CMD29	0x001D	
	CMD30	0x001E	
	CMD32	0x0020	
	CMD33	0x0021	
	CMD38	0x0026	
	CMD42	0x002A	
	CMD52	0x0434	
	CMD53	0x1C35	Single read
		0x0C35	Single write
		0x7C35	Multiple read
		0x6C35	Multiple write
	CMD55	0x0037	
	CMD56	0x0038	
ACMD	ACMD6	0x0046	
	ACMD13	0x004D	
	ACMD22	0x0056	
	ACMD23	0x0057	
	ACMD41	0x0069	
	ACMD42	0x006A	
	ACMD51	0x0073	

chanhvinh.com

38.5 Usage Note

(1) SD_BUF Illegal Write Access

When writing data to SD_BUF0 after the single block write or multi block write command is issued, the data of the size specified by SD_SIZE must be written to.
If the data of the size which exceeds the size specified by SD_SIZE is written to, the ERR4 bit in SD_INFO2 is set to 1. In addition, the data written to SD_BUF0 may not be transmitted and it causes the SCLKDIVEN bit in SD_INFO2 to hold the value of 0. In such cases, clearing the SDRST bit in SOFT_RST to 0 and then restoring its value to 1 clears the SCLKDIVEN bit to 1.
However, for the single byte or three bytes when the number of bytes setting in SD_SIZE is odd, or the fraction of bytes when the number of bytes setting in SD_SIZE is even, since the portion of dummy data writing is regarded as excess data and ignored, it is not within the scope of the above description (the fraction of bytes: the two bytes that are not in a four-byte unit).

(2) Block Number Limitation for Multiple Block Read

When performing a multiple block read of one or two blocks, depending on the timing with which the response register is read, the response value may not be read properly. When receiving one or two blocks of data, use single block read operation.
Figure 38.23 shows the processing flows of SD host interface (hardware) operation and software operation when a multiple block read is performed on two blocks. As shown in the incorrect operation of Figure 38.23, when an interrupt is generated on reception of the CMD18 response and the timing with which the response register is read by the interrupt is delayed, the data during the CMD12 response reception or the CMD12 response may be read. In the case of a multiple block read of three or more blocks, CMD12 is not issued until the block of data has been read, so this problem does not arise. Furthermore, in the case of a multiple block write, since the CMD25 response is read before the block of data is sent, the problem does not arise.

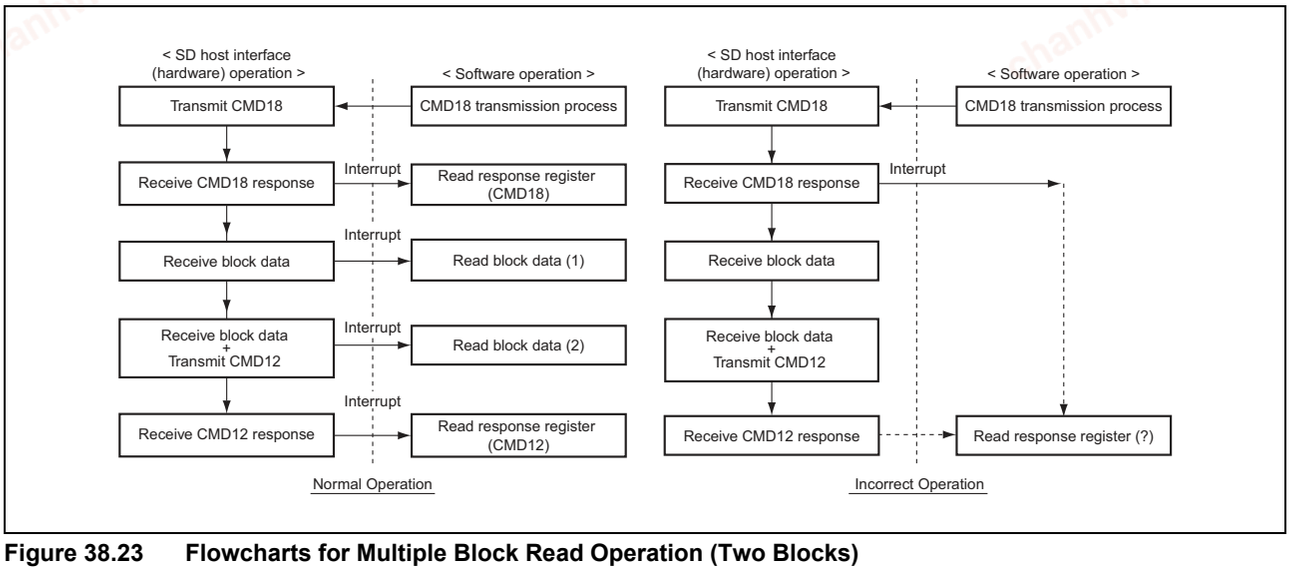


Figure 38.23 Flowcharts for Multiple Block Read Operation (Two Blocks)

(3) Automatic Control of SD_CLK Output

In the SD Card standard, 74 cycles of SD_CLK must be output before initialization of the card. For this reason, use automatic control of SD_CLK output after 74 cycles of SD_CLK have been output. Furthermore, if automatic control of SD_CLK output was in use, SD_CLK output is stopped on completion of the sequence for a communications error or timeout. Thus, in cases where state transitions within the SD card are necessary and so on after completion of the sequence, release automatic control of SD_CLK output and restart supply of SD_CLK to the SD card.

chanhvinh.com

(4) Control of the C52PUB Setting for Multiple Block Write

If the C52PUB bit in SDIO_MODE is set to 1 during a sequence of multiple block write due to CMD53, CMD52 is not issued until SD_BUF becomes empty. For this reason, set the C52PUB bit after suspending writing to SD_BUF by following the appropriate procedure below.

- When DMA transfer is not in use

1. Before setting the C52PUB bit, suspend writing to SD_BUF by making the setting in SD_INFO2 to disable BWE interrupts.
2. Set the C52PUB bit in SDIO_MODE to 1 (so that CMD52 is issued when SD_BUF becomes empty).
3. After the INFO0 interrupt processing in SD_INFO1 due to the issuing of CMD52 has been completed, restart writing to SD_BUF by making the setting in SD_INFO2 to enable BWE interrupts.

- When DMA transfer is in use

1. Every time DMA transfer of the value set in SD_SIZE × n blocks (where n = 1, 2, ...) proceeds, suspend writing to SD_BUF by DMA transfer before the C52PUB bit is set.
2. Set the C52PUB bit in SDIO_MODE to 1 (so that CMD52 is issued when SD_BUF becomes empty).
3. After the INFO0 interrupt processing in SD_INFO1 due to the issuing of CMD52 has been completed, restart writing to SD_BUF by DMA transfer.

(5) Erroneous Detection of SDIO interrupts

The SD host interface erroneously detects an SDIO interrupt when all of the following conditions are satisfied.

1. SDIO access
2. The SDIO device supports the SDIO interrupt (an optional part of the standard).
3. The SDIO device supports 4-bit width multi-block read of CMD53 (an optional part of the standard).
4. The SDIO device does not support the SDIO interrupt between 4-bit width data blocks (an optional part of the standard).

Avoid this problem by following the procedure below when the values of the SMB and S4MI bits in the card capability register of the SDIO device are 1 and 0, respectively.

1. Immediately before executing the multiple block read command, clear (to 0) the IOMOD bit of the SDIO_MODE register.
2. Execute the command.
3. Set (to 1) the IOMOD bit of the SDIO_MODE register immediately after multiple block read operation is completed.

(6) Setting STP bit during multiple read

During multiple block read with auto CMD12 setting 1 to SEC bit of SD_STOP, if performing a force-quit by setting 1 to STP bit of SD_STOP, the command sequence is not terminated depending on the timing of setting 1 to STP bit of SD_STOP.
To avoid this, if setting STP bit of SD_STOP during multiple block transfer, setting 0 to SEC bit of SD_STOP at the same time.
Switching the value of SEC bit of SD_STOP from 1 to 0, even if CBSY bit of SD_INFO2 is 1.
In case the command sequence is not terminated because of not setting 0 to SEC bit of SD_STOP at the same time, setting SDRST bit of SOFT_RST is also available to terminate the command sequence.
If performing a force-quit by setting 1 to IOABT bit of SDIO_MODE during CMD53 multiple block transfer, leave the value (=1) of SEC bit of SD_STOP as it is.

chanhvinh.com

(7) Software Reset

For transitions to the software reset state by the SDRST bit in the SOFT_RST register, see section 42.3.6, Software Reset. However, where the procedure refers to the SRST bit, read this as the SDRST bit in the SOFT_RST register.

chanhvinh.com

39. MMC Host Interface

The MMC host interface is a host controller conforming to the JEDEC Standard JESD84-A441 that allows connection with devices with the MMC interface.

39.1 Features

- Supports 1/4/8-bit MMC bus
- Supports the Single Data Rate only
- MMC clock frequency = P1ϕ frequency/2^n (n = 1 to 10)
- Supports High Priority Interrupt (HPI)*
- Supports background operation
- Data buffer: 512 bytes × 2
- Three types of interrupt requests: normal operation, error/timeout, card detection
- DMA transfer requests: buffer write and buffer read
- Card detection function

Note: * Supports HPI in the sequences of CMD6, CMD24, CMD25 (pre-defined), and CMD38.

Figure 39.1 shows a block diagram of this module.

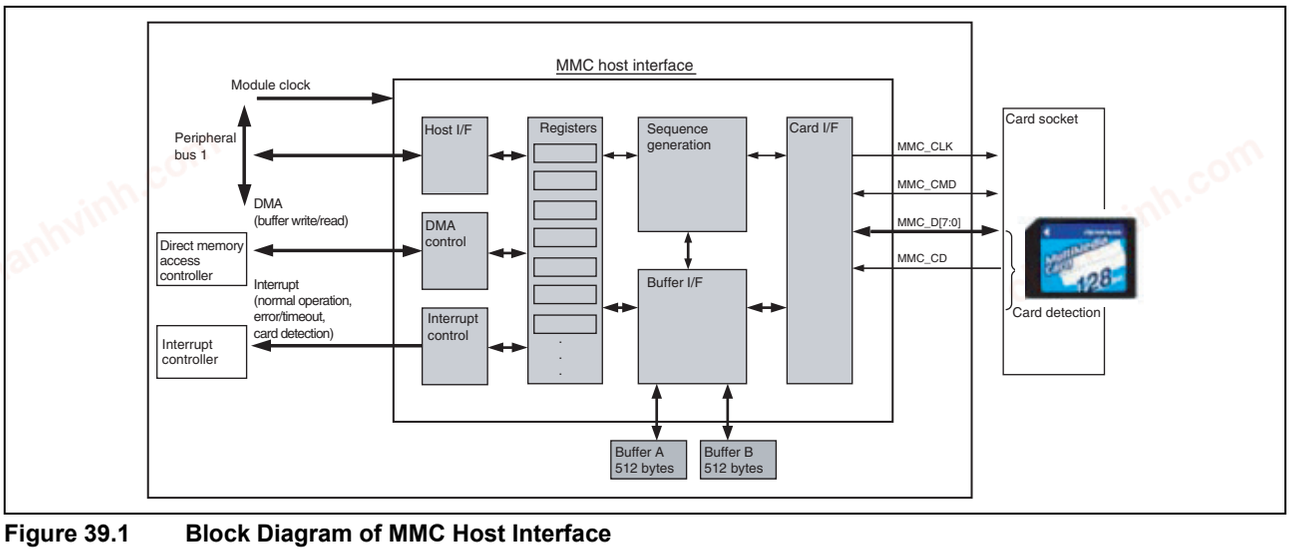


Figure 39.1 Block Diagram of MMC Host Interface

39.2 Input/Output Pins

Table 39.1 shows the pin configuration of this module.

Table 39.1 Pin Configuration

Pin Name	I/O	Function
MMC_CLK	Output	MMC clock
MMC_CMD	Input/output	Command/response
MMC_D[7:0]	Input/output	Transmit/receive data
MMC_CD	Input	Card detection*1

Note 1. Check the specifications of the card socket to be used before connection.

chanhvinh.com

39.3 Register Descriptions

Table 39.2 shows the register configuration of this module.

Table 39.2 Register Configuration

Register Name	Abbreviation	R/W	Address	Access Size
Command setting register	CE_CMD_SET	R/W	H'E804C800 H'E804C802	16 16
Argument register	CE_ARG	R/W	H'E804C808	32
Argument register for automatically-issued CMD12	CE_ARG_CMD12	R/W	H'E804C80C	32
Command control register	CE_CMD_CTRL	R/W	H'E804C810	32
Transfer block setting register	CE_BLOCK_SET	R/W	H'E804C814	32
Clock control register	CE_CLK_CTRL	R/W	H'E804C818	32
Buffer access configuration register	CE_BUF_ACC	R/W	H'E804C81C	32
Response register 3	CE_RESP3	R	H'E804C820	32
Response register 2	CE_RESP2	R	H'E804C824	32
Response register 1	CE_RESP1	R	H'E804C828	32
Response register 0	CE_RESP0	R	H'E804C82C	32
Response register for automatically-issued CMD12	CE_RESP_CMD12	R	H'E804C830	32
Data register	CE_DATA	R/W	H'E804C834	32
Interrupt flag register	CE_INT	R/W	H'E804C840	32
Interrupt enable register	CE_INT_EN	R/W	H'E804C844	32
Status register 1	CE_HOST_STS1	R	H'E804C848	32
Status register 2	CE_HOST_STS2	R	H'E804C84C	32
DMA mode setting register	CE_DMA_MODE	R/W	H'E804C85C	32
Card detection/port control register	CE_DETECT	R/W	H'E804C870	32
Special mode setting register	CE_ADD_MODE	R/W	H'E804C874	32
Version register	CE_VERSION	R/W	H'E804C87C	32

Note: • Do not access registers other than those shown above.

chanhvinh.com

39.3.1 Command Setting Register (CE_CMD_SET)

CE_CMD_SET sets a command sequence.

The command sequence starts when the settings have been made in bits 31 to 16. Note that writing to CE_CMD_SET is disabled while a command sequence is proceeding (i.e., the value of CMDSEQ in CE_HOST_STS1 is 1).

CE_CMD_SET should be set according to the description in section 39.7.12, Setting Values of CE_CMD_SET. This register should be accessed in 16 bits. (The address of bits 31 to 16 is H'E804C800 and that of bits 15 to 0 is H'E804C802.)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	CMD[5:0]						RTYP[1:0]		RBSY	—	WDAT	DWEN	CMLTE	CMD12EN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RIDXC[1:0]		RCRC7C[1:0]		—	CRC16C	—	CRCSTE	TBIT	OPDM	—	—	SBIT	—	DATW[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R	R/W	R/W	R/W	R	R	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	00	R	Reserved These bits are always read as 0. The write value should always be 0.
29 to 24	CMD[5:0]	All 0	R/W	Command Index These bits set a command index ([45:40]). Note: Setting a command index in these bits initiates the command sequence.
23, 22	RTYP[1:0]	00	R/W	Response Type 00: No response 01: 6-byte response (R1, R1b, R3, R4, R5) 10: 17-byte response (R2) 11: Setting prohibited
21	RBSY	0	R/W	Response Busy Select Selects whether “busy” is involved in response reception. 0: No response busy 1: Response busy involved (R1b)
20	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
19	WDAT	0	R/W	Presence/Absence of Data 0: No data 1: With data
18	DWEN	0	R/W	Read/Write (valid when “with data” is selected) 0: Read from the card 1: Write to the card
17	CMLTE	0	R/W	Single/Multi Block Transfer Select (valid when “with data” is selected) 0: Single-block transfer 1: Multi-block transfer
16	CMD12EN	0	R/W	Automatic CMD12 Issuance (valid when multi-block transfer is selected)* 0: Does not issue CMD12 automatically. 1: Issues CMD12 automatically (= automatic CMD12) For details of automatic CMD12 issuance, see section 39.6.4, Automatic CMD12 Issuance. Note: Set the transfer block size to 512 bytes. Set RBSY to 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
15, 14	RIDXC[1:0]	00	R/W	Response Index Check Specify the items to be checked for [45:40] of a 6-byte response or [133:128] of a 17-byte response. 00: Checks the response index (ensure that the response index matches the command index.) 01: Checks the check bits (ensure that the bits are all 1.) 10: No checking 11: Setting prohibited
13, 12	RCRC7C [1:0]	00	R/W	Response CRC7 Check Specify the items to be checked for [7:1] of a 6-byte response or a 17-byte response. 00: Checks CRC7 (set the response type to 01) 01: Checks the check bits (set the response type to 01) 10: Checks internal CRC7 (R2 only) (set the response type to 10) 11: No checking
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	CRC16C	0	R/W	CRC16 Check in Reception (valid when “with data” and “read” are selected) 0: Checks CRC16 1: Does not check CRC16 (use when CMD14)
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	CRCSTE	0	R/W	CRC Status Reception (valid when “with data” and “write” are selected) 0: Receives CRC status 1: Does not receive CRC status (use when CMD19)
7	TBIT	0	R/W	Transmission Bit Setting 0: Sets the transmission bit ([46]) to 1. 1: Sets the transmission bit ([46]) to 0.
6	OPDM	0	R/W	Open-Drain Output Mode 0: Normal output 1: Open-drain output Note: This setting is only applied to the MMC_CMD line.
5, 4	—	00	R	Reserved These bits are always read as 0. The write value should always be 0.
3	SBIT	0	R/W	Read Data Start Bit Detection (valid when "with data" and "read" are selected) 0: Detects the start bit when the MMC_D signals set by DATW are all 0. 1: Detects the start bit when MMC_D[0] is 0.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1, 0	DATW[1:0]	00	R/W	Data Bus Width Setting (valid when "with data" is selected) 00: 1 bit 01: 4 bits 10: 8 bits 11: Setting prohibited

Note: * When HPI is executed, use multi-block transfer (pre-defined) without setting this bit to 1.

chanhvinh.com

39.3.2 Argument Register (CE_ARG)

CE_ARG sets the argument for the command to be transmitted. Set CE_ARG before starting the command sequence.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ARG[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ARG[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	ARG[31:0]	H'00000000	R/W	These bits set [39:8] of a command. Note: Set the argument of automatically-issued CMD12 by CE_ARG_CMD12.

39.3.3 Argument Register for Automatically-Issued CMD12 (CE_ARG_CMD12)

CE_ARG_CMD12 is used to set the argument for the automatically-issued CMD12 in multi-block transfer. For the automatically-issued CMD12, see section 39.6.4, Automatic CMD12 Issuance. Set CE_ARG_CMD12 before starting the command sequence.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	C12ARG[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	C12ARG[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	C12ARG[31:0]	H'00000000	R/W	These bits set [39:8] of the automatically-issued CMD12.

chanhvinh.com

39.3.4 Command Control Register (CE_CMD_CTRL)

CE_CMD_CTRL is used to terminate a command sequence forcibly.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BREAK
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	BREAK	0	R/W	Forcible Termination of Command Sequence To discontinue the current command sequence, write 1 to this bit while the bit is 0 and then write 0. After the above procedure, check if the value of the CMDSEQ bit in CE_HOST_STS1 has become 0 and then perform a software reset. Note: Since a software reset returns the register value to the initial value, the register needs be set again.

39.3.5 Transfer Block Setting Register (CE_BLOCK_SET)

CE_BLOCK_SET specifies the size of the block and the number of blocks for the data to be transferred. Set CE_BLOCK_SET before starting the command sequence.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BLKCNT[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BLKSIZ[15:0]															
Initial value:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	BLKCNT [15:0]	H'0000	R/W	Number of Blocks for Transfer Note: This setting is valid for multi-block transfer.
15 to 0	BLKSIZ [15:0]	H'0200	R/W	Transfer Block Size Note: Transfer block size should be set as follows. • Single-block transfer: 1 to 512 bytes • Multi-block transfer: 512 bytes

chanhvinh.com

39.3.6 Clock Control Register (CE_CLK_CTRL)

CE_CLK_CTRL controls the MMC clock and sets timeout values. Do not change the setting of this register while a command sequence is in progress.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	CLKEN	—	—	—	—	CLKDIV[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SRSPTO[1:0]		SRBSYTO[3:0]				SRWDTO[3:0]				—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	CLKEN	0	R/W	MMC Clock Output Control 0: Does not output the MMC clock (tied to low level) 1: Outputs the MMC clock
23 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19 to 16	CLKDIV[3:0]	0000	R/W	MMC Clock Frequency Setting 0000: P1ϕ/2 0001: P1ϕ/2^2 0111: P1ϕ/2^8 1000: P1ϕ/2^9 1001: P1ϕ/2^{10} 1010 to 1111: Setting prohibited
15, 14	—	00	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	SRSPTO [1:0]	00	R/W	Response Timeout Setting Specifies the timeout period for the RSPTO bit of CE_INT. 00: 64 MMC clock cycles 01: 128 MMC clock cycles 10: 256 MMC clock cycles 11: Setting prohibited
11 to 8	SRBSYTO [3:0]	0000	R/W	Response Busy Timeout Setting Specifies the timeout period for the RBSYTO bit of CE_INT. 0000: 2^{14} MMC clock cycles 0001: 2^{15} MMC clock cycles : 1110: 2^{28} MMC clock cycles 1111: 2^{29} MMC clock cycles
7 to 4	SRWDTO [3:0]	0000	R/W	Write Data/Read Data Timeout Setting Specifies the timeout period for the WDATTO and RDATTO bits of CE_INT. 0000: 2^{14} MMC clock cycles 0001: 2^{15} MMC clock cycles : 1110: 2^{28} MMC clock cycles 1111: 2^{29} MMC clock cycles
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

39.3.7 Buffer Access Configuration Register (CE_BUF_ACC)

CE_BUF_ACC configures the method of accessing data registers and mode of DMA transfer. Do not change the setting of this register while a command sequence is in progress. For explanation of the buffers, see section 39.6.3, Buffer Structure and Buffer Accesses.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	DMAWEN	DMAREN	—	—	—	—	—	—	—	ATYP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

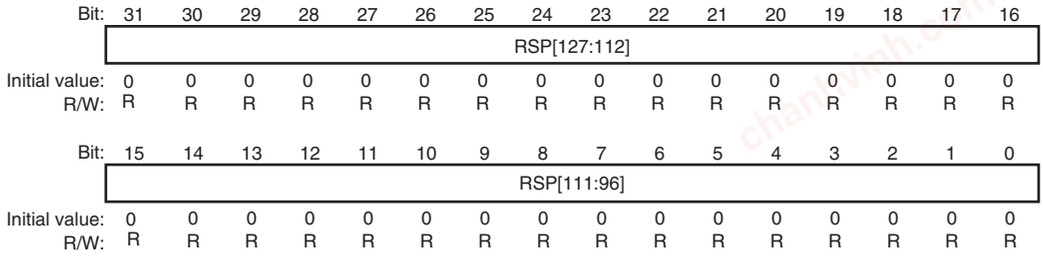
Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25	DMAWEN	0	R/W	Buffer Write DMA Transfer Request Enable 0: Disables DMA transfer request for buffer writing 1: Enables DMA transfer request for buffer writing
24	DMAREN	0	R/W	Buffer Read DMA Transfer Request Enable 0: Disables DMA transfer request for buffer reading 1: Enables DMA transfer request for buffer reading
23 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	ATYP	0	R/W	Buffer Access Select 0: When not swapped byte-wise. 1: When swapped byte-wise. Note: For explanation of buffer access, see section 39.6.3, Buffer Structure and Buffer Accesses.
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

39.3.8 Response Registers 3 to 0 (CE_RESP3 to CE_RESP0)

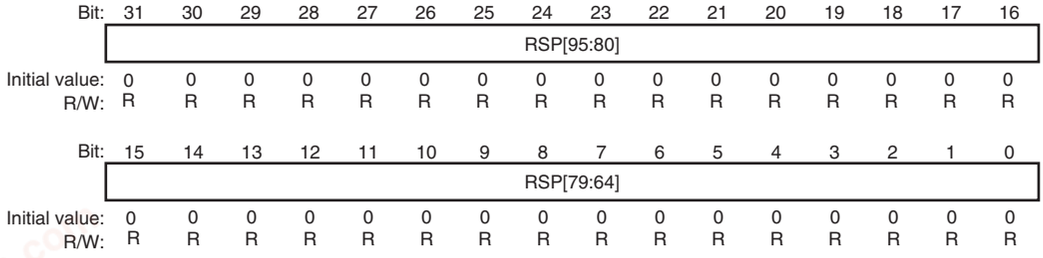
CE_RESP3 to CE_RESP0 are the registers for storing the response that has been received. For the formats of response values, see section 39.6.1, Command/Response Formats.

- CE_RESP3



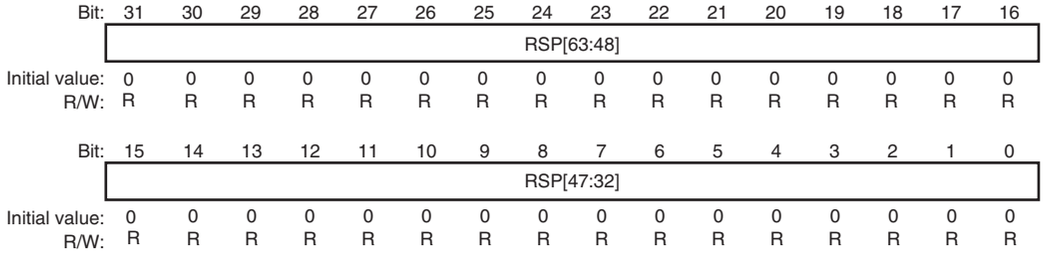
Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RSP[127:96]	H'00000000	R	[127:96] of a 17-byte response are stored.

- CE_RESP2



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RSP[95:64]	H'00000000	R	[95:64] of a 17-byte response are stored.

- CE_RESP1



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RSP[63:32]	H'00000000	R	[63:32] of a 17-byte response are stored.

chanhvinh.com

- CE_RESP0

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RSP[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RSP[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RSP[31:0]	H'00000000	R	[39:8] of a 6-byte response or [31:0] of a 17-byte response are stored. Note: The response to the automatically-issued CMD12 is stored in CE_RESP_CMD12.

39.3.9 Response Register for Automatically-Issued CMD12 (CE_RESP_CMD12)

CE_RESP_CMD12 is the register for storing the response to the automatically-issued CMD12.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RSP12[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RSP12[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RSP12[31:0]	H'00000000	R	[39:8] of a response to the automatically-issued CMD12 are stored.

39.3.10 Data Register (CE_DATA)

CE_DATA is used to access the buffers of this module.

For the write/read data formats, see section 39.6.2, Data Block Format.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DATA[31:0]	H'00000000	R/W	Buffer write/read data [31:0]

chanhvinh.com

39.3.11 Interrupt Flag Register (CE_INT)

CE_INT indicates various status during execution of a command sequence. Each bit is set to 1 when its setting condition has been met. To clear flag(s), write 0 only to the bit(s) to be cleared and write 1 to the other bits.

For the operation in the case of an error or timeout, see section 39.6.7, Operation in the Case of Error/Timeout.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	CMD12 DRE	CMD12 RBE	CMD12 CRE	DTRAN E	BUFR E	BUFW EN	BUFR EN	—	—	RBSY E	CRSP E
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMD VIO	BUF VIO	—	—	WDAT ERR	RDAT ERR	RIDX ERR	RSP ERR	—	—	—	CRCS TO	WDAT TO	RDAT TO	RBSY TO	RSP TO
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bets are always read as 0. The writing value should always be 0.
26	CMD12DRE	0	R/W*1	Automatic CMD12 Issuance & Buffer Read Complete [Setting condition] Response busy for automatically-issued CMD12 and buffer reading have been completed. [Clearing condition] Writing a 0 to this bit Note: When CMD12DRE has been set, CMD12RBE, CMD12CRE, and BUFRE have also been set. So, these bits should be cleared as well.
25	CMD12RBE	0	R/W*1	Automatic CMD12 Issuance Response Busy Complete [Setting condition] Reception of the response and response busy for an automatically-issued CMD12 have been completed. [Clearing condition] Writing a 0 to this bit Note: When CMD12RBE is set, CMD12CRE is also set. So, clear the bit as well. When CMD12RBE is set during a multi-block write, DTRANE is also set. So clear the bit as well.
24	CMD12CRE	0	R/W*1	Automatic CMD12 Response Complete [Setting condition] The response to an automatically-issued CMD12 has been received. [Clearing condition] Writing a 0 to this bit
23	DTRANE	0	R/W*1	Data Transmission Complete [Setting conditions] Transmission of all blocks of data has been completed. • When configured to receive CRC status: Completion of busy (data busy) after reception of CRC status • When configured not to receive CRC status: Completion of data transmission [Clearing condition] Writing a 0 to this bit
22	BUFRE	0	R/W*1	Buffer Read Complete [Setting condition] All blocks of data have been received and the data have been read from the buffer [Clearing condition] Writing a 0 to this bit

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
21	BUFWEN	0	R/W*1	Buffer Write Ready [Setting condition] The buffer has become empty and ready for writing. [Clearing condition] Writing a 0 to this bit Note: When data is written to CE_DATA by the CPU, clear this bit first, and then write the amount of data equivalent to the block size set in the CE_BLOCK_SET register. Note that this bit is not set when DMA transfer request for buffer writing is enabled.
20	BUFREN	0	R/W*1	Buffer Read Ready [Setting condition] Transfer block size of data have been stored in the buffer and it has become ready for reading [Clearing condition] Writing a 0 to this bit Note: When data is read from CE_DATA by the CPU, clear this bit first, and then read the amount of data equivalent to the block size set in the CE_BLOCK_SET register. Note that this bit is not set when DMA transfer request for buffer reading is enabled.
19, 18	—	00	R	Reserved These bits are always read as 0.The write value should always be 0.
17	RBSYE	0	R/W*1	Response Busy Complete [Setting condition] Reception of a response and response busy have been completed [Clearing condition] Writing a 0 to this bit Note: When RBSYE has been set, CRSPE has also been set. So, this bit should be cleared as well. Completion of reception of the response and response busy for automatically-issued CMD12 is reflected in CMD12RBE.
16	CRSPE	0	R/W*1	Command/Response Complete [Setting conditions] A command has been transmitted or a response has been received • When configured not to receive response: A command has been transmitted • When configured to receive 6- or 17-byte response: A response has been received [Clearing condition] Writing a 0 to this bit Note: Completion of reception of the response to automatically-issued CMD12 is reflected in CMD12CRE.
15	CMDVIO	0	R/W*1	Command Issuance Error [Setting conditions] Illegal setting has been made in CE_CMD_SET or CE_BLOCK_SET • During execution of a command sequence: Writing to CMD[5:0] in CE_CMD_SET (The command sequence is not stopped automatically.) • At the start of command sequence: Writing to CMD[5:0] in CE_CMD_SET when the registers have been set for one of the following combinations of selection - No response + response busy - No response + with data - No data + automatic CMD12 issuance - With data + single-block transfer + automatic CMD12 issuance - With data + response busy + automatic CMD12 issuance - With data + transfer block size = 0 - With data + transfer block size ≥ 513 - With data + multi-block transfer + number of blocks for transfer = 0 [Clearing condition] Writing a 0 to this bit

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
14	BUFVIO	0	R/W*1	Buffer Access Error [Setting conditions] Illegal buffer access has been attempted. • CE_ DATA has been accessed exceeding the block size set in BLKSIZ[15:0] in CE_BLOCK_SET • While data is being read from the card: CE_DATA has been accessed with BUFREN not set (when DMA is used, with no DMA transfer request asserted for buffer reading) • While data is being written to the card: CE_DATA has been accessed with BUFWEN not set (when DMA is used, with no DMA transfer request asserted for buffer writing) [Clearing condition] Writing a 0 to this bit Note: When BUFVIO is set, the command sequence is not stopped automatically. When an error occurs, this bit may be set.
13, 12	—	00	R	Reserved These bits are always read as 0. The write value should always be 0.
11	WDATERR	0	R/W*1	Write Data Error [Setting conditions] Error is found in the data that has been written. • Error is in the status of the CRC status • Error is in the end bit of the CRC status [Clearing condition] Writing a 0 to this bit Note: When WDATERR is set, the command sequence is stopped automatically.
10	RDATERR	0	R/W*1	Read Data Error [Setting conditions] Error is found in the read data. • Error is in CRC16 of the read data • Error is in the end bit of the read data [Clearing condition] Writing a 0 to this bit Note: When RDATERR is set, the command sequence is stopped automatically.
9	RIDXERR	0	R/W*1	Response Index Error [Setting condition] Error has been found in the index value of the response. • When an error has been found in [45:40] of a 6-byte response (including automatically-issued CMD12) or [133:128] of a 17-byte response (The items to be checked are set by RIDXC in CE_CMD_SET.) [Clearing condition] Writing a 0 to this bit Note: When RIDXERR is set, the command sequence is stopped automatically.
8	RSPERR	0	R/W*1	Response Error [Setting conditions] Error has been found in the response values of the response. • Transmission bit in the response is high • Error is in the end bit of the response • When an error has been found in [7:1] of a 6-byte response (including automatically-issued CMD12) or a 17-byte response (The items to be checked are set by RCRC7C in CE_CMD_SET.) [Clearing condition] Writing a 0 to this bit Note: When RSPERR is set, the command sequence is stopped automatically.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
4	CRCSTO	0	R/W*1	CRC Status Timeout [Setting condition] CRC status could not be received [Clearing condition] Writing a 0 to this bit Note: The command sequence is not stopped even if CRCSTO is set.
3	WDATTO	0	R/W*1	Write Data Timeout [Setting condition] The busy status remains unchanged after the period set by SRWDTO in CE_CLK_CTRL after the CRC status was received. [Clearing condition] Writing a 0 to this bit Note: The command sequence is not stopped even if WDATTO is set.
2	RDATTO	0	R/W*1	Read Data Timeout [Setting conditions] • Read data could not be received within the period set by SRWDTO in CE_CLK_CTRL after the read command was transmitted • Read data could not be received within the period set by SRWDTO in CE_CLK_CTRL after the read data was received. [Clearing condition] Writing a 0 to this bit Note: The command sequence is not stopped even if RDATTO is set.
1	RBSYTO	0	R/W*1	Response Busy Timeout [Setting condition] The busy status remains unchanged after the period set by SRBSYTO in CE_CLK_CTRL after the command (including automatically-issued CMD12) was transmitted. [Clearing condition] Writing a 0 to this bit Note: The command sequence is not stopped even if RBSYTO is set.
0	RSPTO	0	R/W*1	Response Timeout [Setting condition] Response could not be received within the period set by SRSPTO in CE_CLK_CTRL after the command (including automatically-issued CMD12) was transmitted. [Clearing condition] Writing a 0 to this bit Note: The command sequence is not stopped even if RSPTO is set.

Note 1. A 0 is the only value that can be written to the bit. Writing a 1 is ignored.

chanhvinh.com

39.3.12 Interrupt Enable Register (CE_INT_EN)

CE_INT_EN controls output of the CE_INT-related interrupt signals. If a flag in CE_INT is set to 1 while its corresponding bit in CE_INT_EN is set to 1, an interrupt request is output. For details on interrupt requests, see section 39.4, Interrupt Requests.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	MCMD12DRE	MCMD12RBE	MCMD12CRE	MDTRANE	MBUFRE	MBUFWEN	MBUFREN	—	—	MRBSYE	MCRSPE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MCMDVIO	MBUFVIO	—	—	MWDATERR	MRDATERR	MRIDXERR	MRSPERR	—	—	—	MCRCSTO	MWDATTO	MRDATTO	MRBSYTO	MRSPTO
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26	MCMD12DRE	0	R/W	CMD12DRE Interrupt Enable 0: Disables interrupt output by the CMD12DRE flag 1: Enables interrupt output by the CMD12DRE flag
25	MCMD12RBE	0	R/W	CMD12RBE Interrupt Enable 0: Disables interrupt output by the CMD12RBE flag 1: Enables interrupt output by the CMD12RBE flag
24	MCMD12CRE	0	R/W	CMD12CRE Interrupt Enable 0: Disables interrupt output by the CMD12CRE flag 1: Enables interrupt output by the CMD12CRE flag
23	MDTRANE	0	R/W	DTRANE Interrupt Enable 0: Disables interrupt output by the DTRANE flag 1: Enables interrupt output by the DTRANE flag
22	MBUFRE	0	R/W	BUFRE Interrupt Enable 0: Disables interrupt output by the BUFRE flag 1: Enables interrupt output by the BUFRE flag
21	MBUFWEN	0	R/W	BUFWEN Interrupt Enable 0: Disables interrupt output by the BUFWEN flag 1: Enables interrupt output by the BUFWEN flag
20	MBUFREN	0	R/W	BUFREN Interrupt Enable 0: Disables interrupt output by the BUFREN flag 1: Enables interrupt output by the BUFREN flag
19, 18	—	00	R	Reserved These bits are always read as 0. The write value should always be 0.
17	MRBSYE	0	R/W	RBSYE Interrupt Enable 0: Disables interrupt output by the RBSYE flag 1: Enables interrupt output by the RBSYE flag
16	MCRSPE	0	R/W	CRSPE Interrupt Enable 0: Disables interrupt output by the CRSPE flag 1: Enables interrupt output by the CRSPE flag
15	MCMDVIO	0	R/W	CMDVIO Interrupt Enable 0: Disables interrupt output by the CMDVIO flag 1: Enables interrupt output by the CMDVIO flag
14	MBUFVIO	0	R/W	BUFVIO Interrupt Enable 0: Disables interrupt output by the BUFVIO flag 1: Enables interrupt output by the BUFVIO flag
13, 12	—	00	R	Reserved These bits are always read as 0. The write value should always be 0.
11	MWDATERR	0	R/W	WDATERR Interrupt Enable 0: Disables interrupt output by the WDATERR flag 1: Enables interrupt output by the WDATERR flag

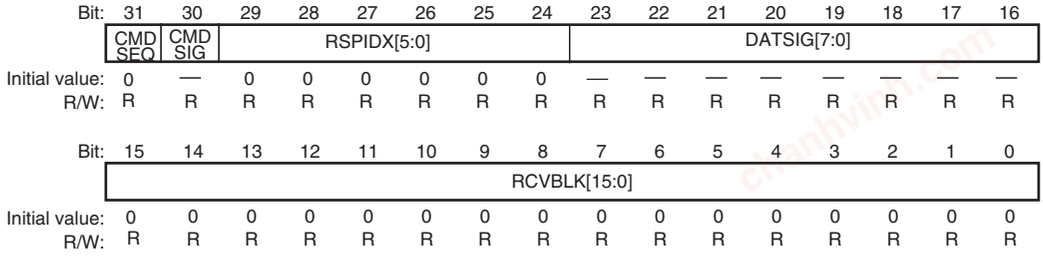
chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
10	MRDATERR	0	R/W	RDATERR Interrupt Enable 0: Disables interrupt output by the RDATERR flag 1: Enables interrupt output by the RDATERR flag
9	MRIDXERR	0	R/W	RIDXERR Interrupt Enable 0: Disables interrupt output by the RIDXERR flag 1: Enables interrupt output by the RIDXERR flag
8	MRSPERR	0	R/W	RSPERR Interrupt Enable 0: Disables interrupt output by the RSPERR flag 1: Enables interrupt output by the RSPERR flag
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	MCRCSTO	0	R/W	CRCSTO Interrupt Enable 0: Disables interrupt output by the CRCSTO flag 1: Enables interrupt output by the CRCSTO flag
3	MWDATTO	0	R/W	WDATTO Interrupt Enable 0: Disables interrupt output by the WDATTO flag 1: Enables interrupt output by the WDATTO flag
2	MRDATTO	0	R/W	RDATTO Interrupt Enable 0: Disables interrupt output by the RDATTO flag 1: Enables interrupt output by the RDATTO flag
1	MRBSYTO	0	R/W	RBSYTO Interrupt Enable 0: Disables interrupt output by the RBSYTO flag 1: Enables interrupt output by the RBSYTO flag
0	MRSPTO	0	R/W	RSPTO Interrupt Enable 0: Disables interrupt output by the RSPTO flag 1: Enables interrupt output by the RSPTO flag

chanhvinh.com

39.3.13 Status Register 1 (CE_HOST _STS1)

CE_HOST_STS1 indicates the number of blocks that have been transferred, states of the MMC_CMD and MMC_D lines, index of the received response, and command sequence status.



Bit	Bit Name	Initial Value	R/W	Description
31	CMDSEQ	0	R	Command Sequence Status 0: Command sequence is in the initial state 1: Command sequence is being executed
30	CMDSIG	Undefined	R	MMC_CMD Line Status Indicates the state on the MMC_CMD line.
29 to 24	RSPIDX [5:0]	H'00	R	Response Index Indicate [45:40] of a 6-byte response or [133:128] of a 17-byte response.
23 to 16	DATSIG [7:0]	Undefined	R	MMC_D Status Indicate the states on the MMC_D[7:0] lines. Note: When an error or timeout occurs, MMCDAT[0] may remain 0.
15 to 0	RCVBLK [15:0]	H'0000	R	Number of Transferred Blocks Indicate the number blocks that have been transferred. • When the DWEN bit in CE_CMD_SET is 0 Number of blocks read from the card • When the DWEN bit in CE_CMD_SET is 1 Number of blocks written to the card

chanhvinh.com

39.3.14 Status Register 2 (CE_HOST _STS2)

CE_HOST _STS2 indicates timeout and error status.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CRCSTE	CRC16E	AC12CRCE	RSPCRC7E	CRCSTEBE	RDATEBE	AC12REBE	RSPEBE	AC12IDXE	RSPIDXE	—	—	—	CRCST[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	STRDATTO	DATBSYTO	CRCSTTO	AC12BSYTO	RSPBSYTO	AC12RSPTO	STRSPTO	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	CRCSTE	0	R	CRC Status Error This bit is set to 1 when an error is found in the CRC status value.
30	CRC16E	0	R	Read Data CRC16 Error This bit is set to 1 when an error is found in CRC16 in the read data.
29	AC12CRCE	0	R	Automatic CMD12 Response CRC7 Error This bit is set to 1 when an error is found in [7:1] of the response to the automatically-issued CMD12. Note: The items to be checked are set by RCRC7C in CE_CMD_SET.
28	RSPCRC7E	0	R	Command Response CRC7 Error (other than automatically-issued CMD12) This bit is set to 1 when an error is found in [7:1] of a 6-byte response or a 17-byte response. Note: The items to be checked are set by RCRC7C in CE_CMD_SET.
27	CRCSTEBE	0	R	CRC Status End Bit Error This bit is set to 1 when an error is found in the end bit in CRC status.
26	RDATEBE	0	R	Read Data End Bit Error This bit is set to 1 when an error is found in the end bit in the read data.
25	AC12REBE	0	R	Automatic CMD12 Response End Bit Error This bit is set to 1 when an error is found in the end bit of the response to the automatically-issued CMD12.
24	RSPEBE	0	R	Command Response End Bit Error (other than automatically-issued CMD12) This bit is set to 1 when an error is found in the end bit of the response.
23	AC12IDXE	0	R	Automatic CMD12 Response Index Error This bit is set to 1 when an error is found in [45:40] of the response to the automatically-issued CMD12. Note: The items to be checked are set by RIDXC in CE_CMD_SET.
22	RSPIDXE	0	R	Command Response Index Error (other than automatically-issued CMD12) This bit is set to 1 when an error is found in [45:40] of a 6-byte response or [133:128] of a 17-byte response. Note: The items to be checked are set by RIDXC in CE_CMD_SET.
21 to 19	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
18 to 16	CRCST [2:0]	000	R	CRC Status Indicate the value of the CRC status that has been received.
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	STRDATTO	0	R	Read Data Timeout • This bit is set to 1 if read data is not received within the period set by the SRWDTO bits in CE_CLK_CTRL after a read command was transmitted. • This bit is set to 1 if read data is not received within the period set by the SRWDTO bits in CE_CLK_CTRL after a read data was received.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
13	DATBSYTO	0	R	Data Busy Timeout This bit is set to 1 if busy status remains unchanged after the period set by the SRWDTO bits in CE_CLK_CTRL after the CRC status was received.
12	CRCSTTO	0	R	CRC Status Timeout This bit is set to 1 if CRC status could not be received.
11	AC12BSYTO	0	R	Automatic CMD12 Response Busy Timeout This bit is set to 1 if busy state remains unchanged after the period set by the SRBSYTO bits in CE_CLK_CTRL after the automatically-issued CMD12 was transmitted.
10	RSPBSYTO	0	R	Response Busy Timeout This bit is set to 1 if busy state remains unchanged after the period set by the SRBSYTO bits in CE_CLK_CTRL after a command (other than automatically-issued CMD12) was transmitted.
9	AC12RSPTO	0	R	Automatic CMD12 Response Timeout This bit is set to 1 if the response is not received within the period set by the SRSPTO bits in CE_CLK_CTRL after the automatically-issued CMD12 was transmitted.
8	STRSPTO	0	R	Response Timeout This bit is set to 1 if the response is not received within the period set by the SRSPTO bits in CE_CLK_CTRL after a command (other than automatically-issued CMD12) was transmitted.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

39.3.15 DMA Mode Setting Register (CE_DMA_MODE)

CE_DMA_MODE sets the transfer unit for DMA transfer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMA SEL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	DMASEL	0	R/W	DMA Transfer Size Select Selects the transfer unit for CE_DATA read/write DMA transfer. Set this bit in combination with the transfer size set in the DMA channel configuration register. 0: 2-byte (word) or 4-byte (longword) unit 1: 64-byte (longword × 16) unit Note: For DMA transfer in 64-byte units, set address H'E804C800 as the destination or source of DMA transfer instead of the data register (CE_DATA). Moreover, when a transfer error or timeout occurs during DMA transfer in 64-byte units, leading to the forced termination of transfer, write 0 to this bit and then set it to 1 again. Also do this whenever a software reset is used.

chanhvinh.com

39.3.16 Card Detection/Port Control Register (CE_DETECT)

CE_DETECT controls card detection. For details on interrupt requests by card detection, see section 39.4, Interrupt Requests.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	CD SIG	CD RISE	CD FALL	—	—	—	—	—	—	MCD RISE	MCD FALL	—	—	—	—
Initial value:	0	—	0	0	0	—	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W*	R/W*	R	R	R	R	R	R	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
14	CDSIG	Undefined	R	MMC_CD Pin Status Indication Indicates the MMC_CD pin status.
13	CDRISE	0	R/W*1	MMC_CD Pin Rise Detection Flag [Setting condition] The MMC_CD pin level changes from low to high. [Clearing condition] Writing a 0 to this bit
12	CDFALL	0	R/W*1	MMC_CD Pin Fall Detection Flag [Setting condition] The MMC_CD pin level changes from high to low. [Clearing condition] Writing a 0 to this bit
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	—	Undefined	R	Reserved The write value should always be 0.
9 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	MCDRISE	0	R/W	CDRISE Interrupt Enable 0: Disables interrupt output by the CDRISE flag. 1: Enables interrupt output by the CDRISE flag.
4	MCDFALL	0	R/W	CDFALL Interrupt Enable 0: Disables interrupt output by the CDFALL flag. 1: Enables interrupt output by the CDFALL flag.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note 1. A 0 is the only value that can be written to the bit. Writing a 1 is ignored.

chanhvinh.com

39.3.17 Special Mode Setting Register (CE_ADD_MODE)

CE_ADD_MODE controls the internal clock.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	CLK MAIN	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19	CLKMAIN	0	R/W	Internal Clock Control 0: Normal mode 1: Low power consumption mode (only card detection is enabled)
18 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

39.3.18 Version Register (CE_VERSION)

CE_VERSION indicates the version number and controls software reset of this module.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SW RST	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VERSION[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	SWRST	0	R/W	Software Reset 0: Software reset cleared (normal operation) 1: Executes software reset. When SWRST is set to 1, all the register values are reset to the initial values. (SWRST is not reset to the initial value)
30 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	VERSION [15:0]	H'0003	R	Version Information Indicates the version number of this module.

chanhvinh.com

39.4 Interrupt Requests

Table 39.3 shows the interrupt request specifications of this module.

This module generates three types of interrupt requests: normal operation, error/timeout, and card detection. When an interrupt flag is set to 1 and also the corresponding interrupt is enabled, an interrupt request is asserted.

Table 39.3 Specifications of Interrupt Requests

Flag Register	Bit	Mask Register	Bit	Interrupt Request
CE_INT	CMD12DRE	CE_INT_EN	MCMD12DRE	Normal operation interrupt (MMC2)
	CMD12RBE		MCMD12RBE	
	CMD12CRE		MCMD12CRE	
	DTRANE		MDTRANE	
	BUFRE		MBUFRE	
	BUFWEN		MBUFWEN	
	BUFREN		MBUFREN	
	RBSYE		MRBSYE	
	CRSPE		MCRSPE	
	CMDVIO		MCMDVIO	Error/timeout interrupt (MMC1)
	BUFVIO		MBUFVIO	
	WDATERR		MWDATERR	
	RDATERR		MRDATERR	
	RIDXERR		MRIDXERR	
	RSPERR		MRSPERR	
	CRCSTO		MCRCSTO	
	WDATTO		MWDATTO	
	RDATTO		MRDATTO	
	RBSYTO		MRBSYTO	
	RSPTO		MRSPTO	
CE_DETECT	CDRISE	CE_DETECT	MCDRISE	Card detection interrupt (MMC0)
	CDFALL		MCDFALL	

chanhvinh.com

39.5 DMA Specifications

This module provides two channels for DMA transfer requests: one for buffer reading and the other for buffer writing.

39.5.1 DMA for Buffer Writing

The DMA transfer request is asserted for buffer writing when the buffer has become empty while the DMAWEN bit in CE_BUF_ACC is 1.

The DMA transfer request stays asserted for the amount of data specified by BLKSIZ (the block size set in CE_BLOCK_SET) × BLKCNT (the number of blocks for transfer set in CE_BLOCK_SET), and negated after the last block has been transferred. Note that the BUFWEN bit in CE_INT will not be asserted in this case.

If an error occurs during DMA transfer or DMA transfer is forcibly terminated, the command sequence is stopped automatically, which causes the DMA transfer request to be negated.

39.5.2 DMA for Buffer Reading

The DMA transfer request is asserted for buffer reading when the buffer stores data of the block size specified in CE_BLOCK_SET while the DMAREN bit in CE_BUF_ACC is 1.

The DMA transfer request stays asserted for the amount of data specified by BLKSIZ (the block size set in CE_BLOCK_SET) × BLKCNT (the number of blocks for transfer set in CE_BLOCK_SET), and negated after the last block has been transferred. Note that the BUFREN bit in CE_INT will not be asserted in this case.

If an error occurs during DMA transfer or DMA transfer is forcibly terminated, the command sequence is stopped automatically, which causes the DMA transfer request to be negated.

chanhvinh.com

39.6 Operation

39.6.1 Command/Response Formats

Figure 39.2 shows the format of the command to be transferred. The command index that is set in CMD[5:0] of CE_CMD_SET and the argument set in ARG[31:0] of CE_ARG are reflected in the command.

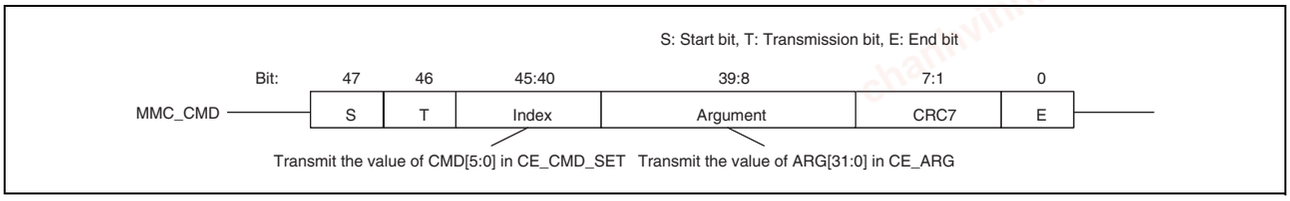


Figure 39.2 Command Format

Figure 39.3 and Figure 39.4 show the formats when a 6-byte response and 17-byte response (R2) are received, respectively. The response index is stored in RSPIDX[5:0] of CE_HOST_STS1, and the status value of the response is stored to CE_RESP0 or CE_RESP3 to CE_RESP0.

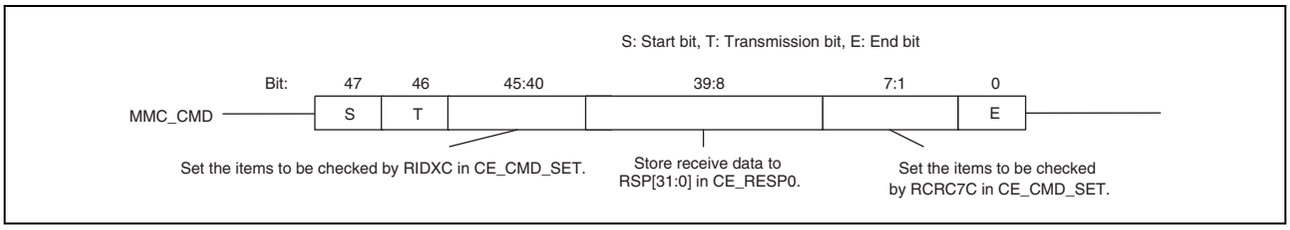


Figure 39.3 Format of 6-Byte Response



Figure 39.4 Format of 17-Byte Response (R2)

chanhvinh.com

39.6.2 Data Block Format

Figure 39.5 shows the format of data blocks. For details on D0 to D3 in Figure 39.5, see section 39.6.3, Buffer Structure and Buffer Accesses. When data is written to a card, data stored in the buffer is transmitted. When data is read from a card, receive data is stored to the buffer.

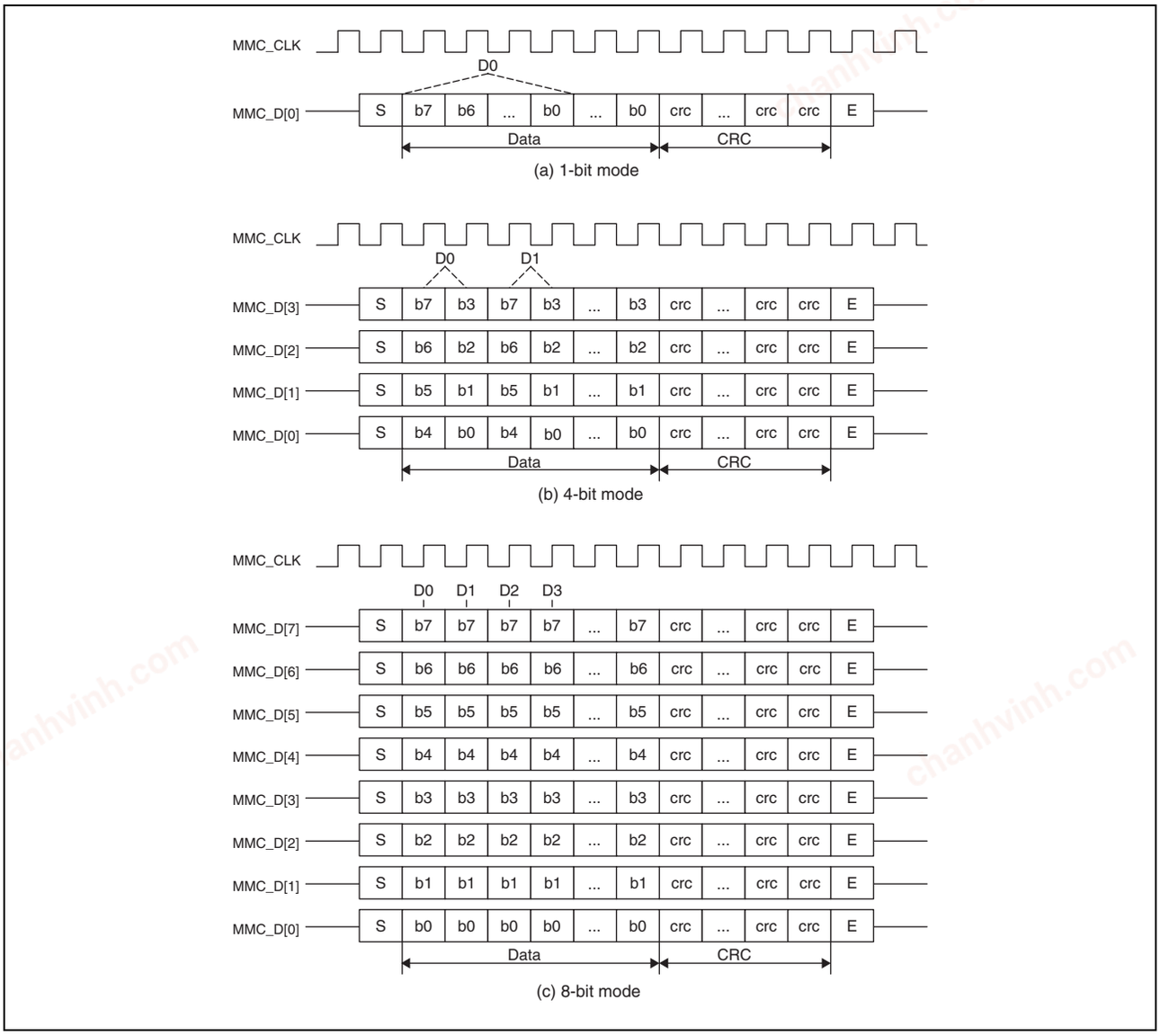


Figure 39.5 Data Block Format

chanhvinh.com

39.6.3 Buffer Structure and Buffer Accesses

This module has two units of 512-byte RAM as shown in Figure 39.6. Therefore, for multi-block writing, if one block of data (512 bytes) stored in one buffer has been transmitted but the other buffer is full, the next block of data can be continuously transmitted. For multi-block reading, if one block of receive data (512 bytes) has been stored in one buffer but the other buffer is empty, the next block of receive data can be continuously stored in the buffer.

If neither of the buffers is empty for multi-block reading, the MMC clock is stopped and reception is suspended. When one of the buffers becomes empty, the MMC clock supply is started to start reception.

The buffer is accessed with CE_DATA. CE_DATA should be accessed for 4 × (n + 1) bytes (n = 0, 1, 2, 3, …, 127).

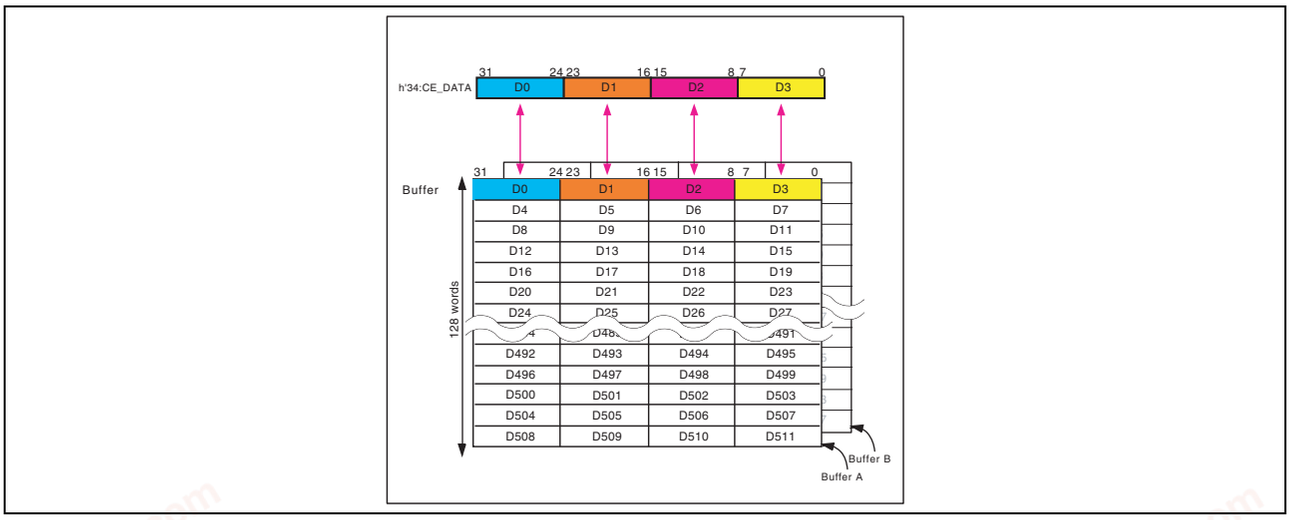


Figure 39.6 Double Buffer Structure

The buffer access select function allows byte-wise swapping of data when the buffer is accessed by writing to or reading from CE_DATA. This function is enabled by the setting of CE_BUFF_ACC. Figure 39.7 shows the specification of byte-wise swapping.

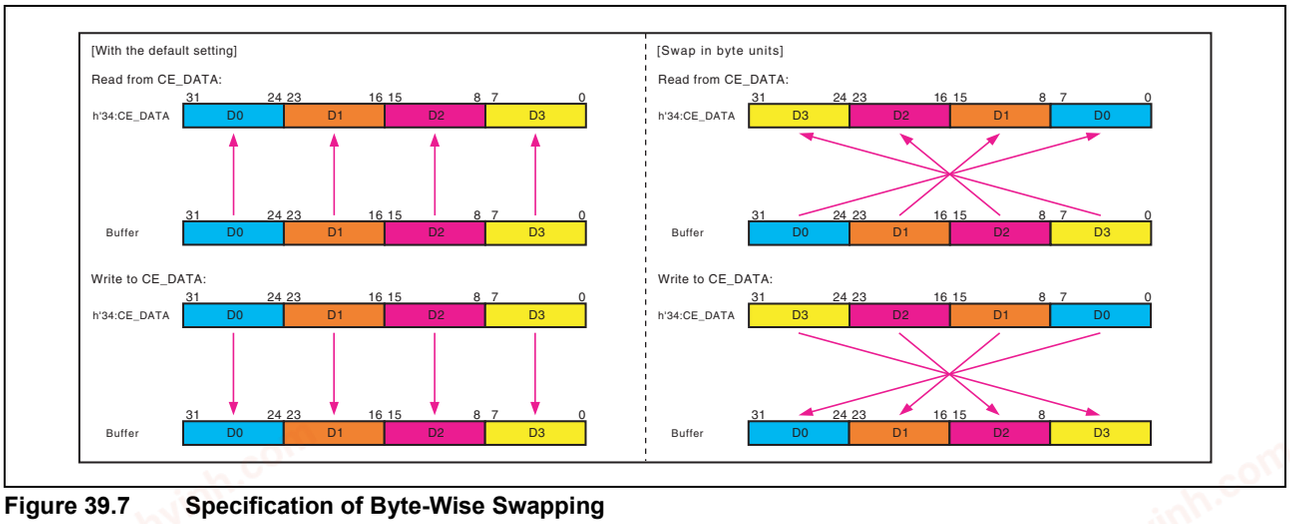


Figure 39.7 Specification of Byte-Wise Swapping

chanhvinh.com

39.6.4 Automatic CMD12 Issuance

This module automatically issues CMD12 when multi-block transfer is performed with the CMD12EN bit in CE_CMD_SET set to 1.

Figure 39.8 shows the timing of automatic CMD12 issuance in multi-block read. CMD12 is issued such that the end bit of the command is sent two bits before the end bit of the data during reception of the last block.

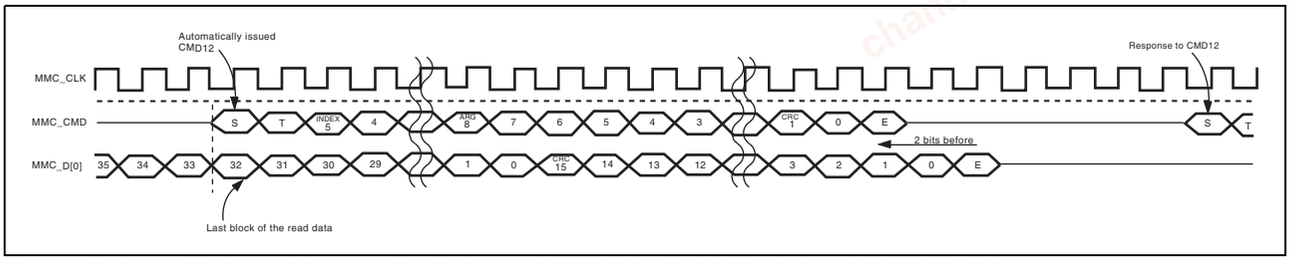


Figure 39.8 Timing of Automatically-Issued CMD12 in Multi-Block Read (1-Bit Mode)

Figure 39.9 shows the timing of automatic CMD12 issuance in multi-block write. CMD12 is issued after the data busy after transmission of the last block has ended.

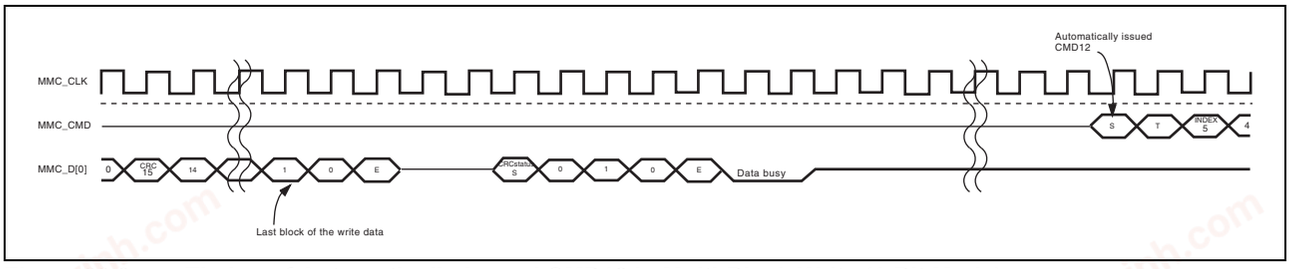


Figure 39.9 Timing of Automatically-Issued CMD12 in Multi-Block Write (1-Bit Mode)

The argument of the automatically-issued CMD12 is set by CE_ARG_CMD12. The value of [39:8] of the response to CMD12 is stored to CE_RESP_CMD12. The busy status on response reception is received.

chanhvinh.com

39.6.5 High Priority Interrupt (HPI)

HPI should be executed as described below with reference to section 39.3, Register Descriptions and section 39.7, Examples of Setting.

(1) When HPI is executed during a write to the device

(a) Terminate the command sequence forcibly.
(b) Wait for the CMDSEQ bit in CE_HOST_STS1 to be cleared to 0.
(c) Issue CMD12 (R1) to make the device in the rcv state enter the prg state. Note that the device does not output the response when the device is in the prg state before issuance of CMD12(R1).
(d) Issue CMD13 (R1).
(e) Issue the HPI command.

(2) When HPI is executed during response busy while not writing to the device

(a) Terminate the command sequence forcibly.
(b) Wait for the CMDSEQ bit in CE_HOST_STS1 to be cleared to 0.
(c) Issue CMD13 (R1).
(d) Issue the HPI command*.

The procedure above is for HPI generated in the CMD6, CMD24, CMD25 (Pre-defined), or CMD38 sequence.

Note: * CMD12 (R1b) or CMD13 (R1b) depending on the device connected

39.6.6 Background Operation

Background operation should be executed as described below with reference to section 39.3, Register Descriptions and section 39.7, Examples of Setting.

To execute background operation, issue CMD6 (R1) to write to the BKOPS_START byte in the EXT_CSD register of the device.

Completion of background operation can be determined by checking the device state with CMD13 (R1) issued after CMD6 (R1) or by polling MMCDAT[0]. When the device state is indicated as “tran” or MMCDAT[0] is high, background operation is determined to be completed.

To suspend background operation, use HPI described in section 39.6.5.

39.6.7 Operation in the Case of Error/Timeout

This module may not be stopped in case of error occurrence. If the command sequence is in progress when an error occurs (check the CMDSEQ bit in CE_HOST_STS1), terminate the sequence forcibly and perform a software reset. The data for transmission or received data that had been stored in the buffers at the time of error occurrence are not guaranteed.

This module is not stopped when a timeout has occurred. To execute the next command after a timeout error has occurred, terminate the sequence forcibly, perform a software reset.

For forcible termination, refer to section 39.7.11, Forcible Termination.

chanhvinh.com

39.7 Examples of Setting

This section shows the procedures for executing typical command sequences.

39.7.1 Legends

Figure 39.10 shows the legends for the symbols used in the figures in the following subsections.

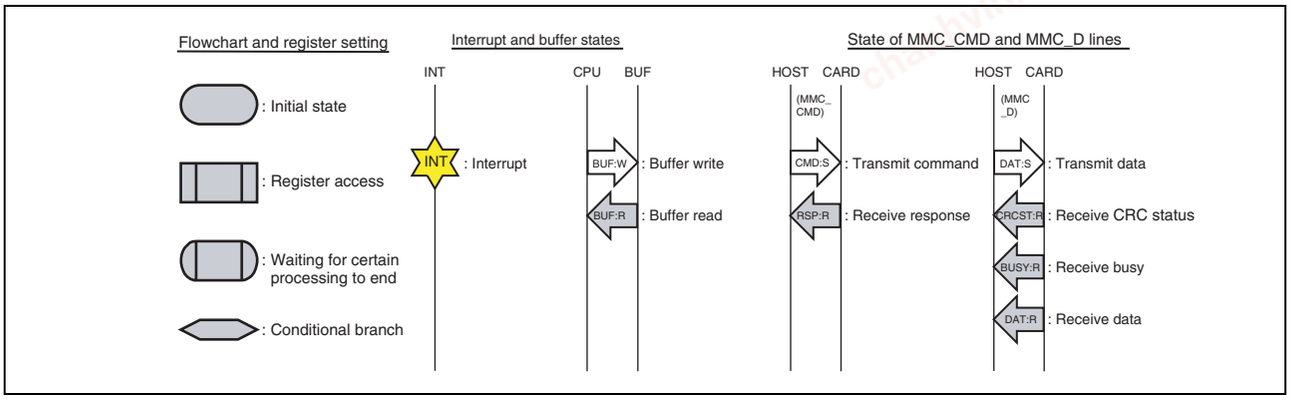


Figure 39.10 Legends for the Symbols Used in the Figures

39.7.2 Command Transmission

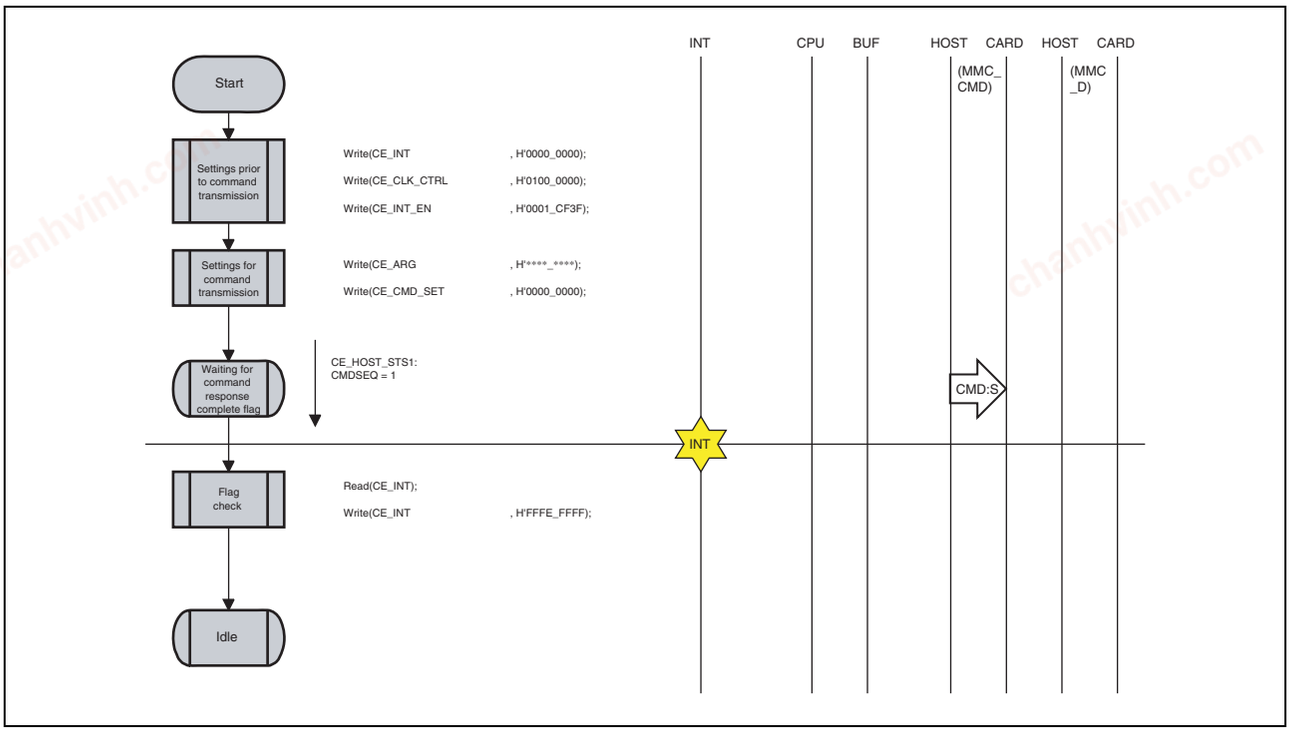


Figure 39.11 Command Transmission (CMD0)

chanhvinh.com

39.7.3 Command Transmission → Response Reception

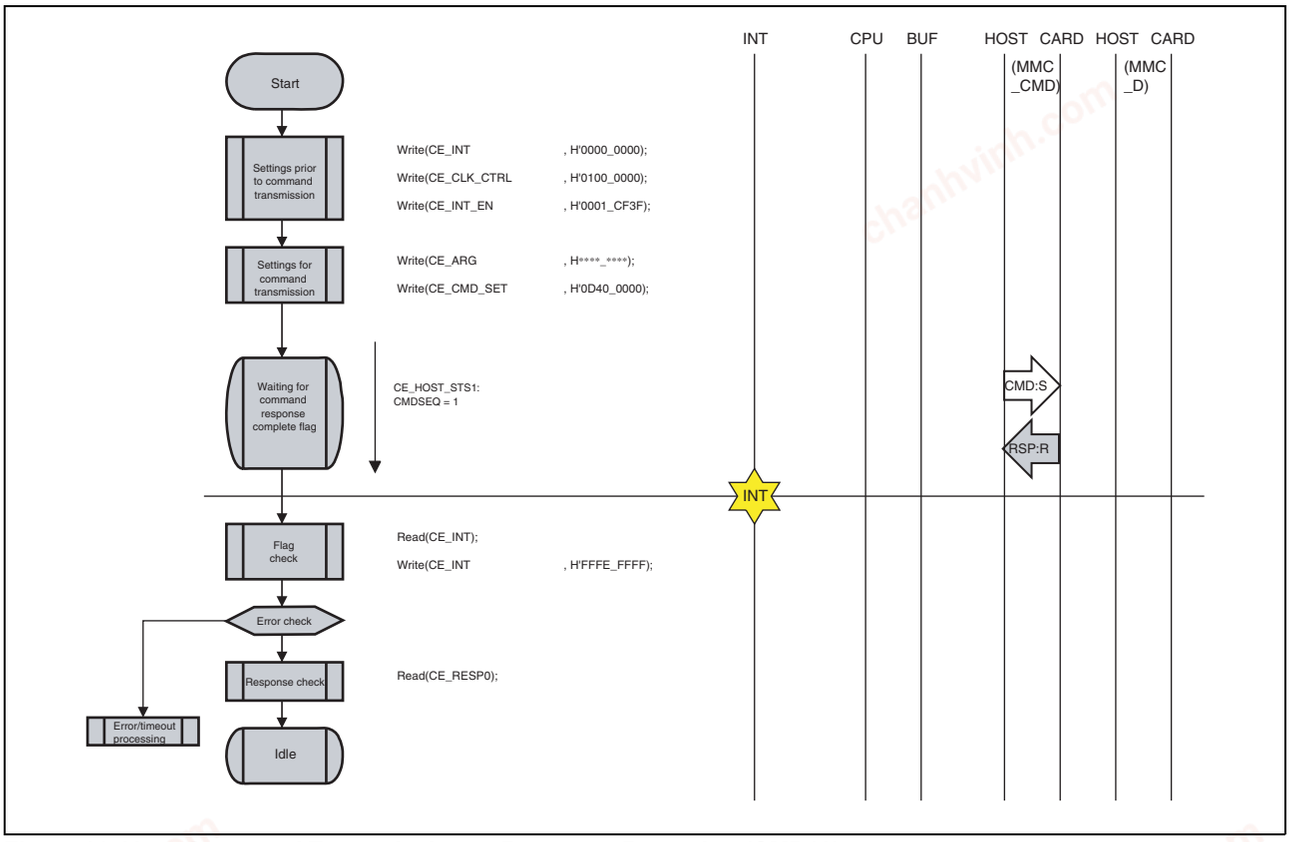


Figure 39.12 Command Transmission → Response Reception (CMD13)

chanhvinh.com

39.7.4 Command Transmission → Response Reception (with Response Busy)

(1) When the busy time period is less than the period set by SRBSYTO in CE_CLK_CTRL

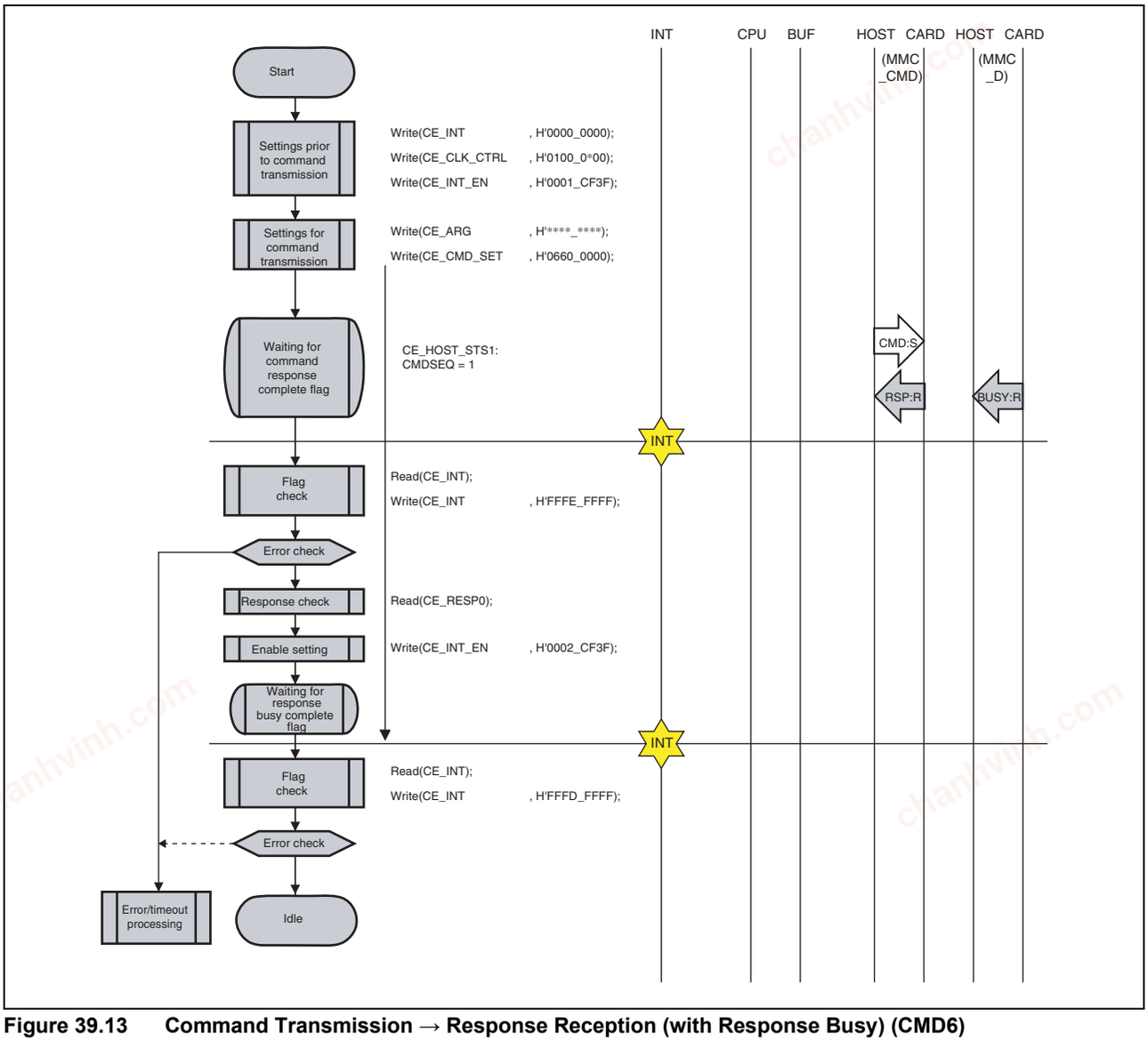


Figure 39.13 Command Transmission → Response Reception (with Response Busy) (CMD6)

chanhvinh.com

(2) When the busy time period may be equal to or beyond the period set by SRBSYTO in CE_CLK_CTRL

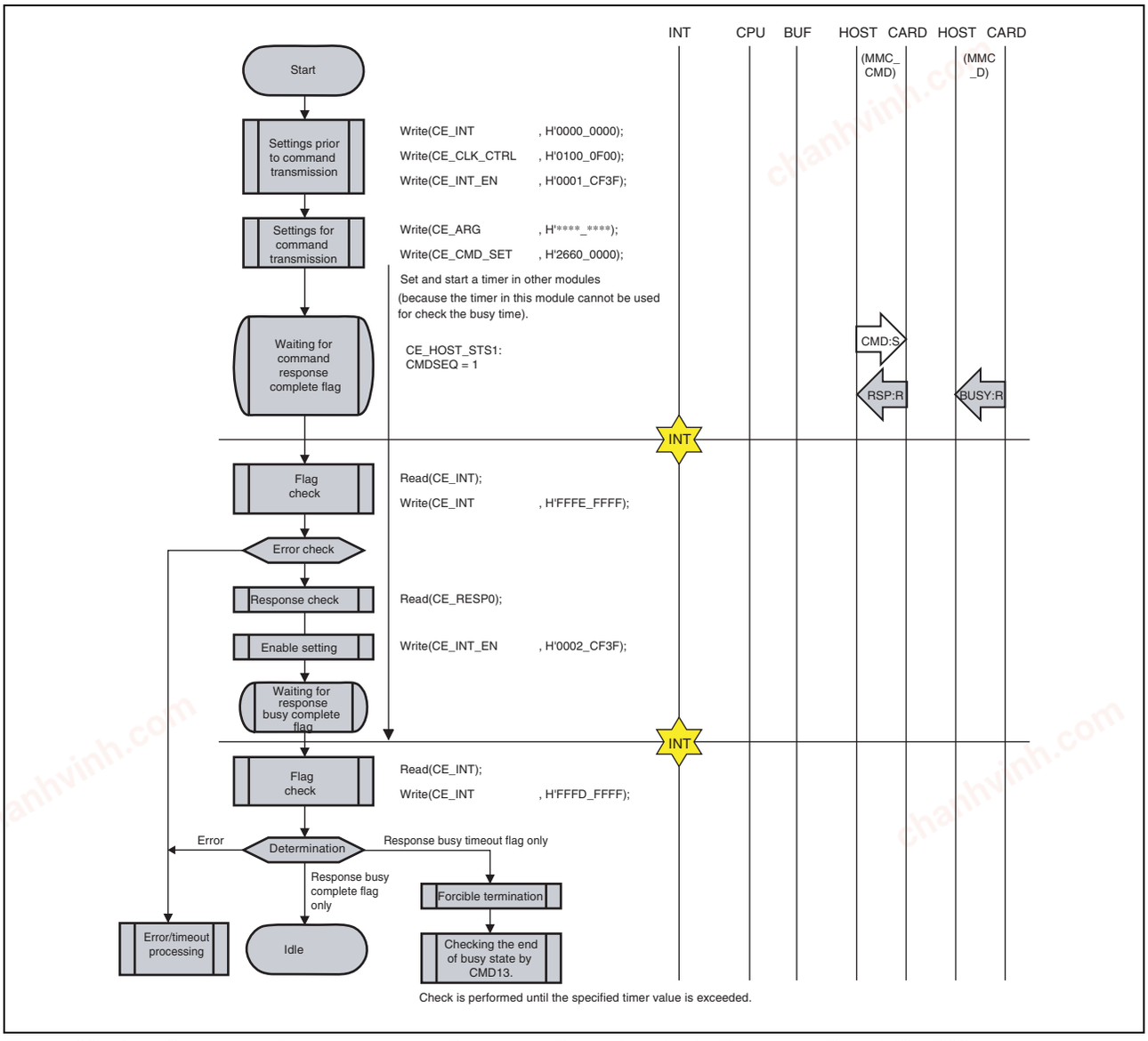


Figure 39.14 Command Transmission → Response Reception (with Response Busy) (CMD38)

chanhvinh.com

39.7.5 Single-Block Read

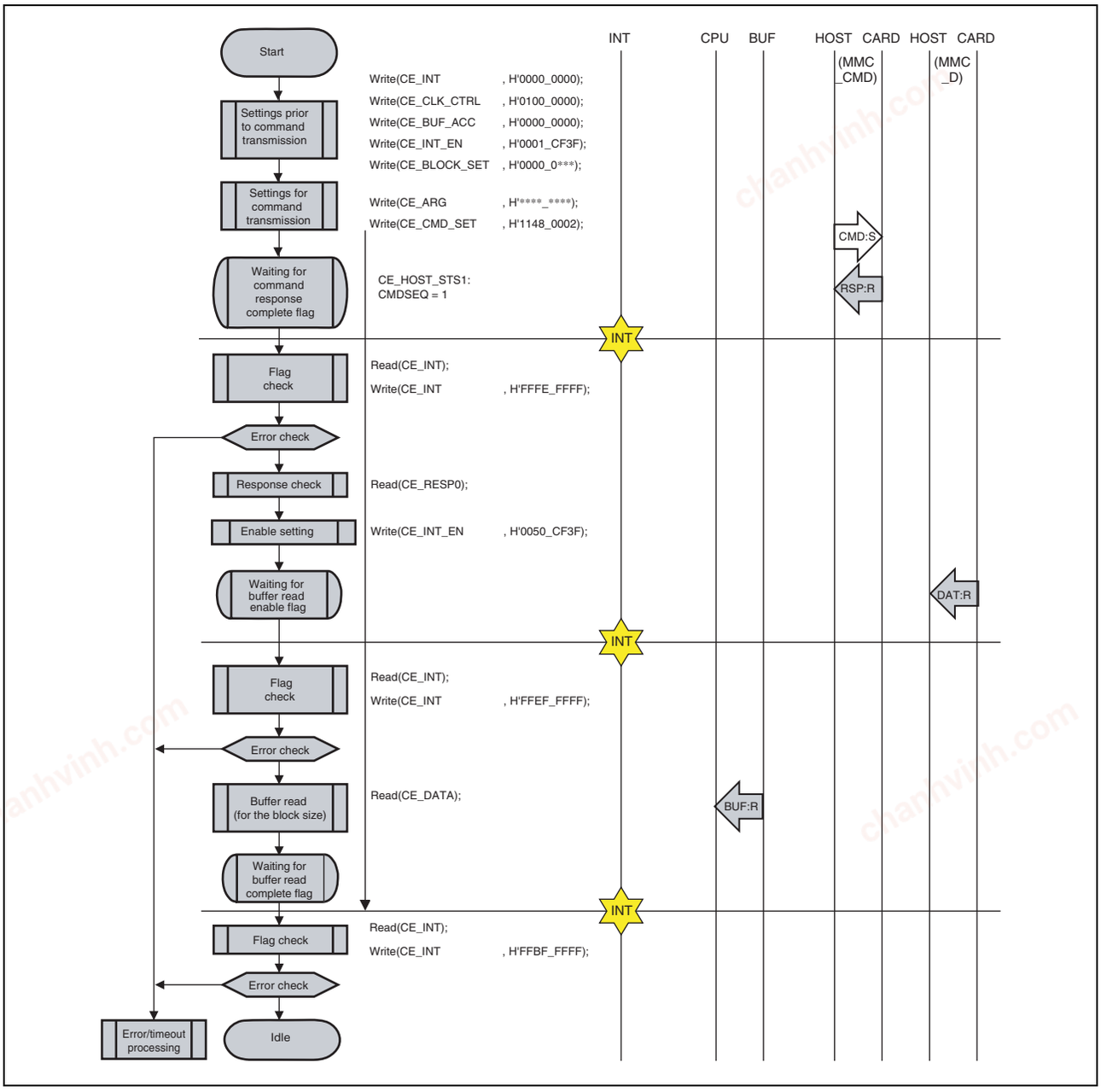


Figure 39.15 Single-Block Read (CMD17)

chanhvinh.com

39.7.6 Multi-Block Read

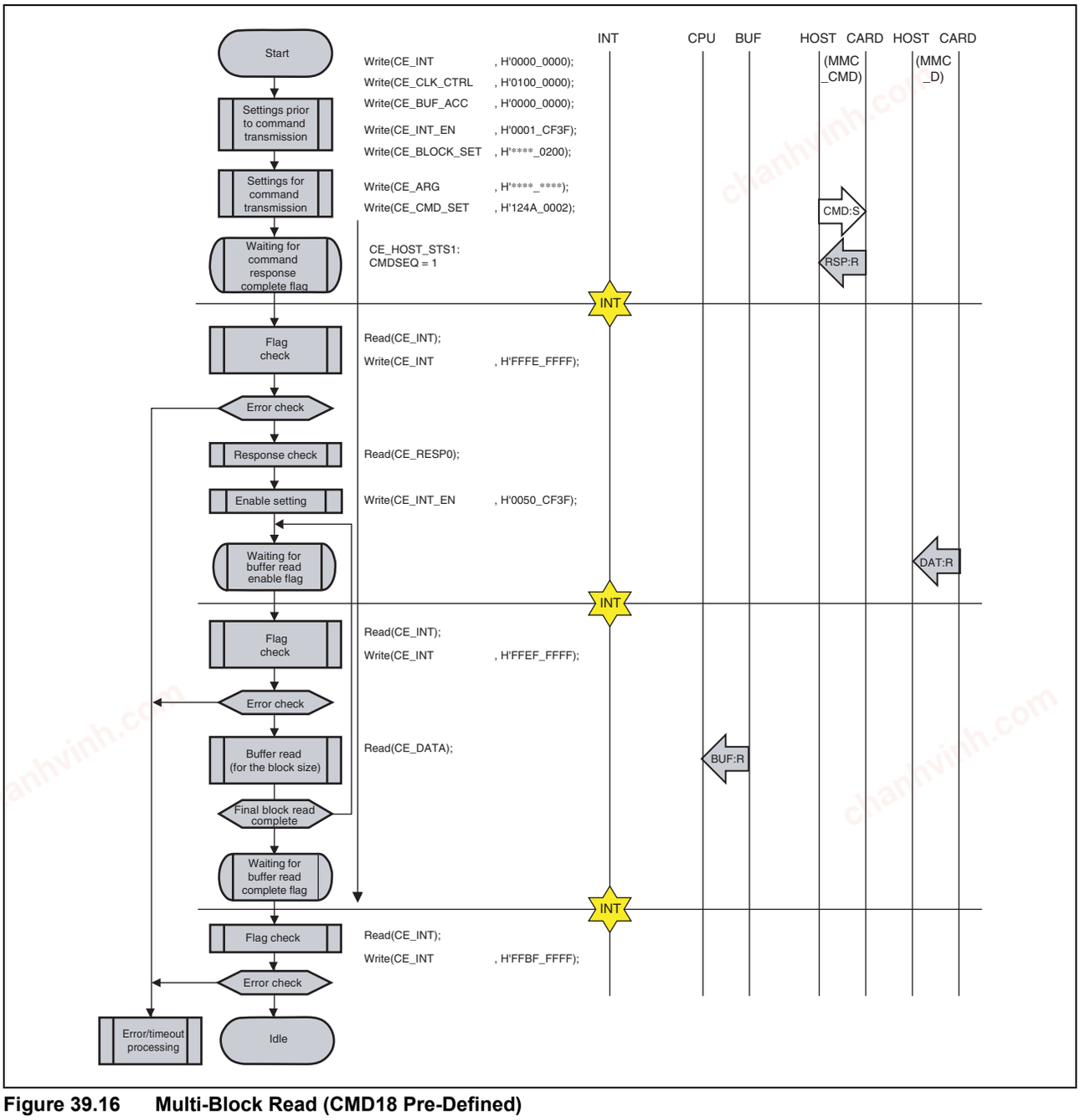


Figure 39.16 Multi-Block Read (CMD18 Pre-Defined)

chanhvinh.com

39.7.7 Multi-Block Read (with Automatic CMD12 Issuance)

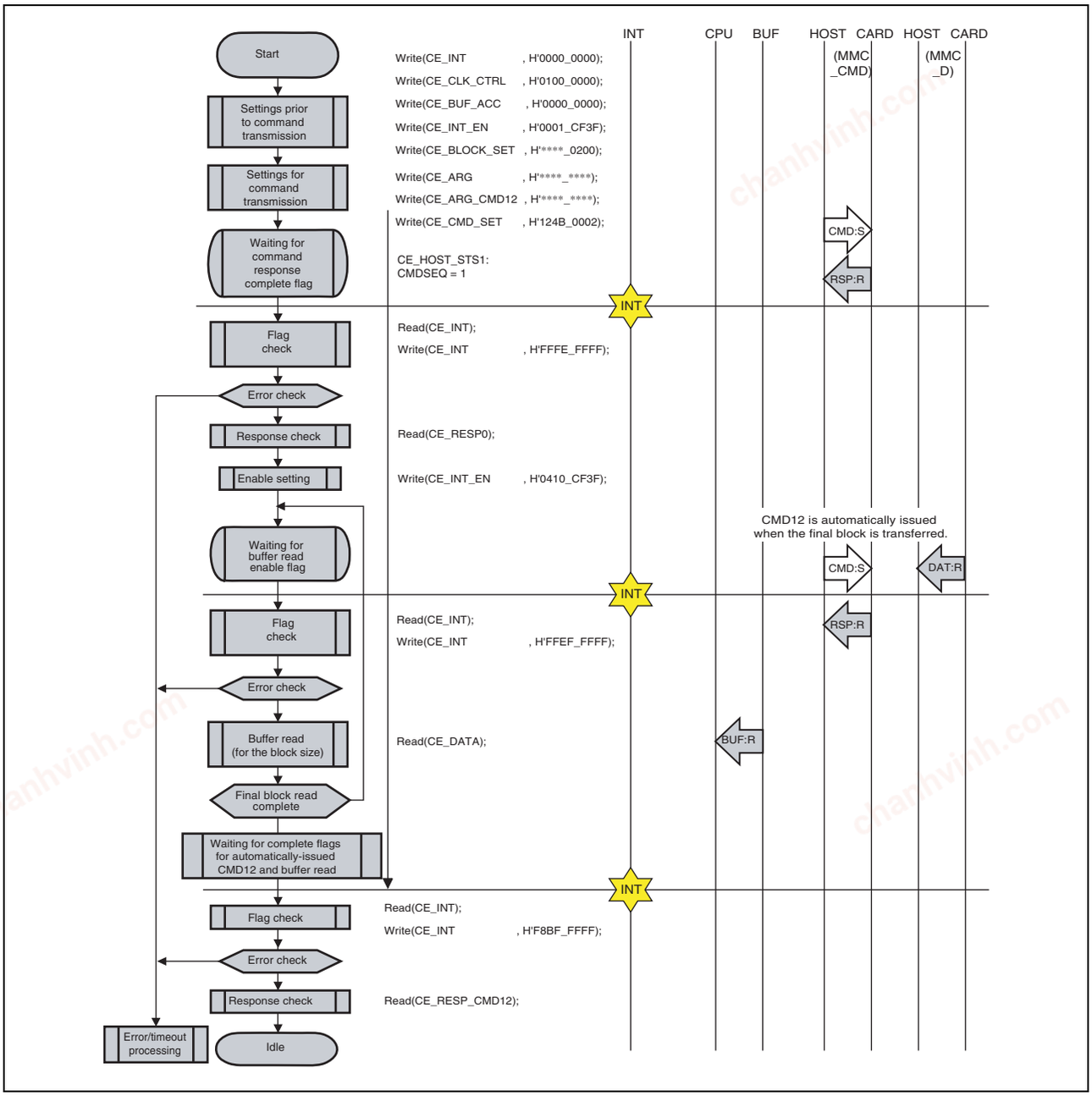


Figure 39.17 Multi-Block Read (with Automatic CMD12 Issuance) (CMD18 Open-Ended)

chanhvinh.com

39.7.8 Single-Block Write

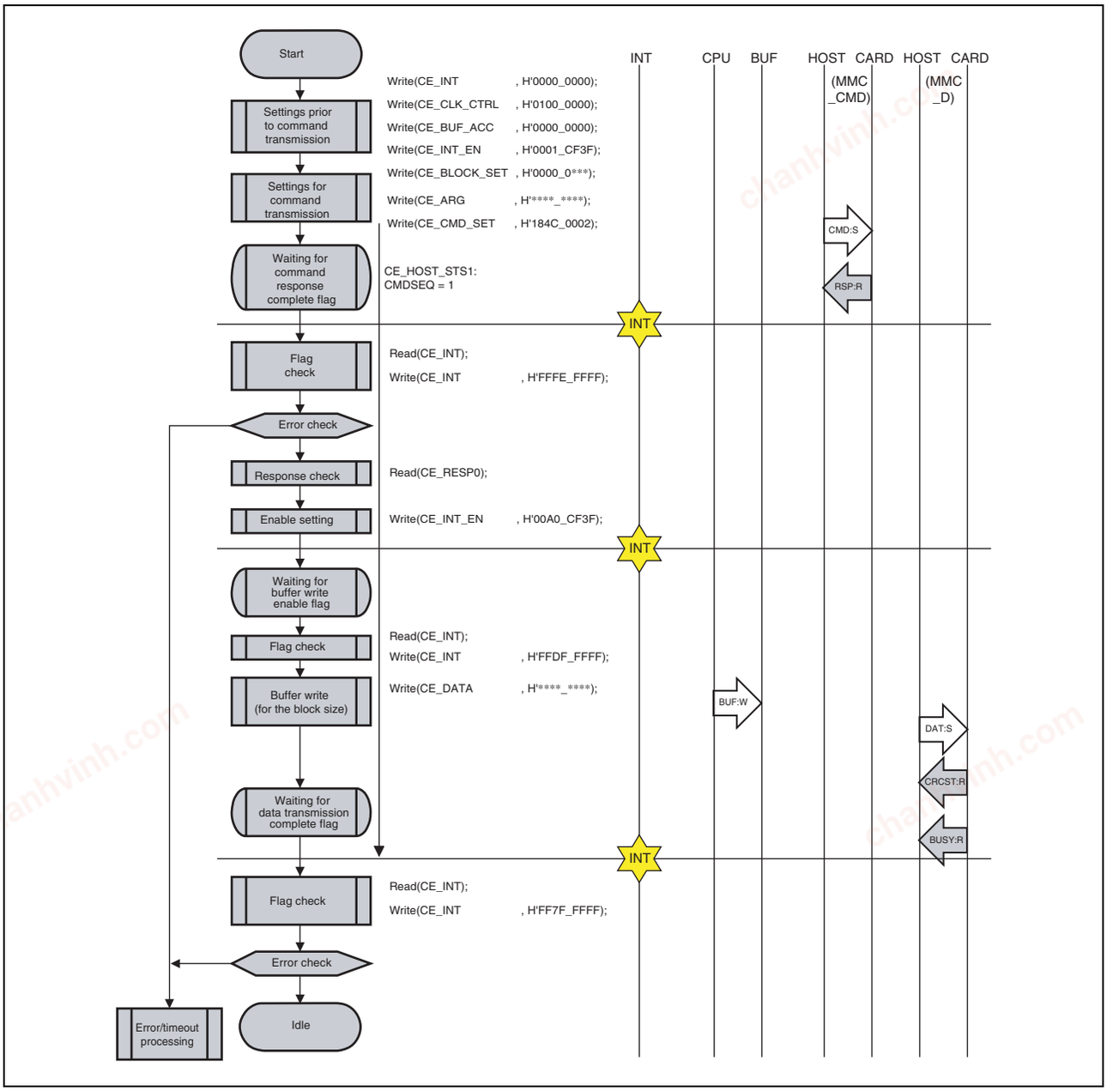


Figure 39.18 Single-Block Write (CMD24)

chanhvinh.com

39.7.9 Multi-Block Write

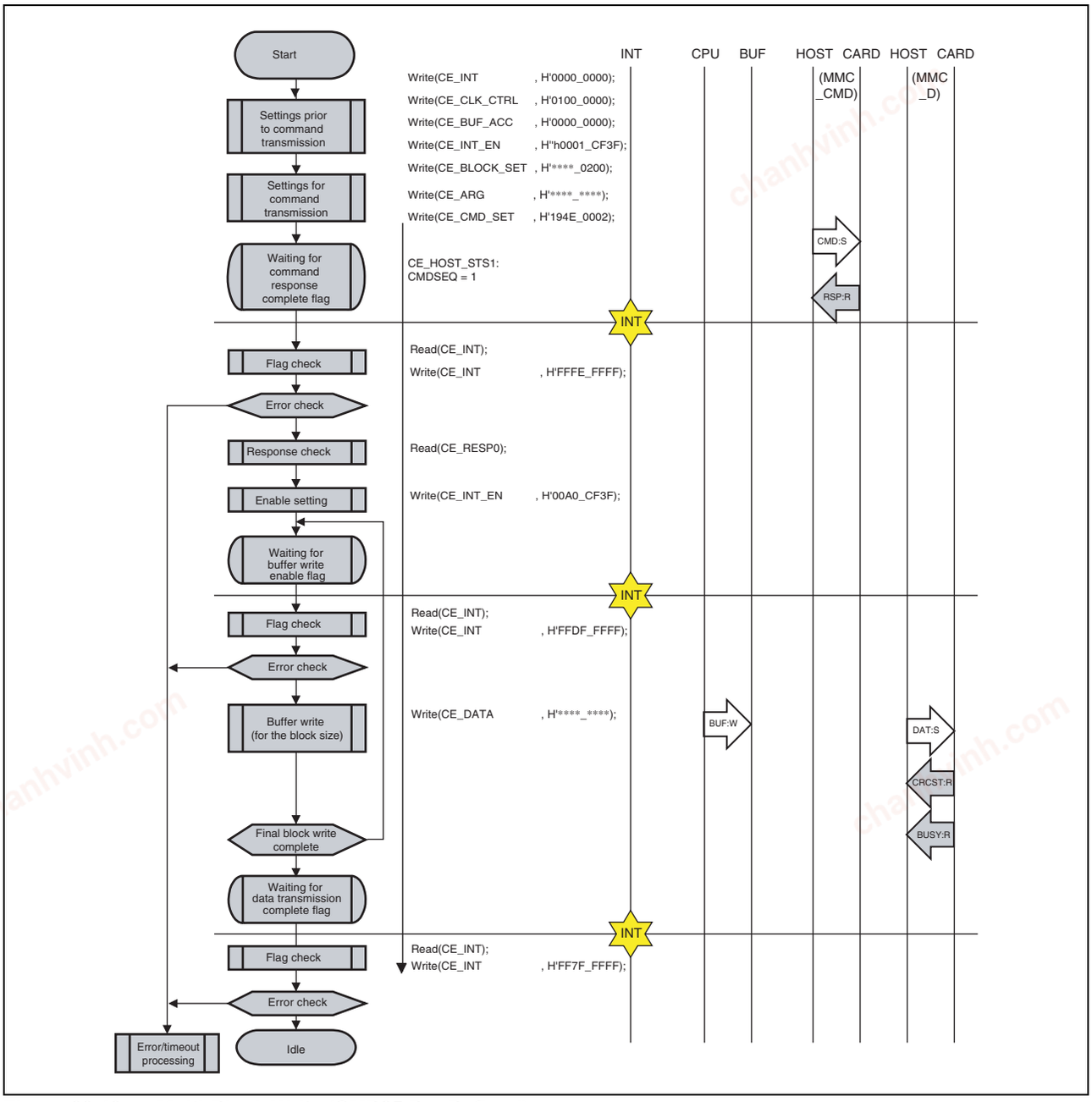


Figure 39.19 Multi-Block Write (CMD25 Pre-Defined)

chanhvinh.com

39.7.10 Multi-Block Write (with Automatic CMD12 Issuance)

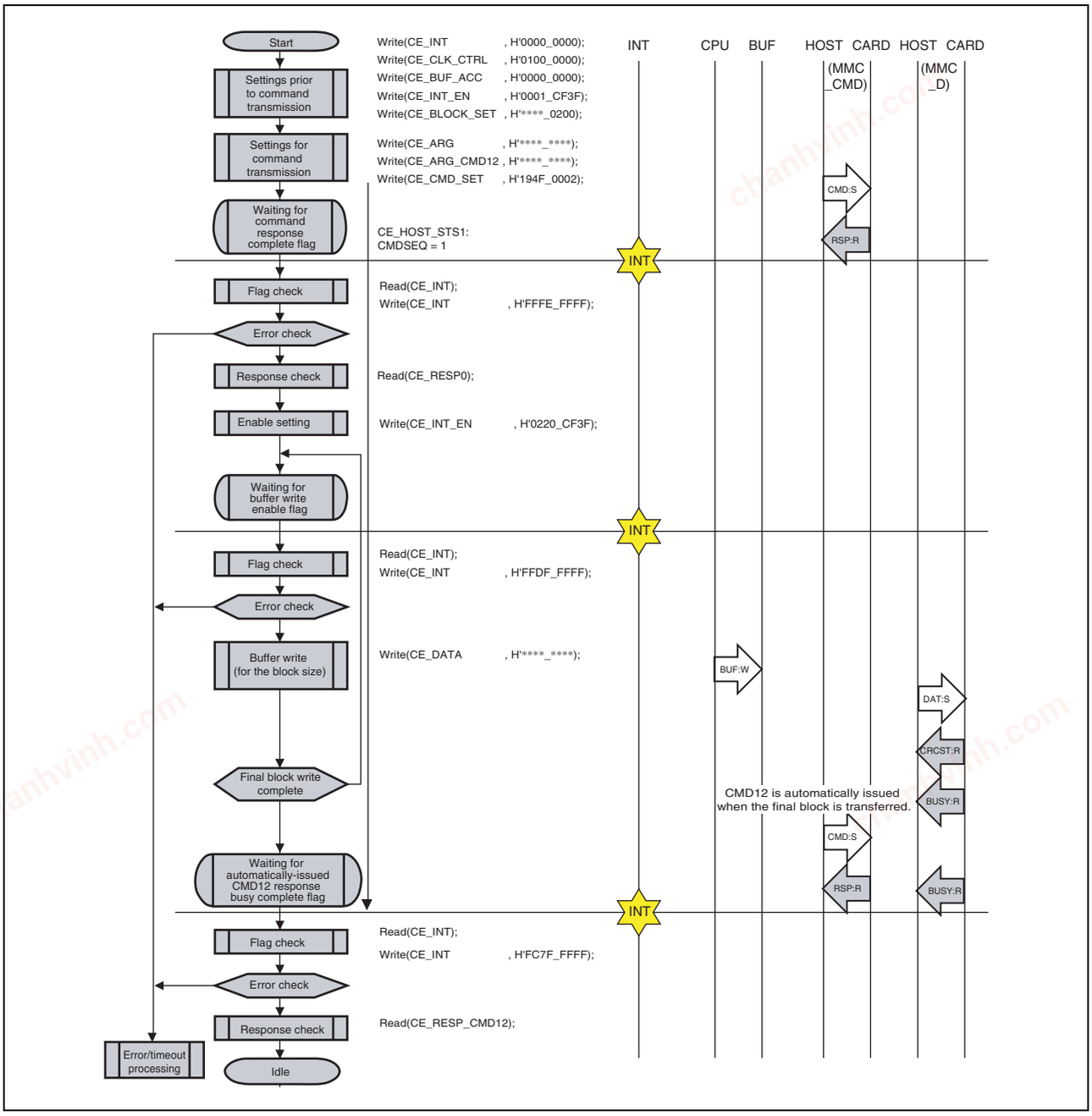


Figure 39.20 Multi-Block Write (with Automatic CMD12 Issuance) (CMD25 Open-Ended)

chanhvinh.com

39.7.11 Forcible Termination

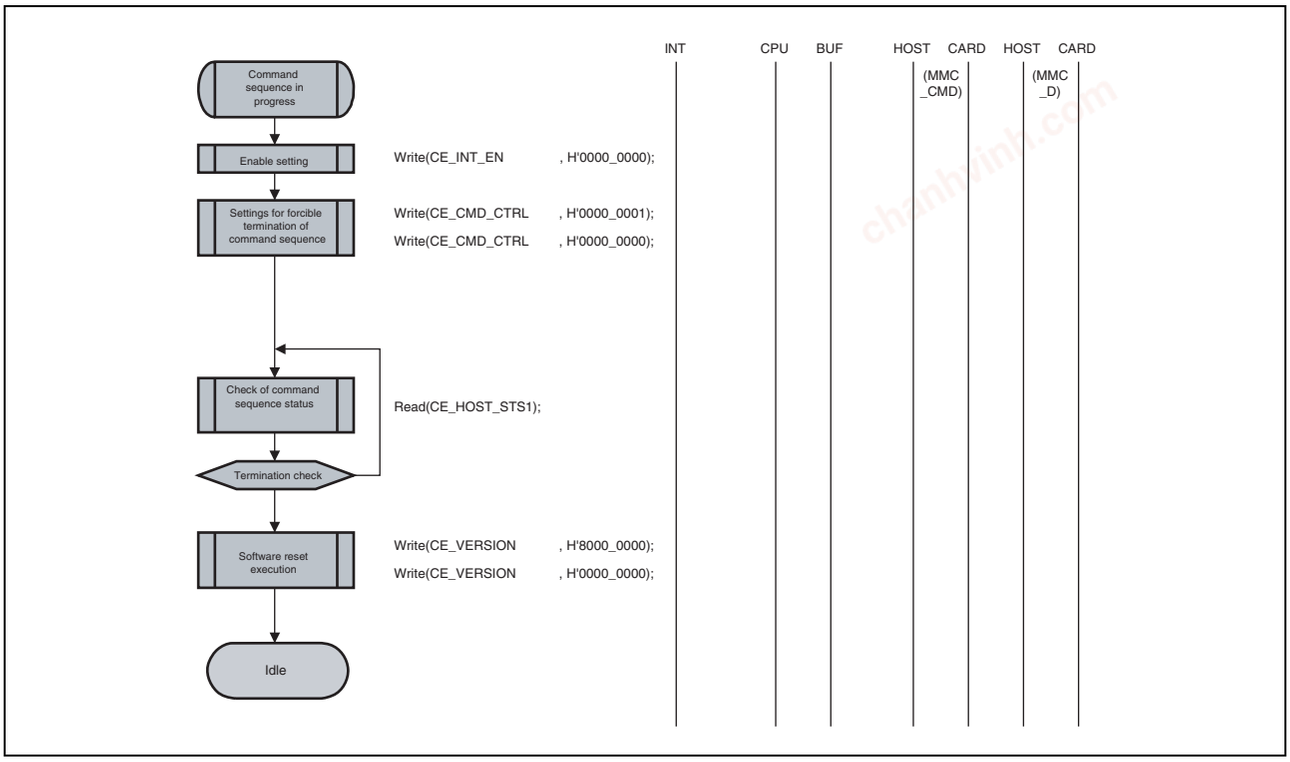


Figure 39.21 Forcible Termination

chanhvinh.com

39.7.12 Setting Values of CE_CMD_SET

Table 39.4 lists the setting values required to issue commands.

Table 39.4 Setting Values of CE_CMD_SET

Command	Response	CE_CMD_SET																						Remarks
		-	-	CMD[5:0]	RTYP[1:0]	RBSY	-	WDAT	DWEN	CMLTE	CMD12EN	RIDXC[1:0]	RCRC7C[1:0]	-	CRC16C	-	CRCSTE	TBIT	OPDM	-	-	SBIT	-	DATW[1:0]	
CMD0	-	0	0	000000	00	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD1	R3	0	0	000001	01	0	0	0	0	0	0	01	01	0	0	0	0	0	0	0	0	0	0	00	
CMD2	R2	0	0	000010	10	0	0	0	0	0	0	01	10	0	0	0	0	0	0	0	0	0	0	00	
CMD3	R1	0	0	000011	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD4	-	0	0	000100	00	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD5	R1b	0	0	000101	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD6	R1	0	0	000110	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	Background operation
	R1b	0	0	000110	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD7	R1	0	0	000111	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
	R1b	0	0	000111	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD8	R1	0	0	001000	01	0	0	1	0	0	0	00	00	0	0	0	0	0	0	0	0	0	*	**	
CMD9	R2	0	0	001001	10	0	0	0	0	0	0	01	10	0	0	0	0	0	0	0	0	0	0	00	
CMD10	R2	0	0	001010	10	0	0	0	0	0	0	01	10	0	0	0	0	0	0	0	0	0	0	00	
CMD12	R1	0	0	001100	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
	R1b	0	0	001100	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD13	R1	0	0	001101	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
	R1b	0	0	001101	01	1	1	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD14	R1	0	0	001110	01	0	0	1	0	0	0	00	00	0	1	0	0	0	0	0	0	1	0	**	
CMD15	-	0	0	001111	00	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD16	R1	0	0	010000	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD17	R1	0	0	010001	01	0	0	1	0	0	0	00	00	0	0	0	0	0	0	0	0	0	*	**	
CMD18	R1	0	0	010010	01	0	0	1	0	1	0	00	00	0	0	0	0	0	0	0	0	0	*	**	Pre-defined
	R1	0	0	010010	01	0	0	1	0	1	1	00	00	0	0	0	0	0	0	0	0	0	*	**	Open-ended
CMD19	R1	0	0	010011	01	0	0	1	1	0	0	00	00	0	0	0	1	0	0	0	0	0	0	**	
CMD23	R1	0	0	010111	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD24	R1	0	0	011000	01	0	0	1	1	0	0	00	00	0	0	0	0	0	0	0	0	0	*	**	
CMD25	R1	0	0	011001	01	0	0	1	1	1	0	00	00	0	0	0	0	0	0	0	0	0	*	**	Pre-defined
	R1	0	0	011001	01	0	0	1	1	1	1	00	00	0	0	0	0	0	0	0	0	0	*	**	Open-ended
CMD26	R1	0	0	011010	01	0	0	1	1	0	0	00	00	0	0	0	0	0	0	0	0	0	*	**	
CMD27	R1	0	0	011011	01	0	0	1	1	0	0	00	00	0	0	0	0	0	0	0	0	0	*	**	
CMD28	R1b	0	0	011100	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD29	R1b	0	0	011101	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	

chanhvinh.com

Command	Response	CE_CMD_SET																					Remarks
		-	-	CMD[5:0]	RTYP[1:0]	RBSY	-	WDAT	DWEN	CMLTE	CMD12EN	RIDXC[1:0]	RCRC7C[1:0]	-	CRC16C	-	CRCSTE	TBIT	OPDM	-	-	SBIT	-	DATW[1:0]	
CMD30	R1	0	0	011110	01	0	0	1	0	0	0	00	00	0	0	0	0	0	0	0	0	*	**	
CMD31	R1	0	0	011111	01	0	0	1	0	0	0	00	00	0	0	0	0	0	0	0	0	*	**	
CMD35	R1	0	0	100011	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	00	
CMD36	R1	0	0	100100	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	00	
CMD38	R1b	0	0	100110	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	00	
CMD39	R4	0	0	100111	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	00	
CMD40	R5	0	0	101000	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	00	Send CMD
	R5	0	0	101000	01	0	0	0	0	0	0	00	00	0	0	0	0	1	1	0	0	0	00	Send RSP
CMD42	R1	0	0	101010	01	0	0	1	1	0	0	00	00	0	0	0	0	0	0	0	0	0	**	
CMD55	R1	0	0	110111	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	00	
CMD56	R1	0	0	111000	01	0	0	1	0	0	0	00	00	0	0	0	0	0	0	0	0	*	**	Read
	R1	0	0	111000	01	0	0	1	1	0	0	00	00	0	0	0	0	0	0	0	0	*	**	Read

Note: • This module does not support CMD11 and CMD20.

chanhvinh.com

39.8 Usage Note

39.8.1 Card Detection

The CDRISE and CDFALL bits in CE_DETECT which are used for the card detection function do not have a chattering elimination function. The chattering elimination processing should be implemented by software.

39.8.2 Multi-Block Transfer

When HPI is executed, use pre-defined multi-block transfer.

39.8.3 Software Reset

For transitions to the software reset state by the SWRST bit in the CE_VERSION register, see section 42.3.6, Software Reset. However, where the procedure refers to the SRST bit, read this as the SWRST bit in the CE_VERSION register.

chanhvinh.com

40. On-Chip RAM

This LSI has an on-chip large-capacity RAM for display area and work area (128 Kbytes of this RAM are shared with the on-chip data retention RAM) and an on-chip data retention RAM, which can retain data in deep standby mode. These memory units can be used to store instructions or data.
The operation and write access to the large-capacity RAM (including on-chip data retention RAM) can be enabled or disabled through the RAM enable bits and RAM write enable bits.
The on-chip data retention RAM is assigned to page 0 in the on-chip large-capacity RAM. Retention or non-retention of data by the on-chip data retention RAM in deep standby mode is selectable on a per-page basis.

40.1 Features

- Page
  - The on-chip large-capacity RAM consists of five pages (four pages in the RZ/A1LC).
  - The on-chip data retention RAM consists of four pages. Page 0 has 16-Kbytes, page 1 has 16-Kbytes, page 2 has 32-Kbytes, and page 3 has 64-Kbytes.
- Memory map
  The on-chip RAM is located in the address spaces shown in Table 40.1 and Table 40.2. Pages 0 to 3 of the large-capacity RAM each have a capacity of 512 Kbytes, and page 4 of the large-capacity RAM has a capacity of 1 Mbyte.

Table 40.1 Address Spaces of On-Chip Large-Capacity RAM

Page	Address	Mirror Address
Page 0 (512KB)	H'20000000 to H'2007FFFF	H'60000000 to H'6007FFFF
Page 1 (512KB)	H'20080000 to H'200FFFFF	H'60080000 to H'600FFFFF
Page 2 (512KB)	H'20100000 to H'2017FFFF	H'60100000 to H'6017FFFF
Page 3 (512KB)	H'20180000 to H'201FFFFF	H'60180000 to H'601FFFFF
Page 4 (1024KB)*	H'20200000 to H'202FFFFF	H'60200000 to H'602FFFFF

Note: * Reserved in the RZ/A1LC.

Table 40.2 Address Spaces of On-Chip Data Retention RAM

Page	Address	Mirror Address
Page 0 (16KB)	H'20000000 to H'20003FFF	H'60000000 to H'60003FFF
Page 1 (16KB)	H'20004000 to H'20007FFF	H'60004000 to H'60007FFF
Page 2 (32KB)	H'20008000 to H'2000FFFF	H'60008000 to H'6000FFFF
Page 3 (64KB)	H'20010000 to H'2001FFFF	H'60010000 to H'6001FFFF

chanhvinh.com

- Ports
  Each page of the on-chip large-capacity RAM has one read and write port and is connected to the AXI bus. The on-chip RAM for data retention, which has a port independent of the port in page 0, is shared with the read and write port in four pages.
- Method of arbitration
  When the same port of the on-chip large-capacity RAM is accessed from different masters simultaneously, the AXI bus performs arbitration in round-robin mode.
- Number of access cycles
  The number of cycles for access to read or write is one cycle of Bϕ.

chanhvinh.com

40.2 Usage Notes

40.2.1 Page Conflict

When the same page of the on-chip large-capacity RAM is accessed from different masters simultaneously, a conflict on the page occurs. Although each access is completed correctly, this kind of conflict degrades the memory access speed. Therefore, it is advisable to provide software measures to prevent such conflicts as far as possible. For example, no conflict will arise if different pages are accessed by each master.

40.2.2 Data Retention

Data in the large-capacity RAM (including on-chip data retention RAM) are retained in the states other than power-on reset and deep standby mode. In power-on reset and deep standby mode, these RAMs operate as described below.

(1) Power-on Reset

(a) On-Chip Large-Capacity RAM (Excluding On-Chip Data Retention RAM)

Data are retained on a power-on reset by disabling the setting of either the VRAME or VRAMWE bit.
Data are not retained when the setting of the VRAME and VRAMWE bits are both enabled.

(b) On-Chip Data Retention RAM

Data are retained on a power-on reset by disabling the setting of any of the VRAME, VRAMWE, or RRAMWE, excluding the case that deep standby mode is canceled by power-on reset.
Data are not retained when the setting of the VRAME, VRAMWE and RRAMWE bits are all enabled.

(2) Deep Standby Mode

(a) On-Chip Large-Capacity RAM (Excluding On-Chip Data Retention RAM)

Data are not retained.

(b) On-Chip Data Retention RAM

Data are retained in deep standby mode by enabling the setting of the RRAMKP bit, excluding the case that deep standby mode is canceled by power-on reset. In the case that deep standby mode is canceled by interrupt or pins for cancelling, power-on reset exception handling is executed, but the data are retained.

chanhvinh.com

41. Ports

41.1 Features

The 176-pin package of this LSI has nine ports: JP0, P0 to P7. The 208-pin package of this LSI has eleven ports: JP0, P0 to P9. Each port pin is multiplexed with other peripheral module pin functions, and by setting the control registers, multiplexed pin functions (alternative functions) can be selectable.

41.1.1 Port group

This LSI has the following number of port groups.

Table 41.1 Port Group

Package	Number	Name
RZ/A1L 176-pin	9	P0 to P7, JP0
RZ/A1L 208-pin	11	P0 to P9, JP0

41.1.2 Port group index n

Throughout this section, the individual port groups are identified by the index "n" (n = 0 to 9), for example PMCn for the port mode control register, which is used for the Pn pin. The JTAG port has only the index 0.

41.1.3 Base address

The address of a register used to control a JTAG port is given as the base address <JPORTn_base> and the offset address from the base address. The address of a register other than the ones used to control JTAG ports is given as the base address <PORTn_base> and the offset address from the base address.

Table 41.2 Base Addresses <PORTn_base> and <JPORTn_base>

<PORTn_base> address	<JPORTn_base> address
H'FCFE 3000	H'FCFE 7B00

chanhvinh.com

41.2 Functional Overview

41.2.1 Mode of Pin Functions

Pins can operate in three modes.

- Port mode (PMCn.PMCnm bit = 0)
  A pin in port mode operates as a general purpose input/output port.
  The I/O mode is selected by setting the PMn.PMnm bit.
- Software I/O control alternative mode (PMCn.PMCnm bit = 1, PIPCn.PIPCnm bit = 0)
  In this mode, the pins operate as alternative functions.
  The I/O mode is selected by setting the PMn.PMnm bit by using software.
- Direct I/O control alternative mode (PMCn.PMCnm bit = 1, PIPCn.PIPCnm bit = 1)
  In this mode, the pins operate as alternative functions.
  Unlike the software I/O alternative mode, however, the I/O mode is selected by the alternative function.

An overview of the register settings is given in the tables below.

Table 41.3 Pin Function Configuration Overview

	Register				
Mode	**PMCn.PMCnm**	**PMn.PMnm**	**PIBCn.PIBCnm**	**PIPCn.PIPCnm**	**I/O**
Port mode	0	0	X	X	O
		1	0		—
			1		I
Software I/O control alternative mode*	1	0	X	0	O
		1			I
Direct I/O control alternative mode*		X		1	Controlled by the alternative function

Caution: When the pins are in alternative mode (PMCn.PMCnm = 1), one of the eight alternative functions is selected by the PFCn, PFCEn, and PFCAEn registers. For details, see Table 41.6, Alternative Mode Selection.

chanhvinh.com

41.2.2 Pin Data Input/Output

The registers used for data input/output are described below. The location that is read via the PPRn register differs depending on the pin mode.

- Output data
  In port mode (PMCn.PMCnm = 0), the value of the Pn.Pnm bit is output from the Pn_m pin.

- Input data
  When the PPRn register is read, either the level of the Pn_m pin, the value of the Pn.Pnm bit, or the value output by the alternative function is returned. Which value is returned depends on the pin mode and setting of several control bits. Table 41.4 shows the different PPRn read values.

Table 41.4 PPRn.PPRnm Bit Read Values

	Register				
Mode	**PMCn.PMCnm**	**PMn.PMnm**	**PIBCn.PIBCnm**	**PIPCn.PIPCnm**	**PPRn.PPRnm Bit Read Value**
Port mode	0	0	X	X	Pn.Pnm bit*
			0		Pn.Pnm bit
		1	1		Pn_m pin
Software I/O control alternative mode	1	0	X	0	Value output by the alternative function*
		1			Pn_m pin
Direct I/O control alternative mode		X		1	• Input: Pn_m pin • Output: Value output by the alternative function*

Caution: When PBDCn.PBDCnm = 1, the level of the Pn_m pin is returned as the read value of the PPRn.PPRnm bit.

chanhvinh.com

41.3 Register Description

The following registers are used for setting the individual pins of the port groups:

Table 41.5 Register Configuration

Symbol	Register Name	R/W	16 Bit Access	32 Bit Access	Address
Pn	Port register	R/W	√	—	<PORTn_base> + 0000H + n × 4
PSRn	Port set/reset register	R/W	—	√	<PORTn_base> + 0100H + n × 4
PPRn	Port pin read register	R	√	—	<PORTn_base> + 0200H + n × 4
PMn	Port mode register	R/W	√	—	<PORTn_base> + 0300H + n × 4
PMCn	Port mode control register	R/W	√	—	<PORTn_base> + 0400H + n × 4
PFCn	Port function control register	R/W	√	—	<PORTn_base> + 0500H + n × 4
PFCEn	Port function control expansion register	R/W	√	—	<PORTn_base> + 0600H + n × 4
PNOTn	Port NOT register	W	√	—	<PORTn_base> + 0700H + n × 4
PMSRn	Port mode set/reset register	R/W	—	√	<PORTn_base> + 0800H + n × 4
PMCSRn	Port mode control set/reset register	R/W	—	√	<PORTn_base> + 0900H + n × 4
PFCAEn	Port Function Control Additional Expansion register	R/W	√	—	<PORTn_base> + 0A00H + n × 4
PIBCn	Port input buffer control register	R/W	√	—	<PORTn_base> + 4000H + n × 4
PBDCn	Port bi-direction control register	R/W	√	—	<PORTn_base> + 4100H + n × 4
PIPCn	Port IP control register	R/W	√	—	<PORTn_base> + 4200H + n × 4
JPPR0	Port pin read register	R	√	—	<JPORTn_base> + 0020H
JPMC0	Port mode control register	R/W	√	—	<JPORTn_base> + 0040H
JPMCSR0	Port mode control set/reset register	R/W	—	√	<JPORTn_base> + 0090H
JPIBC0	Port input buffer control register	R/W	√	—	<JPORTn_base> + 0400H
SNCRn	Serial sound interface noise canceler control register	R/W	—	√	<PORTn_base> + 0C00H

41.3.1 Port Register (Pn)

In port output mode (PMCn.PMCnm = 0 and PMn.PMnm = 0), this register sets the level of the Pn_m pin.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Pn15	Pn14	Pn13	Pn12	Pn11	Pn10	Pn9	Pn8	Pn7	Pn6	Pn5	Pn4	Pn3	Pn2	Pn1	Pn0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	Pn[15:0]	These bits set the output level of the Pn_m pin (m = 0 to 15). 0: Outputs low level 1: Outputs high level

chanhvinh.com

41.3.2 Port Set and Reset Register (PSRn)

This register provides an alternative method for writing data to the Pn register.
The higher 16 bits of the PSRn register specify whether data can be written to the Pn.Pnm bit specified by the lower 16 bits of the PSRn register.
When reading, the higher 16 bits are read as 0000H. The lower 16 bits are read as the value of the Pn register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PSRn 31	PSRn 30	PSRn 29	PSRn 28	PSRn 27	PSRn 26	PSRn 25	PSRn 24	PSRn 23	PSRn 22	PSRn 21	PSRn 20	PSRn 19	PSRn 18	PSRn 17	PSRn 16
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSRn 15	PSRn 14	PSRn 13	PSRn 12	PSRn 11	PSRn 10	PSRn 9	PSRn 8	PSRn 7	PSRn 6	PSRn 5	PSRn 4	PSRn 3	PSRn 2	PSRn 1	PSRn 0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
31 to 16	PSRn[31:16]	These are enable bits that specify whether the value of the corresponding lower bit of the PSRn.PSRnm bit is written to the corresponding Pn.Pnm bit. 0: Pn.Pnm is independent of PSRn.PSRnm. 1: Pn.Pnm becomes the value of the PSRn.PSRnm bit. Example: If PSRn.PSRn31 = 1, the value of PSRn.PSRn15 is written to Pn.Pn15.
15 to 0	PSRn[15:0]	These are data bits that specify the value of the Pn.Pnm bit if the corresponding higher bit, PSRn.PSRn (m+16), is 1. 0: Pn.Pnm = 0 1: Pn.Pnm = 1

41.3.3 Port Pin Read Register (PPRn/JPPR0)

The PPRn register reflects the level of the Pn_m pin, the value of the Pn.Pnm bit, or the value output by the alternative function. The read value depends on register settings as described in Table 41.4, PPRn.PPRnm Bit Read Values.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PPRn 15	PPRn 14	PPRn 13	PPRn 12	PPRn 11	PPRn 10	PPRn 9	PPRn 8	PPRn 7	PPRn 6	PPRn 5	PPRn 4	PPRn 3	PPRn 2	PPRn 1	PPRn 0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit Position	Bit Name	Function
15 to 0	PPRn[15:0]	The level of the Pn_m pin, the value of the Pn.Pnm bit, or the value output by the alternative function.

Note: 1. For the PPRn register read value, see section 41.2.2, Pin Data Input/Output.
2. The JPPR0[1:0] bits are control bits in the JTAG port pin read register (JPPR0).

chanhvinh.com

41.3.4 Port Mode Register (PMn)

This register specifies whether the Pn_m pin is in input mode or in output mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMn 15	PMn 14	PMn 13	PMn 12	PMn 11	PMn 10	PMn 9	PMn 8	PMn 7	PMn 6	PMn 5	PMn 4	PMn 3	PMn 2	PMn 1	PMn 0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	PMn[15:0]	Specifies input/output mode of the corresponding pin: 0: Output mode (output enabled) 1: Input mode (output disabled)

Caution: To use the Pn_m pin in port input mode (PMCn.PMCnm = 0 and PMn.PMnm = 1), the PIBCn.PIBCnm bit must be set to 1 so that the input buffer is enabled.

Note: After the register is reset (PIPCn.PIPCnm bit is 0), the PMn.PMnm bit specifies the I/O direction in port mode (PMCn.PMCnm = 0) and alternative mode (PMCn.PMCnm = 1). For the pins that are automatically configured in boot mode (see Table 8.3, Initial States by Areas in Boot Modes 0 and 1 to 3), the alternative function (not the PMn register) is used to control the I/O direction.

41.3.5 Port Mode Control Register (PMCn/JPMC0)

This register specifies whether the Pn_m pin is in port mode or in alternative mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMCn 15	PMCn 14	PMCn 13	PMCn 12	PMCn 11	PMCn 10	PMCn 9	PMCn 8	PMCn 7	PMCn 6	PMCn 5	PMCn 4	PMCn 3	PMCn 2	PMCn 1	PMCn 0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	PMCn[15:0]	Specifies the operation mode of the corresponding pin: 0: Port mode 1: Alternative mode

Caution: The input/output control is not performed just by setting alternative mode (PMCn.PMCnm). Set 1 in the PIPCn.PIPCnm bit too when the I/O control is to be performed by using the alternative function.

Note: The JPMC0[1:0] bits are control bits in the JTAG port mode control register (JPMC0).

chanhvinh.com

41.3.6 Port Function Control Register (PFCn)

This register, together with the PFCEn and PFCAEn register, specifies the alternative function of the pins.
The I/O direction of several alternative functions can be controlled directly by using the Pn_m pin. For these alternative functions, the PIPCn.PIPCnm bit must be set to “1”.
For all other alternative functions, the I/O direction is specified by using the PMn.PMnm bit.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFCn 15	PFCn 14	PFCn 13	PFCn 12	PFCn 11	PFCn 10	PFCn 9	PFCn 8	PFCn 7	PFCn 6	PFCn 5	PFCn 4	PFCn 3	PFCn 2	PFCn 1	PFCn 0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	PFCn[15:0]	Specifies the alternative function of a pin.

Table 41.6 Alternative Mode Selection

PMCn. PMCnm	PFCAEn. PFCAEnm	PFCEn. PFCEnm	PFCn. PFCnm	PMn. PMnm	Function*2
0	X	X	X	1	Port mode/input
				0	Port mode/output
1*1	0	0	0	1	Alternative mode/1st alternative function/input
				0	Alternative mode/1st alternative function/output
			1	1	Alternative mode/2nd alternative function/input
				0	Alternative mode/2nd alternative function/output
		1	0	1	Alternative mode/3rd alternative function/input
				0	Alternative mode/3rd alternative function/output
			1	1	Alternative mode/4th alternative function/input
				0	Alternative mode/4th alternative function/output
	1	0	0	1	Alternative mode/5th alternative function/input
				0	Alternative mode/5th alternative function/output
			1	1	Alternative mode/6th alternative function/input
				0	Alternative mode/6th alternative function/output
		1	0	1	Alternative mode/7th alternative function/input
				0	Alternative mode/7th alternative function/output
			1	1	Alternative mode/8th alternative function/input
				0	Alternative mode/8th alternative function/output

Cautions:1. When the PIPCn.PIPCnm bit is set to 1, the I/O direction is directly controlled by the alternative function and the setting of the PMn.PMnm bit becomes invalid.

2. Use alternative functions that are open to the public. If a mode to which no alternative function is assigned is selected, correct operation is not guaranteed.

chanhvinh.com

41.3.7 Port Function Control Expansion Register (PFCEn)

This register, together with the PFCn and PFCAEn register, specifies the alternative function of the pins.
The I/O direction of several alternative functions can be controlled directly by using the Pn_m pin. For these alternative functions, the PIPCn.PIPCnm bit must be set to “1”.
For all other alternative functions, the I/O direction is specified by using the PMn.PMnm bit.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFCEn 15	PFCEn 14	PFCEn 13	PFCEn 12	PFCEn 11	PFCEn 10	PFCEn 9	PFCEn 8	PFCEn 7	PFCEn 6	PFCEn 5	PFCEn 4	PFCEn 3	PFCEn 2	PFCEn 1	PFCEn 0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	PFCEn[15:0]	Specifies the alternative function of a pin.

41.3.8 Port NOT Register (PNOTn)

This register allows the Pn.Pnm bit to be inverted without directly writing to the Pn register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PNOTn 15	PNOTn 14	PNOTn 13	PNOTn 12	PNOTn 11	PNOTn 10	PNOTn 9	PNOTn 8	PNOTn 7	PNOTn 6	PNOTn 5	PNOTn 4	PNOTn 3	PNOTn 2	PNOTn 1	PNOTn 0
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit Position	Bit Name	Function
15 to 0	PNOTn[15:0]	Specifies whether to invert Pn.Pnm. 0: Pn.Pnm is not inverted (Pnm → Pnm) 1: Pn.Pnm is inverted ($\overline{\text{Pnm}}$ → Pnm)

chanhvinh.com

41.3.9 Port Mode Set and Reset Register (PMSRn)

This register provides an alternative method for writing data to the PMn register.
The higher 16 bits of the PMSRn register specify whether data can be written to the PMn.PMnm bit specified by the lower 16 bits of the PMSRn register.
When reading, the higher 16 bits are read as 0000H. The lower 16 bits are read as the value of the PMn register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PMSRn 31	PMSRn 30	PMSRn 29	PMSRn 28	PMSRn 27	PMSRn 26	PMSRn 25	PMSRn 24	PMSRn 23	PMSRn 22	PMSRn 21	PMSRn 20	PMSRn 19	PMSRn 18	PMSRn 17	PMSRn 16
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMSRn 15	PMSRn 14	PMSRn 13	PMSRn 12	PMSRn 11	PMSRn 10	PMSRn 9	PMSRn 8	PMSRn 7	PMSRn 6	PMSRn 5	PMSRn 4	PMSRn 3	PMSRn 2	PMSRn 1	PMSRn 0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
31 to 16	PMSRn[31:16]	These are enable bits that specify whether the value of the corresponding lower bit of the PMSRn.PMSRnm bit is written to the corresponding PMn.PMnm bit. 0: PMn.PMnm does not depend on the PMSRn.PMSRnm bit. 1: PMn.PMnm becomes the value of the PMSRn.PMSRnm bit. Example: If PMSRn.PMSRn31 = 1, the value of PMSRn.PMSRn15 is written to PMn.PMn15.
15 to 0	PMSRn[15:0]	These are data bits that specify the value of the PMn.PMnm bit if the corresponding higher bit, PMSRn.PMSRn (m + 16), is 1. 0: PMn.PMnm = 0 1: PMn.PMnm = 1

chanhvinh.com

41.3.10 Port Mode Control Set and Reset Register (PMCSRn/JPMCSR0)

This register provides an alternative method for writing data to the PMCn register.

The higher bits of the PMCSRn register specify whether data can be written to the PMCn.PMCnm bit specified by the lower bits of the PMCSRn register.

When reading, the higher 16 bits are read as 0000H. The lower 16 bits are read as the value of the PMCn register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PMCSRn31	PMCSRn30	PMCSRn29	PMCSRn28	PMCSRn27	PMCSRn26	PMCSRn25	PMCSRn24	PMCSRn23	PMCSRn22	PMCSRn21	PMCSRn20	PMCSRn19	PMCSRn18	PMCSRn17	PMCSRn16
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMCSRn15	PMCSRn14	PMCSRn13	PMCSRn12	PMCSRn11	PMCSRn10	PMCSRn9	PMCSRn8	PMCSRn7	PMCSRn6	PMCSRn5	PMCSRn4	PMCSRn3	PMCSRn2	PMCSRn1	PMCSRn0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
31 to 16	PMCSRn[31:16]	These are enable bits that specify whether the value of the corresponding lower bit of the PMCSRn.PMCSRnm bit is written to the corresponding PMCn.PMCnm bit. 0: PMCn.PMCnm does not depend on the PMCSRn.PMCSRnm bit. 1: PMCn.PMCnm becomes the value of the PMCSRn.PMCSRnm bit. Example: If PMCSRn.PMCSRn31 = 1, the value of PMCSRn.PMCSRn15 is written to PMCn.PMCn15.
15 to 0	PMCSRn[15:0]	These are data bits that specify the value of the PMCn.PMCnm bit if the corresponding higher bit, PMCSRn.PMCSRn (m+16), is 1. 0: PMCn.PMCnm = 0 1: PMCn.PMCnm = 1

Note: The JPMCSR0[1:0] bits are control bits in the JTAG port mode control set and reset register (JPMCSR0).

41.3.11 Port Function Control Additional Expansion Register (PFCAEn)

This register, together with the PFCn and PFCEn register, specifies the alternative function of the pins.

The I/O direction of several alternative functions can be controlled directly by using the Pn_m pin. For these alternative functions, the PIPCn.PIPCnm bit must be set to “1”.

For all other alternative functions, the I/O direction is specified by using the PMn.PMnm bit.

Use the PFCn.PFCnm, PFCEn.PFCEnm, and PFCAEn.PFCAEnm bits to select the alternative functions, and then set the PMCn.PMCnm bit to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFCAEn15	PFCAEn14	PFCAEn13	PFCAEn12	PFCAEn11	PFCAEn10	PFCAEn9	PFCAEn8	PFCAEn7	PFCAEn6	PFCAEn5	PFCAEn4	PFCAEn3	PFCAEn2	PFCAEn1	PFCAEn0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	PFCAEn[15:0]	Specifies the alternative function of a pin.

Caution: Although some alternative functions are assigned to multiple pins in this LSI, only one of the pins can be used for each alternative function. Do not set the same alternative function in multiple pins. For example, when the a/b/c pin is used as the b pin, the b/d/e pin cannot be used as the b pin. Before using the b/d/e pin, set the pin as an alternative function other than b.

Note: For the allocation of the alternative functions, see the table of the pin functions of each port.

chanhvinh.com

41.3.12 Port Input Buffer Control Register (PIBCn/JPIBC0)

This register specifies whether to enable or disable the Pn_m pin's input buffer in port input mode (PMCn.PMCnm = 0 and PMn.PMnm = 1 or JPMC0.JPMC0 = 0).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIBCn 15	PIBCn 14	PIBCn 13	PIBCn 12	PIBCn 11	PIBCn 10	PIBCn 9	PIBCn 8	PIBCn 7	PIBCn 6	PIBCn 5	PIBCn 4	PIBCn 3	PIBCn 2	PIBCn 1	PIBCn 0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	PIBCn[15:0]	These bits enable or disable the input buffer. 0: Input buffer disabled 1: Input buffer enabled

Note: When the input buffer is disabled, through current does not flow even when the pin level is Hi-Z. Thus the pin does not need to be fixed to a high or low level externally (except for P2_0, P2_2, P2_7, P2_9, P5_8, P5_9, P5_10, P6_7, P7_2, P7_3, P7_6, and P7_9).
The JPIBC0[1:0] bits are control bits in the JTAG port input buffer control register (JPIBC0).

Caution: The input buffer is enabled regardless of the settings in this register when the output buffer is enabled and PBDCn.PBDCnm = 1.

41.3.13 Port Bidirection Control Register (PBDCn)

This register specifies whether to enable or disable the input buffers when the output buffers are enabled. If the PBDCn.PBDCnm bit is set to "1" when the output buffers are enabled, bidirectional mode is enabled, and the level of the Pn_m pin is always read via the PPRn.PPRnm bit.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PBDCn 15	PBDCn 14	PBDCn 13	PBDCn 12	PBDCn 11	PBDCn 10	PBDCn 9	PBDCn 8	PBDCn 7	PBDCn 6	PBDCn 5	PBDCn 4	PBDCn 3	PBDCn 2	PBDCn 1	PBDCn 0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	PBDCn[15:0]	Enables/disables bidirectional mode of the corresponding pin: 0: Bidirectional mode disabled 1: Bidirectional mode enabled

Caution: When using the Pn_m pin as an alternative function (PMCn.PMCnm = 1 and PMn.PMnm = 0), the level of the Pn_m pin can be read at the PPRn.PPRnm bit by setting PBDCn.PBDCnm to 1. Note, however, that in this case, the level of the Pn_m pin will be input to the alternative function that the Pn_m pin is being used as.

chanhvinh.com

41.3.14 Port IP Control Register (PIPCn)

This register specifies whether the I/O direction of the Pn_m pin is controlled by the PMn.PMnm bit or by the alternative function in alternative mode (PMCn.PMCnm = 1). Set the PIPCn.PIPCnm bit to 0 to use the alternative functions listed in Table 41.7. Set the PIPCn.PIPCnm bit to 1 to use the alternative functions not listed in Table 41.7. If the PIPCn.PIPCnm bit is set to 1, the I/O direction is controlled by the alternative function, and the setting of the PMn.PMnm bit becomes invalid.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIPCn 15	PIPCn 14	PIPCn 13	PIPCn 12	PIPCn 11	PIPCn 10	PIPCn 9	PIPCn 8	PIPCn 7	PIPCn 6	PIPCn 5	PIPCn 4	PIPCn 3	PIPCn 2	PIPCn 1	PIPCn 0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
15 to 0	PIPCn[15:0]	These bits specify how the I/O direction is controlled in alternative mode. 0: The I/O direction is controlled by the PMn.PMnm bit (software I/O control). 1: The I/O direction is controlled by the alternative function (direct I/O control).

Table 41.7 Alternative Functions for Which the PIPCn.PIPCnm Bit Should be Set to 0

Function	Pin Name	Remark
Multi-function timer pulse unit 2	TIOC0A	Set the PIPCn.PIPCnm bit to 0, and specify input or output by using the PMn.PMnm bit.
	TIOC0B	
	TIOC0C	
	TIOC0D	
	TIOC1A	
	TIOC1B	
	TIOC2A	
	TIOC2B	
	TIOC3A	
	TIOC3B	
	TIOC3C	
	TIOC3D	
	TIOC4A	
	TIOC4B	
	TIOC4C	
	TIOC4D	
Serial sound interface	SSITxD0	Set the PIPCn.PIPCnm bit and the PMn.PMnm bit to 0.
	SSITxD1	
	SSITxD3	
Watchdog timer	$\overline{\text{WDTOVF}}$	Set the PIPCn.PIPCnm bit and the PMn.PMnm bit to 0.

chanhvinh.com

41.3.15 Serial Sound Interface Noise Canceler Control Register (SNCR)

This register controls the noise canceler in the input route from the LSI pin to a serial sound interface and the pins for MII supported in the Ethernet. Four lower-order bits can be set only when slave mode is selected for the corresponding channel of the serial sound interface.
In master mode, the bit corresponding to the channel of the serial sound interface must be set to 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	ETSEL*	—	—	SSI3NCE	SSI2NCE	SSI1NCE	SSI0NCE
R/W:	R	R	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W

Bit Position	Bit Name	Function
6	ETSEL*	Enables or disables the Ethernet controller and EthernetAVB with respect to the pins for MII supported in the Ethernet. 0: The Ethernet controller is enabled and the EthernetAVB is disabled. 1: The EthernetAVB is enabled and the Ethernet controller is disabled.
5, 4	—	Reserved The write value should always be 0.
3	SSI3NCE	Serial Sound Interface Channel 3 Noise Canceler Enable Enables or disables the noise canceler of SSISCK3, SSIWS3, and SSIRxD3. 0: Noise canceler is disabled. 1: Noise canceler is enabled.
2	SSI2NCE	Serial Sound Interface Channel 2 Noise Canceler Enable Enables or disables the noise canceler of SSISCK2, SSIWS2, and SSIDATA2. 0: Noise canceler is disabled. 1: Noise canceler is enabled.
1	SSI1NCE	Serial Sound Interface Channel 1 Noise Canceler Enable Enables or disables the noise canceler of SSISCK1, SSIWS1, and SSIRxD1. 0: Noise canceler is disabled. 1: Noise canceler is enabled.
0	SSI0NCE	Serial Sound Interface Channel 0 Noise Canceler Enable Enables or disables the noise canceler of SSISCK0, SSIWS0, and SSIRxD0. 0: Noise canceler is disabled. 1: Noise canceler is enabled.

Note: * This bit is only present in the RZ/A1LU. For the RZ/A1L and RZ/A1LC, the write value should always be 0.

chanhvinh.com

41.4 Port Function

Table 41.8 Number and Function of Ports

Port	Pin Name	Size (Number of Ports)	Input/Output	Note
JTAG Port 0	JP0_0, JP0_1	2 bits	input	
Port 0	P0_0 to P0_3	4 bits	input	
Port 1	P1_0 to P1_15	16 bits	input/open-drain output	Output of only the P1_0 to P1_7 pins is open-drain output.
Port 2	P2_0 to P2_9	10 bits	input/output	
Port 3	P3_0 to P3_15	16 bits	input/output	
Port 4	P4_0 to P4_7	8 bits	input/output	
Port 5	P5_0 to P5_15	16 bits	input/output	
Port 6	P6_0 to P6_15	16 bits	input/output	
Port 7	P7_0 to P7_11	12 bits	input/output	
Port 8	P8_0 to P8_15	16 bits	input/output	208-pin package only
Port 9	P9_0 to P9_5	6 bits	input/output	208-pin package only

41.5 JTAG Port 0 (JP0)

Table 41.9 Pin Function (JP0)

Port Mode		Alternative Mode	
Input	Output	Input	Output
JP0_0		TDI	
JP0_1			TDO

Caution: Only input in port mode.

Table 41.10 Control Registers (JP0)

Register	Register Size	Valid Bit		Offset Address <JPORTn_base>	Initial Value
		Location	R/W		
JPPR0	16	1 and 0	R	H'0020	H'0000
JPMC0	16	1 and 0	R/W	H'0040	H'FFFF
JPMCSR0	32	17 and 16, 1 and 0	W, R/W	H'0090	H'0000_FFFF
JPIBC0	16	1 and 0	R/W	H'0400	H'0000

Caution: An initial value is read if the bit is invalid. The write value should always be the initial value.

chanhvinh.com

41.6 Port 0 (P0)

Table 41.11 Pin Function (P0)

Port Mode		Alternative Mode							
		1st Alternative		2nd Alternative		3rd Alternative		4th Alternative	
Input	Output	Input	Output	Input	Output	Input	Output	Input	Output
P0_0				RxD0		IRQ4			
P0_1				RxD2		SSIRxD3		$\overline{\text{ADTRG}}$	
P0_2				RxD1		IRQ7			
P0_3				RxD3		SPDIF_IN			

Cautions: 1. Only input in port mode.

2. P0_0, P0_1, P0_2, P0_3 pins are MD_BOOT0, MD_BOOT1, MD_CLK, MD_CLKS functions in the state of $\overline{\text{RES}}$ = L.

Table 41.12 Control Registers (P0)

Register	Register Size	Valid Bit Location	Valid Bit R/W	Offset Address <PORTn_base>	Initial Value
PPR0	16	3 to 0	R	H'0200	H'0000
PMC0	16	3 to 0	R/W	H'0400	H'0000
PFC0	16	3 to 0	R/W	H'0500	H'0000
PFCE0	16	3 to 0	R/W	H'0600	H'0000
PMCSR0	32	19 to 16, 3 to 0	W, R/W	H'0900	H'0000_0000
PFCAE0	16	3 to 0	R/W	H'0A00	H'0000
PIBC0	16	3 to 0	R/W	H'4000	H'0000

Caution: An initial value is read if the bit is invalid. The write value should always be the initial value.

chanhvinh.com

41.7 Port 1 (P1)

Table 41.13 Pin Function (P1)

Port Mode		Alternative Mode							
		1st Alternative		2nd Alternative		3rd Alternative		4th Alternative	
Input	Output	Input	Output	Input	Output	Input	Output	Input	Output
P1_0	P1_0	RIIC0SCL		IRQ4		ET_RXD0		DV0_DATA0	
P1_1	P1_1	RIIC0SDA		IRQ5		ET_RXD1		DV0_DATA1	
P1_2	P1_2	RIIC1SCL		IRQ6		ET_RXD2		DV0_DATA2	
P1_3	P1_3	RIIC1SDA		IRQ7		ET_RXD3		DV0_DATA3	
P1_4	P1_4	RIIC2SCL		IRQ0		DREQ0		VIO_D0	
P1_5	P1_5	RIIC2SDA		IRQ1				VIO_D1	
P1_6	P1_6	RIIC3SCL		IRQ2		SSIRxD0		VIO_D2	
P1_7	P1_7	RIIC3SDA		IRQ3		RxD2		VIO_D3	
P1_8		AN0		IRQ0		RxD0		DV0_DATA4	
P1_9		AN1		IRQ1		RxD1		DV0_DATA5	
P1_10		AN2		IRQ2		RxD2		DV0_DATA6	
P1_11		AN3		IRQ3		RxD3		DV0_DATA7	
P1_12		AN4		IRQ4		ET_RXD0		VIO_D4	
P1_13		AN5		IRQ5		ET_RXD1		VIO_D5	
P1_14		AN6		IRQ6		ET_RXD2		VIO_D6	
P1_15		AN7		IRQ7		ET_RXD3		VIO_D7	

Caution: P1_0 to P1_7 pins are input with open-drain output. P1_8 to P1_15 pins are only input.

Table 41.14 Control Registers (P1)

Register	Register Size	Valid Bit		Offset Address <PORTn_base>	Initial Value
		Location	R/W		
P1	16	7 to 0	R/W	H'0004	H'0000
PSR1	32	23 to 16, 7 to 0	W, R/W	H'0104	H'0000_0000
PPR1	16	15 to 0	R	H'0204	H'0000
PM1	16	7 to 0	R/W	H'0304	H'FFFF*2
PMC1	16	15 to 0	R/W	H'0404	H'0000
PFC1	16	15 to 0	R/W	H'0504	H'0000
PFCE1	16	15 to 0	R/W	H'0604	H'0000
PNOT1	16	7 to 0	W	H'0704	H'0000
PMSR1	32	23 to 16, 7 to 0	W, R/W	H'0804	H'0000_FFFF
PMCSR1	32	23 to 16, 7 to 0	W, R/W	H'0904	H'0000_0000
PFCAE1	16	15 to 0	R/W	H'0A04	H'0000
PIBC1	16	15 to 0	R/W	H'4004	H'0000
PBDC1	16	15 to 0	R/W	H'4104	H'0000
PIPC1	16	7 to 0	R/W	H'4204	H'FF00*2

Cautions: 1. An initial value is read if the bit is invalid. The write value should always be the initial value.

2. The values of bits 8 to 15 of the PM1 and PIPC1 registers are fixed. When read, all of these bits are always read as 1.

chanhvinh.com

41.8 Port 2 (P2)

Table 41.15 Pin Function (P2)

Port Mode		Alternative Mode									
		1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative	
Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output
P2_0	P2_0		$\overline{CS3}$	RLIN30R*		SPDIF_IN		IRQ7			
P2_1	P2_1		$\overline{RAS}$		RLIN30TX*		SPDIF_OUT	IRQ6			
P2_2	P2_2		$\overline{CAS}$	CAN1RX		TIOC0C		IRQ5			
P2_3	P2_3		CKE		CAN1TX	TIOC0D					
P2_4	P2_4		$\overline{WE0}$/ DQMLL			TIOC4A					
P2_5	P2_5		$\overline{WE1}$/$\overline{WE}$/ DQMLU			TIOC3A					
P2_6	P2_6		RD/$\overline{WR}$	SSIRxD3		TIOC2A					
P2_7	P2_7		$\overline{CS0}$	SSISCK3		TIOC1A		IRQ2			
P2_8	P2_8		$\overline{RD}$		SSITxD3	TIOC0A					CAN0TX
P2_9	P2_9		A0	SSIWS3		SCK0		IRQ1		CAN0RX	

Note: * RZ/A1L only

Table 41.16 Control Registers (P2)

Register	Register Size	Valid Bit		Offset Address <PORTn_base>	Initial Value
		Location	R/W		
P2	16	9 to 0	R/W	H'0008	H'0000
PSR2	32	25 to 16, 9 to 0	W, R/W	H'0108	H'0000_0000
PPR2	16	9 to 0	R	H'0208	H'0000
PM2	16	9 to 0	R/W	H'0308	H'FFFF
PMC2	16	9 to 0	R/W	H'0408	H'01C0 (In boot mode 0), H'0000 (Other than in boot mode 0)
PFC2	16	9 to 0	R/W	H'0508	H'0000
PFCE2	16	9 to 0	R/W	H'0608	H'0000
PNOT2	16	9 to 0	W	H'0708	H'0000
PMSR2	32	25 to 16, 9 to 0	W, R/W	H'0808	H'0000_FFFF
PMCSR2	32	25 to 16, 9 to 0	W, R/W	H'0908	H'0000_01C0 (In boot mode 0), H'0000_0000 (Other than in boot mode 0)
PFCAE2	16	9 to 0	R/W	H'0A08	H'0000
PIBC2	16	9 to 0	R/W	H'4008	H'0000
PBDC2	16	9 to 0	R/W	H'4108	H'01C0 (In boot mode 0), H'0000 (Other than in boot mode 0)
PIPC2	16	9 to 0	R/W	H'4208	H'01C0 (In boot mode 0), H'0000 (Other than in boot mode 0)

Cautions: 1. An initial value is read if the bit is invalid. The write value should always be the initial value.

2. For the pins that are automatically configured in boot mode, see Table 8.3, Initial States by Areas in Boot Modes 0 and 1 to 3.

chanhvinh.com

41.9 Port 3 (P3)

Table 41.17 Pin Function (P3)

Port Mode		Alternative Mode											
		1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative	
Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output
P3_0	P3_0		A1	SD_D2_0			LCD0_DATA0	ET_TXCLK					
P3_1	P3_1		A2	SD_D3_0			LCD0_DATA1		ET_TXER				
P3_2	P3_2		A3	SD_CMD_0			LCD0_DATA2		ET_TXEN				
P3_3	P3_3		A4		SD_CLK_0		LCD0_DATA3	ET_RXCLK					
P3_4	P3_4		A5	SD_D0_0			LCD0_DATA4	ET_RXER					
P3_5	P3_5		A6	SD_D1_0			LCD0_DATA5	ET_RXDV					
P3_6	P3_6		A7	SD_WP_0			LCD0_DATA6	ET_COL					
P3_7	P3_7		A8	SD_CD_0			LCD0_DATA7	ET_CRS					
P3_8	P3_8		A9			AUDIO_CLK		DV0_DATA8		SCK3			
P3_9	P3_9		A10				SPDIF_OUT	DV0_DATA9			TxD3		
P3_10	P3_10		A11	SPBIO01_0		TIOC3B		DV0_DATA10		RxD3			
P3_11	P3_11		A12	SPBIO11_0		TIOC3A		DV0_DATA11					
P3_12	P3_12		A13	SPBIO21_0		TIOC3C		DV0_DATA12					
P3_13	P3_13		A14	SPBIO31_0		TIOC3D		DV0_DATA13					
P3_14	P3_14		A15	VIO_CLK		SPDIF_IN		DV0_DATA14		SCK1			AUDIO_XOUT2
P3_15	P3_15		A16	VIO_FLD				DV0_DATA15			TxD1		

Table 41.18 Control Registers (P3)

Register	Register Size	Valid Bit Location	Valid Bit R/W	Offset Address <PORTn_base>	Initial Value
P3	16	15 to 0	R/W	H'000C	H'0000
PSR3	32	31 to 16, 15 to 0	W, R/W	H'010C	H'0000_0000
PPR3	16	15 to 0	R	H'020C	H'0000
PM3	16	15 to 0	R/W	H'030C	H'FFFF
PMC3	16	15 to 0	R/W	H'040C	H'FFFF (In boot mode 0), H'0000 (Other than in boot mode 0)
PFC3	16	15 to 0	R/W	H'050C	H'0000
PFCE3	16	15 to 0	R/W	H'060C	H'0000
PNOT3	16	15 to 0	W	H'070C	H'0000
PMSR3	32	31 to 16, 15 to 0	W, R/W	H'080C	H'0000_FFFF
PMCSR3	32	31 to 16, 15 to 0	W, R/W	H'090C	H'0000_FFFF (In boot mode 0), H'0000_0000 (Other than in boot mode 0)
PFCAE3	16	15 to 0	R/W	H'0A0C	H'0000
PIBC3	16	15 to 0	R/W	H'400C	H'0000
PBDC3	16	15 to 0	R/W	H'410C	H'FFFF (In boot mode 0), H'0000 (Other than in boot mode 0)
PIPC3	16	15 to 0	R/W	H'420C	H'FFFF (In boot mode 0), H'0000 (Other than in boot mode 0)

Caution: For the pins that are automatically configured in boot mode, see Table 8.3, Initial States by Areas in Boot Modes 0 and 1 to 3.

chanhvinh.com

41.10 Port 4 (P4)

Table 41.19 Pin Function (P4)

Port Mode		Alternative Mode									
		1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative	
Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output
P4_0	P4_0		A17	VIO_VD		TIOC1B			ET_MDC	$\overline{\text{CTS1}}$	
P4_1	P4_1		A18	VIO_HD		TIOC2B		ET_MDIO		$\overline{\text{RTS1}}$	
P4_2	P4_2		A19	SPBIO20_0			TRACEDATA2*				
P4_3	P4_3		A20	SPBIO30_0			TRACEDATA3*				
P4_4	P4_4		A21		SPBCLK_0		TRACECLK*				
P4_5	P4_5		A22		SPBSSL_0		TRACECTL*				
P4_6	P4_6		A23	SPBIO00_0			TRACEDATA0*				
P4_7	P4_7		A24	SPBIO10_0			TRACEDATA1*				

Note: * RZ/A1L only

Table 41.20 Control Registers (P4)

Register	Register Size	Valid Bit Location	Valid Bit R/W	Offset Address <PORTn_base>	Initial Value
P4	16	7 to 0	R/W	H'0010	H'0000
PSR4	32	23 to 16, 7 to 0	W, R/W	H'0110	H'0000_0000
PPR4	16	7 to 0	R	H'0210	H'0000
PM4	16	7 to 0	R/W	H'0310	H'FFFF
PMC4	16	7 to 0	R/W	H'0410	H'000F (In boot mode 0), H'0000 (Other than in boot mode 0)
PFC4	16	7 to 0	R/W	H'0510	H'0000
PFCE4	16	7 to 0	R/W	H'0610	H'0000
PNOT4	16	7 to 0	W	H'0710	H'0000
PMSR4	32	23 to 16, 7 to 0	W, R/W	H'0810	H'0000_FFFF
PMCSR4	32	23 to 16, 7 to 0	W, R/W	H'0910	H'0000_000F (In boot mode 0), H'0000_0000 (Other than in boot mode 0)
PFCAE4	16	7 to 0	R/W	H'0A10	H'0000
PIBC4	16	7 to 0	R/W	H'4010	H'0000
PBDC4	16	7 to 0	R/W	H'4110	H'000F (In boot mode 0), H'0000 (Other than in boot mode 0)
PIPC4	16	7 to 0	R/W	H'4210	H'000F (In boot mode 0), H'0000 (Other than in boot mode 0)

Cautions: 1. An initial value is read if the bit is invalid. The write value should always be the initial value.

2. For the pins that are automatically configured in boot mode, see Table 8.3, Initial States by Areas in Boot Modes 0 and 1 to 3.

chanhvinh.com

41.11 Port 5 (P5)

Table 41.21 Pin Function (P5)

Port Mode		Alternative Mode									
		1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative	
Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output
P5_0	P5_0	D0		MMC_D4			ET_TXD0	DV0_DATA16			LCD0_TCON0
P5_1	P5_1	D1		MMC_D5			ET_TXD1	DV0_DATA17			LCD0_TCON1
P5_2	P5_2	D2		MMC_D6			ET_TXD2	DV0_DATA18			LCD0_TCON2
P5_3	P5_3	D3		MMC_D7			ET_TXD3	DV0_DATA19			LCD0_TCON3
P5_4	P5_4	D4		RSPCK2		SSISCK1		DV0_DATA20			
P5_5	P5_5	D5		SSL20		SSIWS1		DV0_DATA21			
P5_6	P5_6	D6		MOSI2			SSITxD1	DV0_DATA22		SCK2	
P5_7	P5_7	D7		MISO2		SSIRxD1		DV0_DATA23			TxD2
P5_8	P5_8	D8		CAN0RX		TIOC4A		IRQ3			
P5_9	P5_9	D9			CAN0TX	TIOC4B		IRQ4			
P5_10	P5_10	D10		IERxD*		TIOC4C		IRQ5			
P5_11	P5_11	D11			IETxD*	TIOC4D		IRQ6			
P5_12	P5_12	D12		SSISCK2		SCK4			AUDIO_XOUT2		
P5_13	P5_13	D13		SSIWS2			AUDIO_XOUT		AUDIO_XOUT3		
P5_14	P5_14	D14		SSIDATA2		RxD4		TIOC2A			
P5_15	P5_15	D15		SD_WP_1			TxD4				

Note: * RZ/A1L only

Table 41.22 Control Registers (P5)

Register	Register Size	Valid Bit Location	Valid Bit R/W	Offset Address <PORTn_base>	Initial Value
P5	16	15 to 0	R/W	H'0014	H'0000
PSR5	32	31 to 16, 15 to 0	W, R/W	H'0114	H'0000_0000
PPR5	16	15 to 0	R	H'0214	H'0000
PM5	16	15 to 0	R/W	H'0314	H'FFFF
PMC5	16	15 to 0	R/W	H'0414	H'FFFF (In boot mode 0), H'0000 (Other than in boot mode 0)
PFC5	16	15 to 0	R/W	H'0514	H'0000
PFCE5	16	15 to 0	R/W	H'0614	H'0000
PNOT5	16	15 to 0	W	H'0714	H'0000
PMSR5	32	31 to 16, 15 to 0	W, R/W	H'0814	H'0000_FFFF
PMCSR5	32	31 to 16, 15 to 0	W, R/W	H'0914	H'0000_FFFF (In boot mode 0), H'0000_0000 (Other than in boot mode 0)
PFCAE5	16	15 to 0	R/W	H'0A14	H'0000
PIBC5	16	15 to 0	R/W	H'4014	H'0000
PBDC5	16	15 to 0	R/W	H'4114	H'FFFF (In boot mode 0), H'0000 (Other than in boot mode 0)
PIPC5	16	15 to 0	R/W	H'4214	H'FFFF (In boot mode 0), H'0000 (Other than in boot mode 0)

Caution: For the pins that are automatically configured in boot mode, see Table 8.3, Initial States by Areas in Boot Modes 0 and 1 to 3.

chanhvinh.com

41.12 Port 6 (P6)

Table 41.23 Pin Function (P6)

Port Mode		Alternative Mode															
		1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative		8th Alternative	
Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output
P6_0	P6_0	D16			LCD0_DATA8	RSPCK0		TCLKA			$\overline{\text{WDTOVF}}$						
P6_1	P6_1	D17			LCD0_DATA9	SSL00		TCLKB									
P6_2	P6_2	D18			LCD0_DATA10	MOSI0		TCLKC									
P6_3	P6_3	D19			LCD0_DATA11	MISO0		TCLKD									
P6_4	P6_4	D20			LCD0_DATA12	SSISCK3		MLB_CLK*1									
								AVB_CAPTURE*2									
P6_5	P6_5	D21			LCD0_DATA13	SSIWS3		MLB_SIG*1									
								AVB_GPTP_EXTERN*2									
P6_6	P6_6	D22			LCD0_DATA14		SSITxD3	MLB_DAT*1									
P6_7	P6_7	D23			LCD0_DATA15	SSIRxD3		IRQ0		TIOC3A		RLIN30RX*1					TRACEDATA0*3
P6_8	P6_8	D24			LCD0_DATA16	SSISCK0		IRQ1		TIOC3B			RLIN30TX*1				TRACEDATA1*3
P6_9	P6_9	D25			LCD0_DATA17	SSIWS0		IRQ2		TIOC3C							TRACEDATA2*3
P6_10	P6_10	D26			LCD0_DATA18		SSITxD0	IRQ3		TIOC3D			CAN1TX				TRACEDATA3*3
P6_11	P6_11	D27			LCD0_DATA19	SSIRxD0		SSIDATA2		SCK0		CAN1RX					TRACECTL*3
P6_12	P6_12	D28			LCD0_DATA20	RSPCK1		SSISCK2		$\overline{\text{RTS0}}$		DV0_DATA0					
P6_13	P6_13	D29			LCD0_DATA21	SSL10		SSIWS2		$\overline{\text{CTS0}}$		DV0_DATA1					
P6_14	P6_14	D30			LCD0_DATA22	MOSI1		SSIDATA2		RxD0		DV0_DATA2					
P6_15	P6_15	D31			LCD0_DATA23	MISO1					TxD0	DV0_DATA3					

Note: 1. RZ/A1L only
2. RZ/A1LU only
3. RZ/A1LU and RZ/A1LC only

chanhvinh.com

Table 41.24 Control Registers (P6)

Register	Register Size	Valid Bit Location	R/W	Offset Address <PORTn_base>	Initial Value
P6	16	15 to 0	R/W	H'0018	H'0000
PSR6	32	31 to 16, 15 to 0	W, R/W	H'0118	H'0000_0000
PPR6	16	15 to 0	R	H'0218	H'0000
PM6	16	15 to 0	R/W	H'0318	H'FFFF*
PMC6	16	15 to 0	R/W	H'0418	H'0000*
PFC6	16	15 to 0	R/W	H'0518	H'0000*
PFCE6	16	15 to 0	R/W	H'0618	H'0000*
PNOT6	16	15 to 0	W	H'0718	H'0000
PMSR6	32	31 to 16, 15 to 0	W, R/W	H'0818	H'0000_FFFF
PMCSR6	32	31 to 16, 15 to 0	W, R/W	H'0918	H'0000_0000
PFCAE6	16	15 to 0	R/W	H'0A18	H'0000*
PIBC6	16	15 to 0	R/W	H'4018	H'0000
PBDC6	16	15 to 0	R/W	H'4118	H'0000
PIPC6	16	15 to 0	R/W	H'4218	H'0000

Caution: An internal power-on reset by the watchdog timer does not initialize PM6[0], PMC6[0], PFC6[0], PFCE6[0], and PFCAE6[0].

chanhvinh.com

41.13 Port 7 (P7)

Table 41.25 Pin Function (P7)

Port Mode		Alternative Mode																	
		1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative		8th Alternative	
Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output
P7_0	P7_0	LCD0_ EXTCLK		MMC_ CD		SD_ CD_1			SPDIF_ OUT	TIOC2A		DV0_ DATA4		SCI_SCK0			TRACE CLK*2
P7_1	P7_1		$\overline{CS1}$		AUDIO_ XOUT	SD_ WP_1			TxD2			DV0_ DATA5		SCI_ RXD0/ IrRXD			
P7_2	P7_2		$\overline{CS4}$	MMC_D1		SD_D1_1		IRQ4		CAN0R X		DV0_ DATA6			SCI_ TXD0/ IrTXD		
P7_3	P7_3		$\overline{CS5}$	MMC_D0		SD_D0_1		IRQ3			CAN0T X	DV0_ DATA7		$\overline{SCI_CTS0/RTS0}$			
P7_4	P7_4	$\overline{WAIT}$			MMC_ CLK		SD_ CLK_1				IETxD*1		LCD0_ CLK	SCI_SCK1			
P7_5	P7_5		$\overline{BS}$	MMC_CMD		SD_CMD_1			TxD0	IERxD*1			LCD0_ TCON4	SCI_ RXD1			
P7_6	P7_6		$\overline{WE2}$/ DQMUL	MMC_D3		SD_D3_1		IRQ6		$\overline{CTS2}$			LCD0_ TCON5		SCI_ TXD1		
P7_7	P7_7		$\overline{WE3}$/ DQMUU /$\overline{AH}$	MMC_D2		SD_D2_1		IRQ5		$\overline{RTS2}$			LCD0_ TCON6	$\overline{SCI_CTS1/RTS1}$			
P7_8	P7_8		$\overline{CS2}$	SSISCK1		DV0_ CLK		IRQ3			TxD0						
P7_9	P7_9		A25	SSIWS1		DV0_ VSYNC		IRQ5		SCK3		TIOC1A					
P7_10	P7_10		TEND0		SSITxD1	DV0_ HSYNC				RxD3							
P7_11	P7_11		DACK0	SSIRxD1		CAN_ CLK		SCK2			TxD3		AUDIO_ XOUT		AUDIO_ XOUT3		

Note: 1. RZ/A1L only
2. RZ/A1LU and RZ/A1LC only

Table 41.26 Control Registers (P7)

Register	Register Size	Valid Bit Location	Valid Bit R/W	Offset Address <PORTn_base>	Initial Value
P7	16	11 to 0	R/W	H'001C	H'0000
PSR7	32	27 to 16, 11 to 0	W, R/W	H'011C	H'0000_0000
PPR7	16	11 to 0	R	H'021C	H'0000
PM7	16	11 to 0	R/W	H'031C	H'FFFF
PMC7	16	11 to 0	R/W	H'041C	H'0000
PFC7	16	11 to 0	R/W	H'051C	H'0000
PFCE7	16	11 to 0	R/W	H'061C	H'0000
PNOT7	16	11 to 0	W	H'071C	H'0000
PMSR7	32	27 to 16, 11 to 0	W, R/W	H'081C	H'0000_FFFF
PMCSR7	32	27 to 16, 11 to 0	W, R/W	H'091C	H'0000_0000
PFCAE7	16	11 to 0	R/W	H'0A1C	H'0000
PIBC7	16	11 to 0	R/W	H'401C	H'0000
PBDC7	16	11 to 0	R/W	H'411C	H'0000
PIPC7	16	11 to 0	R/W	H'421C	H'0000

Caution: An initial value is read if the bit is invalid. The write value should always be the initial value.

chanhvinh.com

41.14 Port 8 (P8)

Table 41.27 Pin Function (P8)

Port Mode		Alternative Mode									
		1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative	
Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output
P8_0	P8_0		LCD0_DATA0		ET_TXD0	SSISCK1		SCK3			
P8_1	P8_1		LCD0_DATA1		ET_TXD1	SSIWS1		RxD3			
P8_2	P8_2		LCD0_DATA2		ET_TXD2		SSITxD1		TxD3		
P8_3	P8_3		LCD0_DATA3		ET_TXD3	SSIRxD1					
P8_4	P8_4		LCD0_DATA4	ET_TXCLK				$\overline{\text{CTS2}}$		TIOC0A	
P8_5	P8_5		LCD0_DATA5		ET_TXER			$\overline{\text{RTS2}}$		TIOC0B	
P8_6	P8_6		LCD0_DATA6		ET_TXEN	IRQ6		$\overline{\text{CTS1}}$		TIOC0C	
P8_7	P8_7		LCD0_DATA7	ET_RXD0		IRQ7		$\overline{\text{RTS1}}$		TIOC0D	
P8_8	P8_8		LCD0_TCON0	ET_RXD1			AUDIO_XOUT	SCK2			AUDIO_XOUT3
P8_9	P8_9		LCD0_TCON1	ET_RXD2			CAN1TX	RxD2			AUDIO_XOUT2
P8_10	P8_10		LCD0_TCON2	ET_RXD3		CAN1RX			TxD2		
P8_11	P8_11		LCD0_TCON3			SSISCK2		SCK4			
P8_12	P8_12		LCD0_TCON4	SPDIF_IN		SSIWS2		RxD4			
P8_13	P8_13		LCD0_TCON5		SPDIF_OUT	SSIDATA2			TxD4		
P8_14	P8_14		LCD0_TCON6	ET_COL		SD_CD_0		SCK1			
P8_15	P8_15			ET_CRS		SD_WP_0		RxD1			

Table 41.28 Control Registers (P8)

Register	Register Size	Valid Bit		Offset Address <PORTn_base>	Initial Value
		Location	R/W		
P8	16	15 to 0	R/W	H'0020	H'0000
PSR8	32	31 to 16, 15 to 0	W, R/W	H'0120	H'0000_0000
PPR8	16	15 to 0	R	H'0220	H'0000
PM8	16	15 to 0	R/W	H'0320	H'FFFF
PMC8	16	15 to 0	R/W	H'0420	H'0000
PFC8	16	15 to 0	R/W	H'0520	H'0000
PFCE8	16	15 to 0	R/W	H'0620	H'0000
PNOT8	16	15 to 0	W	H'0720	H'0000
PMSR8	32	31 to 16, 15 to 0	W, R/W	H'0820	H'0000_FFFF
PMCSR8	32	31 to 16, 15 to 0	W, R/W	H'0920	H'0000_0000
PFCAE8	16	15 to 0	R/W	H'0A20	H'0000
PIBC8	16	15 to 0	R/W	H'4020	H'0000
PBDC8	16	15 to 0	R/W	H'4120	H'0000
PIPC8	16	15 to 0	R/W	H'4220	H'0000

chanhvinh.com

41.15 Port 9 (P9)

Table 41.29 Pin Function (P9)

Port Mode		Alternative Mode									
		1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative	
Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output
P9_0	P9_0				ET_MDC	SD_D1_0			TxD1		
P9_1	P9_1			ET_MDIO		SD_D0_0		$\overline{\text{CTS0}}$			
P9_2	P9_2	RSPCK2		ET_RXCLK			SD_CLK_0	$\overline{\text{RTS0}}$		TIOC1A	
P9_3	P9_3	SSL20		ET_RXER		SD_CMD_0		SCK0		TIOC1B	
P9_4	P9_4	MOSI2		ET_RXDV		SD_D3_0		RxD0		TIOC2A	
P9_5	P9_5	MISO2				SD_D2_0			TxD0	TIOC2B	

Table 41.30 Control Registers (P9)

Register	Register Size	Valid bit Location	Valid bit R/W	Offset Address <PORTn_base>	Initial value
P9	16	5 to 0	R/W	H'0024	H'0000
PSR9	32	21 to 16, 5 to 0	W, R/W	H'0124	H'0000_0000
PPR9	16	5 to 0	R	H'0224	H'0000
PM9	16	5 to 0	R/W	H'0324	H'FFFF
PMC9	16	5 to 0	R/W	H'0424	H'0000
PFC9	16	5 to 0	R/W	H'0524	H'0000
PFCE9	16	5 to 0	R/W	H'0624	H'0000
PNOT9	16	5 to 0	W	H'0724	H'0000
PMSR9	32	21 to 16, 5 to 0	W, R/W	H'0824	H'0000_FFFF
PMCSR9	32	21 to 16, 5 to 0	W, R/W	H'0924	H'0000_0000
PFCAE9	16	5 to 0	R/W	H'0A24	H'0000
PIBC9	16	5 to 0	R/W	H'4024	H'0000
PBDC9	16	5 to 0	R/W	H'4124	H'0000
PIPC9	16	5 to 0	R/W	H'4224	H'0000

Caution: An initial value is read if the bit is invalid. The write value should always be the initial value.

chanhvinh.com

41.16 Port Control Logical Diagram

The figure below is a logical diagram of the port control function.
Note: This figure shows the logic for reference, not the circuit.

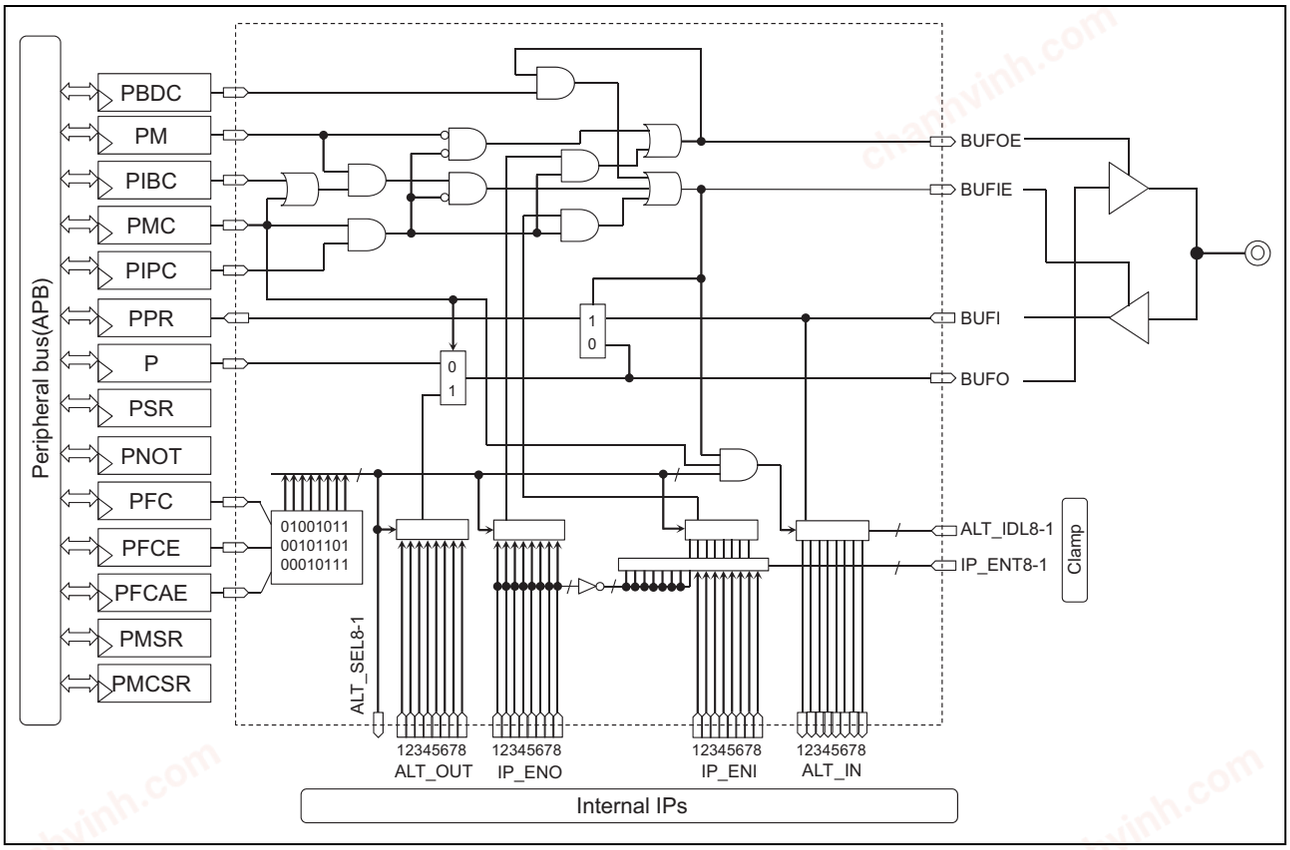


Figure 41.1 Port Control Logical Diagram

chanhvinh.com

41.17 Flowchart Examples of Port Setting

(a) shows an example flow of setting the Pn_m pin to port mode. (b) shows an example flow of setting the Pn_m pin to software I/O control alternative mode. (c) shows an example flow of setting the Pn_m pin to direct I/O control alternative mode.

(a) In Port Mode

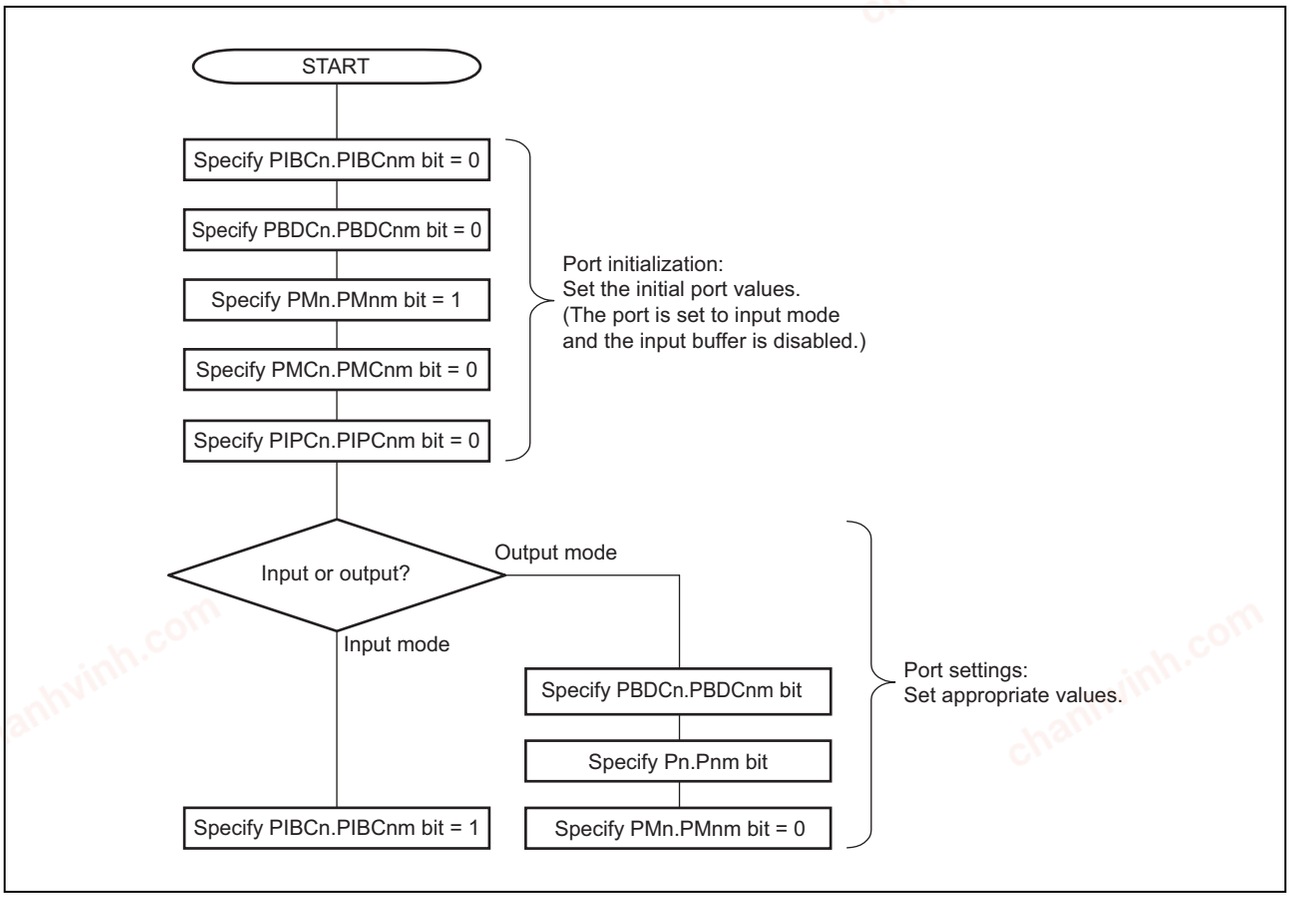


Figure 41.2 Example Flow of Port Settings (in Port Mode) (a)

chanhvinh.com

(b) In Software I/O Control Alternative Mode

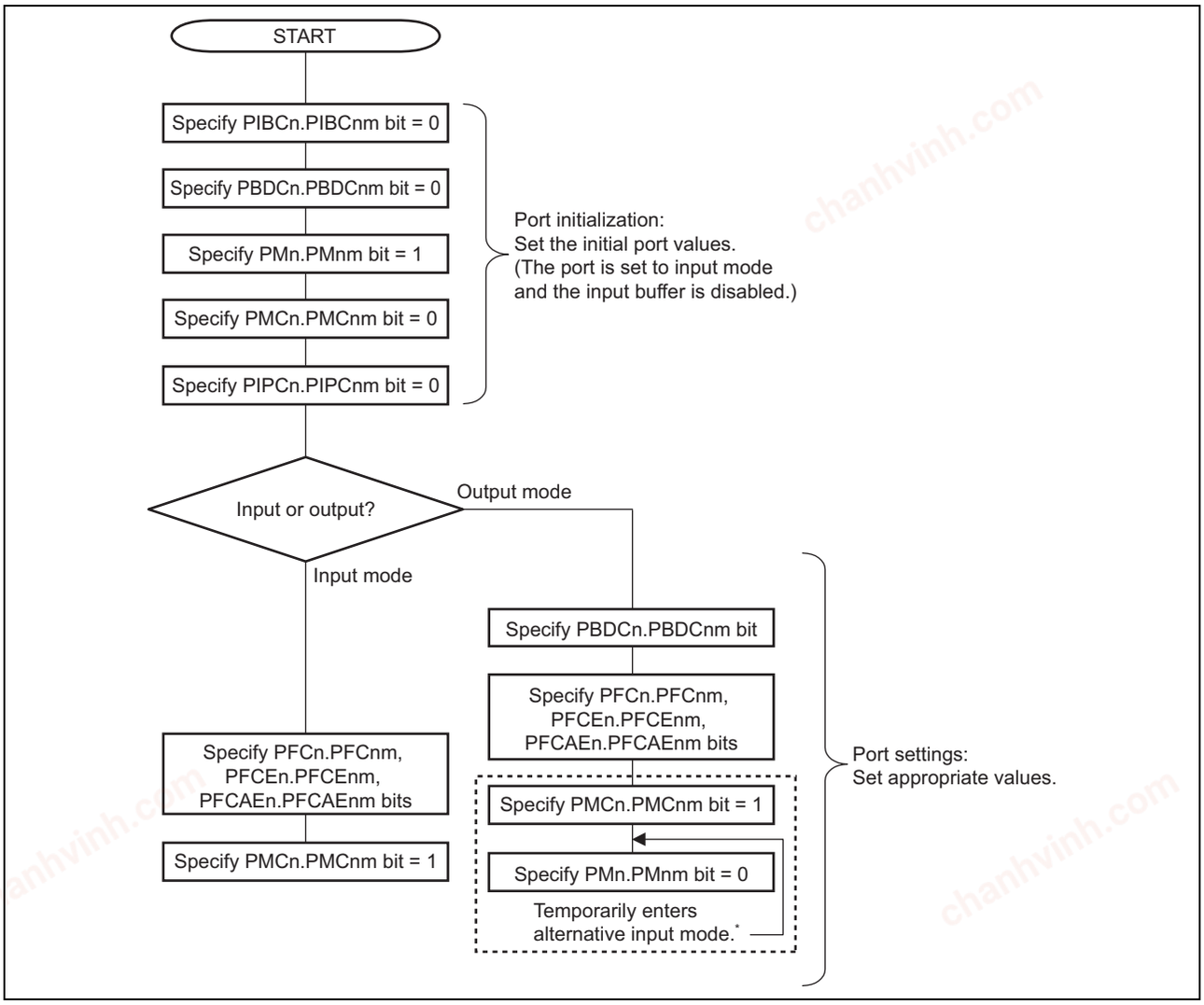


Figure 41.3 Example Flow of Port Settings (in Software I/O Control Alternative Mode) (b)

Caution: When the Pn_m pin is set to the output mode of the software I/O control alternative mode, alternative input mode is temporarily enabled during the period from when the PMCn.PMCnm bit is set to 1 to when the PMn.PMnm bit is set to 0. Therefore, if a signal regarding interrupt has been set for the port alternative function, configure the setting so that interrupt does not occur or becomes invalid.

chanhvinh.com

(c) In Direct I/O Control Alternative Mode

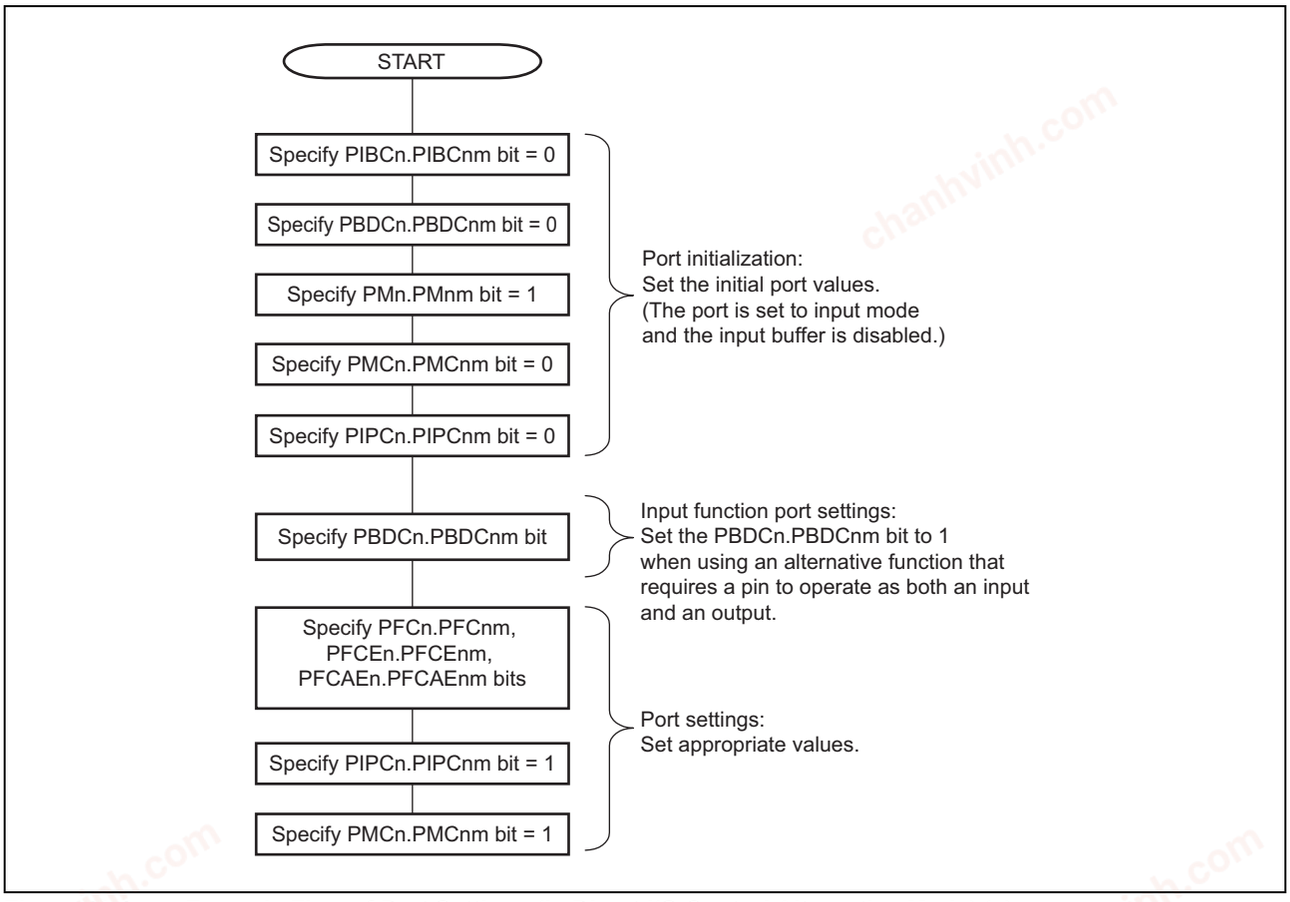


Figure 41.4 Example Flow of Port Settings (in Direct I/O Control Alternative Mode) (c)

chanhvinh.com

42. Power-Down Modes

This LSI supports sleep mode, software standby mode, deep standby mode, and module standby mode. In power-down modes, functions of CPU, clocks, on-chip memory, or part of on-chip peripheral modules are halted or the power-supply is turned off, through which low power consumption is achieved. These modes are canceled by a reset or interrupt.

42.1 Features

42.1.1 States of Processing and Power-Down Modes

(1) States of processing

This LSI has three general states of processing: the reset state, the program execution state, and the power-down modes. Figure 42.1 shows transitions between these states.

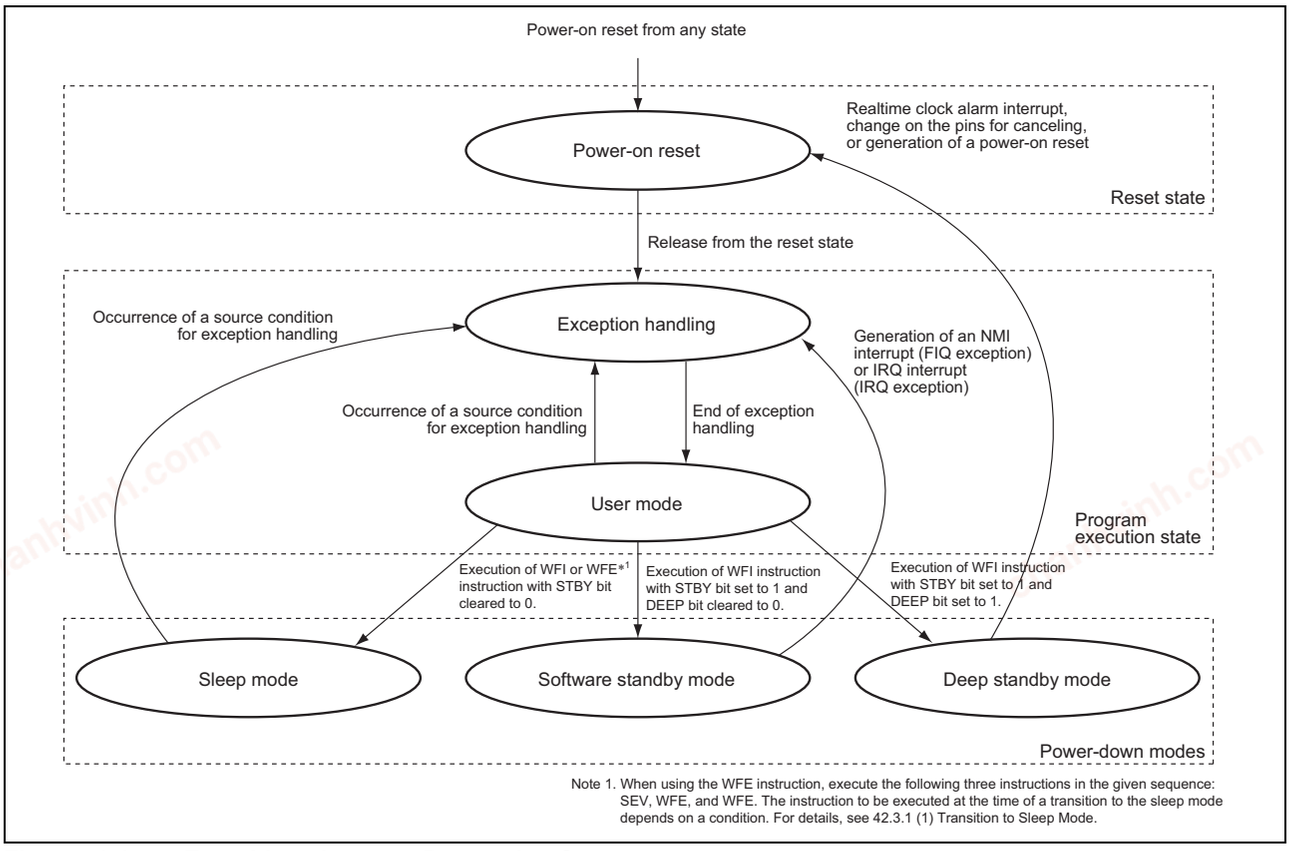


Figure 42.1 Transitions between Processing States

(2) Power-down modes

This LSI has the following power-down modes and function:

1. Sleep mode
2. Software standby mode
3. Deep standby mode
4. Module standby function

Table 42.1 shows the transition conditions for entering the modes from the program execution state, as well as the CPU and peripheral module states in each mode and the procedures for canceling each mode.

chanhvinh.com

Table 42.1 States of Power-Down Modes

Power-Down Mode	Transition Conditions	State: CPU, CPU Register, Primary Cache, TLB	State: Secondary Cache	State: Large-Capacity On-Chip RAM (including on-chip data-retention RAM)	State: On-Chip Peripheral Modules	State: Realtime Clock	State: Internal Power Supply	State: External Memory	Canceling Procedure
Sleep mode	Execute WFI or WFE*6 instruction with STBY bit in STBCR1 cleared to 0	Halted Contents are held.*4	Running	Running	Running	Running*1	Applied	Auto-refresh	• Interrupt • Power-on reset
Software standby mode	Execute WFI instruction with STBY and DEEP bits in STBCR1 set to 1 and 0, respectively	Halted Contents are held.*4	Halted Contents are held.*4	Halted Contents are held.*5	Halted	Running*1	Applied	Self-refresh	• NMI or IRQ interrupt • Power-on reset
Deep standby mode	Execute WFI instruction with STBY and DEEP bits in STBCR1 set to 1	Halted Contents are not held.	Halted Contents are not held.	Halted Contents in on-chip data-retention RAM are held*2, and the other contents in large-capacity on-chip RAM are not held.	Halted	Running*1	Shut off	Self-refresh	• Power-on reset*3 • Realtime clock alarm interrupt*3 • Change on the pins for canceling*3
Module standby mode	Set the MSTP bits in STBCR2 to STBCR12 to 1	Running	Running	Running	Specified module halted	Halted	Applied	Auto-refresh	• Clear MSTP bit to 0

Note 1. The realtime clock operates when the START bit in the RCR2 register is set to 1. For details, see section 13, Realtime Clock. When deep standby mode is canceled by a power-on reset, the running state cannot be retained. Make the initial setting for the realtime clock again.

Note 2. Setting the bits RRAMKP3 to RRAMKP0 in the RRAMKP register to 1 enables to retain the data in the corresponding area on the on-chip data-retention RAM during the transition to deep standby mode. When the deep standby mode is canceled by a power-on reset, the retained contents are initialized.

Note 3. Deep standby mode can be canceled by a power-on reset, a realtime clock alarm interrupt, or change on the pins for canceling (P2_0, P2_2, P2_7, P2_9, P5_8, P5_9, P5_10, P6_7, P7_2, P7_3, P7_6, P7_9, and NMI). Even when deep standby mode is canceled by a source other than a power-on reset, the reset exception handling is executed instead of the interrupt exception handling.

Note 4. When sleep mode or software standby mode is canceled by a power-on reset, the retained contents are initialized.

Note 5. By setting the VRAME bit in the SYSCR1 register or VRAMWE bit in the SYSCR2 register to enable accesses, the retained contents are initialized when software standby mode is canceled by a power-on reset. By setting the VRAME bit in the SYSCR1 register or VRAMWE bit in the SYSCR2 register to disable accesses, contents in the large-capacity on-chip RAM (including on-chip data-retention RAM) can be retained when software standby mode is canceled by a power-on reset. Note that the area at addresses from H'2002_0000 to H'2002_3FFF is used as working memory for the boot program in boot modes 1 to 3. In boot modes 2 and 3, a 28-Kbyte program is transferred from the external flash memory to the area at addresses from H'2002_4000 to H'2002_AFFF. For details, see section 3, Boot Mode.

Note 6. When using the WFE instruction, execute the following three instructions in the given sequence: SEV, WFE, and WFE. The instruction to be executed at the time of a transition to the sleep mode depends on a condition. For details, see section 42.3.1, (1) Transition to Sleep Mode.

chanhvinh.com

42.2 Register Descriptions

Table 42.2 shows the register configuration.

Table 42.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Standby control register 1	STBCR1	R/W	H'00	H'FCFE0020	8
Standby control register 2	STBCR2	R/W	H'6A	H'FCFE0024	8
Standby control register 3	STBCR3	R/W	H'FD	H'FCFE0420	8
Standby control register 4	STBCR4	R/W	H'FF	H'FCFE0424	8
Standby control register 5	STBCR5	R/W	H'FF	H'FCFE0428	8
Standby control register 6	STBCR6	R/W	H'FE	H'FCFE042C	8
Standby control register 7	STBCR7	R/W	H'3F	H'FCFE0430	8
Standby control register 8	STBCR8	R/W	H'FF	H'FCFE0434	8
Standby control register 9	STBCR9	R/W	H'FF	H'FCFE0438	8
Standby control register 10	STBCR10	R/W	H'FF	H'FCFE043C	8
Standby control register 11	STBCR11	R/W	H'FF	H'FCFE0440	8
Standby control register 12	STBCR12	R/W	H'FF	H'FCFE0444	8
Software reset control register 1	SWRSTCR1	R/W	H'00	H'FCFE0460	8
Software reset control register 2 (RZ/A1LU only)	SWRSTCR2	R/W	H'00	H'FCFE0464	8
System control register 1	SYSCR1	R/W	H'FF	H'FCFE0400	8
System control register 2	SYSCR2	R/W	H'FF	H'FCFE0404	8
System control register 3	SYSCR3	R/W	H'00	H'FCFE0408	8
CPU status register	CPUSTS	R	H'00	H'FCFE0018	8
Standby request register 1	STBREQ1	R/W	H'00	H'FCFE0030	8
Standby request register 2	STBREQ2	R/W	H'00	H'FCFE0034	8
Standby acknowledge register 1	STBACK1	R	H'00	H'FCFE0040	8
Standby acknowledge register 2	STBACK2	R	H'00	H'FCFE0044	8
On-chip data-retention RAM area setting register	RRAMKP	R/W	H'00	H'FCFF1800	8
Deep standby control register	DSCTR	R/W	H'00	H'FCFF1802	8
Deep standby cancel source select register	DSSSR	R/W	H'0000	H'FCFF1804	16
Deep standby cancel edge select register	DSESR	R/W	H'0000	H'FCFF1806	16
Deep standby cancel source flag register	DSFR	R/W	H'0000	H'FCFF1808	16
XTAL crystal oscillator gain control register	XTALCTR	R/W	H'00	H'FCFF1810	8

chanhvinh.com

42.2.1 Standby Control Register 1 (STBCR1)

STBCR1 is an 8-bit readable/writable register that specifies the state of the power-down mode.

Note: • When writing to this register, see section 42.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	STBY	DEEP	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	STBY	0	R/W	Software Standby, Deep Standby
6	DEEP	0	R/W	Specifies transition to software standby mode or deep standby mode. 0x: Executing WFI or WFE*1 instruction puts chip into sleep mode. 10: Executing WFI instruction puts chip into software standby mode.*2 11: Executing WFI instruction puts chip into deep standby mode.*2 Note 1. When using the WFE instruction, execute the following three instructions in the given sequence: SEV, WFE, and WFE. The instruction to be executed at the time of a transition to the sleep mode depends on a condition. For details, see section 42.3.1, (1) Transition to Sleep Mode. Note 2. Do not execute the WFE instruction while the STBY bit in STBCR1 is 1.
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

[Legend]
x: Don't care

chanhvinh.com

42.2.2 Standby Control Register 2 (STBCR2)

STBCR2 is an 8-bit readable/writable register that controls the operation of each module.

Note: • When writing to this register, see section 42.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	HIZ	-	-	-	-	-	-	MSTP 20
Initial value:	0	1	1	0	1	0	1	0
R/W:	R/W	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	HIZ	0	R/W	Port High Impedance Selects whether the state of specific output pin is retained or high impedance in software standby mode or deep standby mode. As to which pins are controlled, see section 48.1, Pin States in section 48, States and Handling of Pins. 0: The pin state is retained in software standby mode or deep standby mode. 1: The pin is set to high-impedance in software standby mode or deep standby mode.
6, 5	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
4	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
3	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
0	MSTP20	0	R/W	Module Stop 20*1 [not for disclosure] When the MSTP20 bit is set to 1, the clock supply to CoreSight is halted. 0: CoreSight runs. 1: Clock supply to CoreSight is halted.

Note 1. The transition to and release from module stop mode proceed in the steps described in section 42.3.5, Module Standby Function.

chanhvinh.com

42.2.3 Standby Control Register 3 (STBCR3)

STBCR3 is an 8-bit readable/writable register that controls the operation of each module.

Note: • When writing to this register, see section 42.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	MSTP 37*1	MSTP 36	MSTP 35*1	—	MSTP 33	MSTP 32	MSTP 31	—
Initial value:	1	1	1	1	1	1	0	1
R/W:	R/W	R/W	R/W	R	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP37*1	1	R/W	Module Stop 37 When the MSTP37 bit is set to 1, the clock supply to the IEBus™ controller is halted. 0: The IEBus™ controller runs. 1: Clock supply to the IEBus™ controller is halted.
6	MSTP36	1	R/W	Module Stop 36 When the MSTP36 bit is set to 1, the clock supply to serial communication interface (IrDA) is halted. 0: Serial communication interface (IrDA) runs. 1: Clock supply to serial communication interface (IrDA) is halted.
5	MSTP35*1	1	R/W	Module Stop 35 When the MSTP35 bit is set to 1, the clock supply to the LIN interface channel 0 is halted. 0: The LIN interface channel 0 runs. 1: Clock supply to the LIN interface channel 0 is halted.
4	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
3	MSTP33	1	R/W	Module Stop 33 When the MSTP33 bit is set to 1, the clock supply to the multi-function timer pulse unit 2 is halted. 0: The multi-function timer pulse unit 2 runs. 1: Clock supply to the multi-function timer pulse unit 2 is halted.
2	MSTP32	1	R/W	Module Stop 32 When the MSTP32 bit is set to 1, the clock supply to the CAN interface is halted. 0: The CAN interface runs. 1: Clock supply to the CAN interface is halted.
1	MSTP31	0	R/W	Module Stop 31 When the MSTP31 bit is set to 1, the clock supply to the A/D converter is halted and the analog power supply to the A/D converter is shut off. 0: Clock signal and analog voltage are supplied to the A/D converter. 1: Supply of clock signal and analog voltage to the A/D converter is shut off.
0	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.

Note 1. This bit is only present in the RZ/A1L. For the RZ/A1LU and RZ/A1LC, the write value should always be 1.

chanhvinh.com

42.2.4 Standby Control Register 4 (STBCR4)

STBCR4 is an 8-bit readable/writable register that controls the operation of each module.

Note: • When writing to this register, see section 42.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	MSTP47	MSTP46	MSTP45	MSTP44	MSTP43	—	—	—
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP47	1	R/W	Module Stop 47 When the MSTP47 bit is set to 1, the clock supply to channel 0 of the serial communication interface with FIFO is halted. 0: Channel 0 of the serial communication interface with FIFO runs. 1: Clock supply to channel 0 of the serial communication interface with FIFO is halted.
6	MSTP46	1	R/W	Module Stop 46 When the MSTP46 bit is set to 1, the clock supply to channel 1 of the serial communication interface with FIFO is halted. 0: Channel 1 of the serial communication interface with FIFO runs. 1: Clock supply to channel 1 of the serial communication interface with FIFO is halted.
5	MSTP45	1	R/W	Module Stop 45 When the MSTP45 bit is set to 1, the clock supply to channel 2 of the serial communication interface with FIFO is halted. 0: Channel 2 of the serial communication interface with FIFO runs. 1: Clock supply to channel 2 of the serial communication interface with FIFO is halted.
4	MSTP44	1	R/W	Module Stop 44 When the MSTP44 bit is set to 1, the clock supply to channel 3 of the serial communication interface with FIFO is halted. 0: Channel 3 of the serial communication interface with FIFO runs. 1: Clock supply to channel 3 of the serial communication interface with FIFO is halted.
3	MSTP43	1	R/W	Module Stop 43 When the MSTP43 bit is set to 1, the clock supply to channel 4 of the serial communication interface with FIFO is halted. 0: Channel 4 of the serial communication interface with FIFO runs. 1: Clock supply to channel 4 of the serial communication interface with FIFO is halted.
2 to 0	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.

chanhvinh.com

42.2.5 Standby Control Register 5 (STBCR5)

STBCR5 is an 8-bit readable/writable register that controls the operation of each module.

Note: • When writing to this register, see section 42.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	MSTP57	MSTP56	—	—	—	—	MSTP51	MSTP50
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP57	1	R/W	Module Stop 57 When the MSTP57 bit is set to 1, the clock supply to channel 0 of the serial communication interface (SCI) is halted. 0: Channel 0 of the serial communication interface (SCI) runs. 1: Clock supply to channel 0 of the serial communication interface (SCI) is halted.
6	MSTP56	1	R/W	Module Stop 56 When the MSTP56 bit is set to 1, the clock supply to channel 1 of the serial communication interface (SCI) is halted. 0: Channel 1 of the serial communication interface (SCI) runs. 1: Clock supply to channel 1 of the serial communication interface (SCI) is halted.
5 to 2	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
1	MSTP51	1	R/W	Module Stop 51 When the MSTP51 bit is set to 1, the clock supply to channel 0 of the OS timer is halted. 0: Channel 0 of the OS timer runs. 1: Clock supply to channel 0 of the OS timer is halted.
0	MSTP50	1	R/W	Module Stop 50 When the MSTP50 bit is set to 1, the clock supply to channel 1 of the OS timer is halted. 0: Channel 1 of the OS timer runs. 1: Clock supply to channel 1 of the OS timer is halted.

chanhvinh.com

42.2.6 Standby Control Register 6 (STBCR6)

STBCR6 is an 8-bit readable/writable register that controls the operation of each module.

Note: • When writing to this register, see section 42.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	MSTP 67	MSTP 66	—	—	—	—	MSTP 61*2	MSTP 60
Initial value:	1	1	1	1	1	1	1	0
R/W:	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP67	1	R/W	Module Stop 67 When the MSTP67 bit is set to 1, the clock supply to the A/D converter is halted. 0: The A/D converter runs. 1: Clock supply to channel 0 of the A/D converter is halted.
6	MSTP66	1	R/W	Module Stop 66*1 When the MSTP66 bit is set to 1, the clock supply to the capture engine unit is halted. 0: The capture engine unit runs. 1: Clock supply to the capture engine unit is halted.
5 to 2	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
1	MSTP61*2	1	R/W	Module Stop 61*1 When the MSTP61 bit is set to 1, the clock supply to the JPEG codec unit is halted. 0: The JPEG codec unit runs. 1: Clock supply to the JPEG codec unit is halted.
0	MSTP60	0	R/W	Module Stop 60 When the MSTP60 bit is set to 1, the clock supply to the realtime clock is halted. 0: The realtime clock runs. 1: Clock supply to the realtime clock is halted. Note: When the realtime clock is halted, set the bits in registers shown below. • Set bit RTCEN in the control register 2 (RCR2) to 0. • Set bits RCKSEL[1:0] in the control register 5 (RCR5) to 00. After the settings above, set bit MSTP60 to 1.

Note 1. The transition to and release from module stop mode proceed in the steps described in section 42.3.5, Module Standby Function.

Note 2. This bit is only present in the RZ/A1LU. For the RZ/A1L and RZ/A1LC, the write value should always be 1.

chanhvinh.com

42.2.7 Standby Control Register 7 (STBCR7)

STBCR7 is an 8-bit readable/writable register that controls the operation of each module.

Note: • When writing to this register, see section 42.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	MSTP74	—	—	MSTP71	MSTP70
Initial value:	0	0	1	1	1	1	1	1
R/W:	R	R	R	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
4	MSTP74	1	R/W	Module Stop 74*1 When the MSTP74 bit is set to 1, the clock supply to the Ethernet controller is halted. 0: The Ethernet controller runs. 1: Clock supply to the Ethernet controller is halted.
3, 2	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
1	MSTP71	1	R/W	Module Stop 71 When the MSTP71 bit is set to 1, the clock supply to channel 0 of the USB 2.0 host/function module is halted. 0: Channel 0 of the USB 2.0 host/function module runs. 1: Clock supply to channel 0 of the USB 2.0 host/function module is halted.
0	MSTP70	1	R/W	Module Stop 70 When the MSTP70 bit is set to 1, the clock supply to channel 1 of the USB 2.0 host/function module is halted. 0: Channel 1 of the USB 2.0 host/function module runs. 1: Clock supply to channel 1 of the USB 2.0 host/function module is halted.

Note 1. The transition to and release from module stop mode proceed in the steps described in section 42.3.5, Module Standby Function.

chanhvinh.com

42.2.8 Standby Control Register 8 (STBCR8)

STBCR8 is an 8-bit readable/writable register that controls the operation of each module.

Note: • When writing to this register, see section 42.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	MSTP84	MSTP83*2	MSTP82*3	MSTP81	—
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
4	MSTP84	1	R/W	Module Stop 84 When the MSTP84 bit is set to 1, the clock supply to the MMC host interface is halted. 0: The MMC host interface runs. 1: Clock supply to the MMC host interface is halted.
3	MSTP83*2	1	R/W	Module Stop 83*1 When the MSTP83 bit is set to 1, the clock supply to the media local bus is halted. 0: The media local bus runs. 1: Clock supply to the media local bus is halted.
2	MSTP82*3	1	R/W	Module Stop 82*1 When the MSTP82 bit is set to 1, the clock supply to the EthernetAVB is halted. 0: The EthernetAVB runs. 1: Clock supply to the EthernetAVB is halted.
1	MSTP81	1	R/W	Module Stop 81 When the MSTP81 bit is set to 1, the clock supply to SCUX is halted. 0: SCUX runs. 1: Clock supply to SCUX is halted.
0	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.

Note 1. The transition to and release from module stop mode proceed in the steps described in section 42.3.5, Module Standby Function.
Note 2. This bit is only present in the RZ/A1L. For the RZ/A1LU and RZ/A1LC, the write value should always be 1.
Note 3. This bit is only present in the RZ/A1LU. For the RZ/A1L and RZ/A1LC, the write value should always be 1.

chanhvinh.com

42.2.9 Standby Control Register 9 (STBCR9)

STBCR9 is an 8-bit readable/writable register that controls the operation of each module.

Note: • When writing to this register, see section 42.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	MSTP 97	MSTP 96	MSTP 95	MSTP 94	MSTP 93	—	MSTP 91	—
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP97	1	R/W	Module Stop 97 When the MSTP97 bit is set to 1, the clock supply to channel 0 of the I²C bus interface is halted. 0: Channel 0 of the I²C bus interface runs. 1: Clock supply to channel 0 of the I²C bus interface is halted.
6	MSTP96	1	R/W	Module Stop 96 When the MSTP96 bit is set to 1, the clock supply to channel 1 of the I²C bus interface is halted. 0: Channel 1 of the I²C bus interface runs. 1: Clock supply to channel 1 of the I²C bus interface is halted.
5	MSTP95	1	R/W	Module Stop 95 When the MSTP95 bit is set to 1, the clock supply to channel 2 of the I²C bus interface is halted. 0: Channel 2 of the I²C bus interface runs. 1: Clock supply to channel 2 of the I²C bus interface is halted.
4	MSTP94	1	R/W	Module Stop 94 When the MSTP94 bit is set to 1, the clock supply to channel 3 of the I²C bus interface is halted. 0: Channel 3 of the I²C bus interface runs. 1: Clock supply to channel 3 of the I²C bus interface is halted.
3	MSTP93	1	R/W	Module Stop 93 When the MSTP93 bit is set to 1, the clock supply to channel 0 of the SPI multi I/O bus controller is halted. 0: Channel 0 of the SPI multi I/O bus controller runs. 1: Clock supply to channel 0 of the SPI multi I/O bus controller is halted.
2	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
1	MSTP91	1	R/W	Module Stop 91*1 When the MSTP91 bit is set to 1, the clock supply to the video display controller 5 is halted. 0: The video display controller 5 runs. 1: Clock supply to the video display controller 5 is halted.
0	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.

Note 1. The transition to and release from module stop mode proceed in the steps described in section 42.3.5, Module Standby Function.

chanhvinh.com

42.2.10 Standby Control Register 10 (STBCR10)

STBCR10 is an 8-bit readable/writable register that controls the operation of each module.

Note: • When writing to this register, see section 42.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	MSTP 107	MSTP 106	MSTP 105	—	—	MSTP 102*1	MSTP 101	—
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R	R	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP107	1	R/W	Module Stop 107 When the MSTP107 bit is set to 1, the clock supply to channel 0 of the Renesas serial peripheral interface is halted. 0: Channel 0 of the Renesas serial peripheral interface runs. 1: Clock supply to channel 0 of the Renesas serial peripheral interface is halted.
6	MSTP106	1	R/W	Module Stop 106 When the MSTP106 bit is set to 1, the clock supply to channel 1 of the Renesas serial peripheral interface is halted. 0: Channel 1 of the Renesas serial peripheral interface runs. 1: Clock supply to channel 1 of the Renesas serial peripheral interface is halted.
5	MSTP105	1	R/W	Module Stop 105 When the MSTP105 bit is set to 1, the clock supply to channel 2 of the Renesas serial peripheral interface is halted. 0: Channel 2 of the Renesas serial peripheral interface runs. 1: Clock supply to channel 2 of the Renesas serial peripheral interface is halted.
4, 3	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
2	MSTP102*1	1	R/W	Module Stop 102 When the MSTP102 bit is set to 1, the clock supply to the CD-ROM decoder is halted. 0: The CD-ROM decoder runs. 1: Clock supply to the CD-ROM decoder is halted.
1	MSTP101	1	R/W	Module Stop 101 When the MSTP101 bit is set to 1, the clock supply to the Renesas SPDIF interface is halted. 0: The Renesas SPDIF interface runs. 1: Clock supply to the Renesas SPDIF interface is halted.
0	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.

Note 1. This bit is only present in the RZ/A1L. For the RZ/A1LU and RZ/A1LC, the write value should always be 1.

chanhvinh.com

42.2.11 Standby Control Register 11 (STBCR11)

STBCR11 is an 8-bit readable/writable register that controls the operation of each module.

Note: • When writing to this register, see section 42.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	—	—	MSTP115	MSTP114	MSTP113	MSTP112	—	—
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
5	MSTP115	1	R/W	Module Stop 115 When the MSTP115 bit is set to 1, the clock supply to channel 0 of the serial sound interface is halted. 0: Channel 0 of the serial sound interface runs. 1: Clock supply to channel 0 of the serial sound interface is halted.
4	MSTP114	1	R/W	Module Stop 114 When the MSTP114 bit is set to 1, the clock supply to channel 1 of the serial sound interface is halted. 0: Channel 1 of the serial sound interface runs. 1: Clock supply to channel 1 of the serial sound interface is halted.
3	MSTP113	1	R/W	Module Stop 113 When the MSTP113 bit is set to 1, the clock supply to channel 2 of the serial sound interface is halted. 0: Channel 2 of the serial sound interface runs. 1: Clock supply to channel 2 of the serial sound interface is halted.
2	MSTP112	1	R/W	Module Stop 112 When the MSTP112 bit is set to 1, the clock supply to channel 3 of the serial sound interface is halted. 0: Channel 3 of the serial sound interface runs. 1: Clock supply to channel 3 of the serial sound interface is halted.
1, 0	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.

chanhvinh.com

42.2.12 Standby Control Register 12 (STBCR12)

STBCR12 is an 8-bit readable/writable register that controls the operation of each module.

Note: • When writing to this register, see section 42.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	MSTP123	MSTP122	MSTP121	MSTP120
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
3	MSTP123	1	R/W	Module Stop 123 When the MSTP123 bit is set to 1, the SD host interface 00 is halted. 0: The SD host interface 00 runs. 1: Clock supply to the SD host interface 00 is halted.
2	MSTP122	1	R/W	Module Stop 122 When the MSTP122 bit is set to 1, the SD host interface 01 is halted. 0: The SD host interface 01 runs. 1: Clock supply to the SD host interface 01 is halted.
1	MSTP121	1	R/W	Module Stop 121 When the MSTP121 bit is set to 1, the SD host interface 10 is halted. 0: The SD host interface 10 runs. 1: Clock supply to the SD host interface 10 is halted.
0	MSTP120	1	R/W	Module Stop 120 When the MSTP120 bit is set to 1, the SD host interface 11 is halted. 0: The SD host interface 11 runs. 1: Clock supply to the SD host interface 11 is halted.

Note: For details on this register, see section 38.3.2, Card Detect/Write Protect.

chanhvinh.com

42.2.13 Software Reset Control Register 1 (SWRSTCR1)

SWRSTCR1 is an 8-bit readable/writable register that controls a software reset for the serial sound interface and the operation of the crystal resonator for audio.

Note: • When writing to this register, see section 42.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	AXT ALE	SRST 16	SRST 15	SRST 14	SRST 13	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	AXTALE	0	R/W	AUDIO_X1 Clock Control Controls the function of AUDIO_X1 pin. 0: Runs the on-chip crystal oscillator/enables the external clock input. 1: Halts the on-chip crystal oscillator/disables the external clock input.
6	SRST16	0	R/W	Serial Sound Interface Channel 0 Software Reset Controls the serial sound interface channel 0 reset with software. 0: The serial sound interface channel 0 reset is canceled. 1: The serial sound interface channel 0 is reset.
5	SRST15	0	R/W	Serial Sound Interface Channel 1 Software Reset Controls the serial sound interface channel 1 reset with software. 0: The serial sound interface channel 1 reset is canceled. 1: The serial sound interface channel 1 is reset.
4	SRST14	0	R/W	Serial Sound Interface Channel 2 Software Reset Controls the serial sound interface channel 2 reset with software. 0: The serial sound interface channel 2 reset is canceled. 1: The serial sound interface channel 2 is reset.
3	SRST13	0	R/W	Serial Sound Interface Channel 3 Software Reset Controls the serial sound interface channel 3 reset with software. 0: The serial sound interface channel 3 reset is canceled. 1: The serial sound interface channel 3 is reset.
2 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

42.2.14 Software Reset Control Register 2 (SWRSTCR2)

This register is only provided in the RZ/A1LU.

SWRSTCR2 is an 8-bit readable/writable register that controls a software reset for each module.

Note: • When writing to this register, see section 42.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	SRST21	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	SRST21	0	R/W	JPEG Codec Unit Software Reset Controls the JPEG codec unit reset with software. 0: The JPEG codec unit reset is canceled. 1: The JPEG codec unit is reset.
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

chanhvinh.com

42.2.15 System Control Register 1 (SYSCR1)

SYSCR1 is an 8-bit readable/writable register that enables or disables access (read and write) to a specified page in the large-capacity on-chip RAM.

When a VRAMEn (n = 0 to 4) bit in SYSCR1 is set to 1, access to page n is enabled. When a VRAMEn bit is cleared to 0, page n cannot be accessed. In this case, an undefined value is returned when reading data or fetching an instruction from page n, and writing to page n is ignored. The initial value of a VRAMEn bit is 1.

SYSCR1 should be set with a program located in an area other than the large-capacity on-chip RAM. Furthermore, an instruction to read SYSCR1 should be located immediately after the instruction to write to SYSCR1. If not, normal access is not guaranteed.

Note: • When writing to this register, see section 42.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	VRAME 4	VRAME 3	VRAME 2	VRAME 1	VRAME 0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
4	VRAME4	1	R/W	RAM Enable 4 (corresponding area: page 4*1 in large-capacity on-chip RAM) 0: Access to page 4 is disabled. 1: Access to page 4 is enabled.
3	VRAME3	1	R/W	RAM Enable 3 (corresponding area: page 3*1 in large-capacity on-chip RAM) 0: Access to page 3 is disabled. 1: Access to page 3 is enabled.
2	VRAME2	1	R/W	RAM Enable 2 (corresponding area: page 2*1 in large-capacity on-chip RAM) 0: Access to page 2 is disabled. 1: Access to page 2 is enabled.
1	VRAME1	1	R/W	RAM Enable 1 (corresponding area: page 1*1 in large-capacity on-chip RAM) 0: Access to page 1 is disabled. 1: Access to page 1 is enabled.
0	VRAME0	1	R/W	RAM Enable 0 (corresponding area: page 0*1 in large-capacity on-chip RAM) 0: Access to page 0 is disabled. 1: Access to page 0 is enabled.

Note 1. For addresses in each page, see section 40, On-Chip RAM.

chanhvinh.com

42.2.16 System Control Register 2 (SYSCR2)

SYSCR2 is an 8-bit readable/writable register that enables or disables writing to a specified page in the large-capacity on-chip RAM.

When a VRAMWEn (n = 0 to 4) bit in SYSCR2 is set to 1, writing to page n is enabled. When a VRAMWEn bit is cleared to 0, writing to page n is ignored. The initial value of a VRAMWEn bit is 1.

SYSCR2 should be set with a program located in an area other than the large-capacity on-chip RAM. Furthermore, an instruction to read SYSCR2 should be located immediately after the instruction to write to SYSCR2. If not, normal access is not guaranteed.

Note: • When writing to this register, see section 42.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	VRAM WE4	VRAM WE3	VRAM WE2	VRAM WE1	VRAM WE0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
4	VRAMWE4	1	R/W	RAM Write Enable 4 (corresponding area: page 4*1 in large-capacity on-chip RAM) 0: Writing to page 4 is disabled. 1: Writing to page 4 is enabled.
3	VRAMWE3	1	R/W	RAM Write Enable 3 (corresponding area: page 3*1 in large-capacity on-chip RAM) 0: Writing to page 3 is disabled. 1: Writing to page 3 is enabled.
2	VRAMWE2	1	R/W	RAM Write Enable 2 (corresponding area: page 2*1 in large-capacity on-chip RAM) 0: Writing to page 2 is disabled. 1: Writing to page 2 is enabled.
1	VRAMWE1	1	R/W	RAM Write Enable 1 (corresponding area: page 1*1 in large-capacity on-chip RAM) 0: Writing to page 1 is disabled. 1: Writing to page 1 is enabled.
0	VRAMWE0	1	R/W	RAM Write Enable 0 (corresponding area: page 0*1 in large-capacity on-chip RAM) 0: Writing to page 0 is disabled. 1: Writing to page 0 is enabled.

Note 1. For addresses in each page, see section 40, On-Chip RAM.

chanhvinh.com

42.2.17 System Control Register 3 (SYSCR3)

SYSCR3 is an 8-bit readable/writable register that enables or disables writing to a specified page in the on-chip data-retention RAM.

When a RRAMWEn (n = 0 to 3) bit in SYSCR3 is set to 1, writing to page n is enabled. When a RRAMWEn bit is cleared to 0, writing to page n is ignored. The initial value of a RRAMWEn bit is 0.

SYSCR3 should be set with a program located in an area other than the on-chip data-retention RAM.

Note: • When writing to this register, see section 42.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	RRAMWE3	RRAMWE2	RRAMWE1	RRAMWE0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	RRAMWE3	0	R/W	RAM Write Enable 3 (corresponding area: page 3*2 in on-chip data-retention RAM) 0: Writing to page 3 is disabled. 1: Writing to page 3 is enabled.
2	RRAMWE2	0	R/W	RAM Write Enable 2 (corresponding area: page 2*2 in on-chip data-retention RAM) 0: Writing to page 2 is disabled. 1: Writing to page 2 is enabled.
1	RRAMWE1	0	R/W	RAM Write Enable 1 (corresponding area: page 1*2 in on-chip data-retention RAM) 0: Writing to page 1 is disabled. 1: Writing to page 1 is enabled.
0	RRAMWE0	0	R/W	RAM Write Enable 0 (corresponding area: page 0*2 in on-chip data-retention RAM) 0: Writing to page 0 is disabled. 1: Writing to page 0 is enabled.

Note 1. For addresses in each page, see section 40, On-Chip RAM.

Note 2. When the VRAME0 bit in SYSCR1 is cleared to 0 (access to page 0 in large-capacity on-chip RAM is invalid), the on-chip data-retention RAM cannot be accessed (read and written), regardless of the setting of this bit.
When the VRAMWE0 bit in SYSCR2 is cleared to 0 (writing to page 0 in large-capacity on-chip RAM is invalid), the on-chip data-retention RAM cannot be written, regardless of the setting of this bit.

chanhvinh.com

42.2.18 CPU Status Register (CPUSTS)

CPUSTS is an 8-bit readable register that indicates the status of the CPU.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	ISBUSY	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0.
4	ISBUSY	0	R	State during Changing of the Frequency of CPU and Return from Software Standby Indicates that the frequency of CPU is being changed or that return from software standby is in progress. Do not execute a WFI instruction while this bit is 1. 0: The frequency of CPU is not being changed and returning from software standby is not in progress 1: The frequency of CPU is being changed or return from software standby is in progress
3 to 0	—	All 0	R	Reserved These bits are always read as 0.

chanhvinh.com

42.2.19 Standby Request Register 1 (STBREQ1)

This register is used to request notification of whether a CPU or peripheral module is ready for standby. The CPU or peripheral module returns a standby acknowledgement on receipt of a request for notification if it is ready for standby.

Note: • When writing to this register, see section 42.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	—	—	STBRQ 15	—	STBRQ 13*2	STBRQ 12*2	—	STBRQ 10
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R/W	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	STBRQ15	0	R/W	Standby Request to CoreSight 0: The standby request to CoreSight is invalid. 1: The standby request to CoreSight is valid.*1
4	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
3	STBRQ13*2	0	R/W	Standby Request to JPEG Codec Unit 0: The standby request to the JPEG codec unit is invalid. 1: The standby request to the JPEG codec unit is valid.*1
2	STBRQ12*2	0	R/W	Standby Request to EthernetAVB 0: The standby request to the EthernetAVB is invalid. 1: The standby request to the EthernetAVB is valid.*1
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	STBRQ10	0	R/W	Standby Request to Capture Engine Unit 0: The standby request to the capture engine unit is invalid. 1: The standby request to the capture engine unit is valid.*1

Note 1. When the MSTP bit for the corresponding module is 1, writing 1 to the STBRQ bit has no effect.
Note 2. This bit is only present in the RZ/A1LU. For the RZ/A1L and RZ/A1LC, the write value should always be 0.

chanhvinh.com

42.2.20 Standby Request Register 2 (STBREQ2)

This register is used to request notification of whether a peripheral module is ready for standby. The peripheral module returns a standby acknowledgement on receipt of a request for notification if it is ready for standby.

Note: • When writing to this register, see section 42.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	STBRQ 27*2	STBRQ 26	STBRQ 25	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	STBRQ27*2	0	R/W	Standby Request to the media local bus 0: The standby request to the media local bus is invalid. 1: The standby request to the media local bus is valid.*1
6	STBRQ26	0	R/W	Standby Request to Ethernet Controller 0: The standby request to the Ethernet controller is invalid. 1: The standby request to the Ethernet controller is valid.*1
5	STBRQ25	0	R	Standby Request to Video Display Controller 5 0: The standby request to the video display controller 5 is invalid. 1: The standby request to of the video display controller 5 is valid.*1
4 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note 1. When the MSTP bit for the corresponding module is 1, writing 1 to the STBRQ bit has no effect.
Note 2. This bit is only present in the RZ/A1L. For the RZ/A1LU and RZ/A1LC, the write value should always be 0.

chanhvinh.com

42.2.21 Standby Acknowledge Register 1 (STBACK1)

This register is used to provide notification that a CPU or peripheral module is in the standby ready state. The CPU or peripheral module returns a standby acknowledgement on reception of a standby notification request if it is ready for standby. This register is a read-only register.

Bit:	7	6	5	4	3	2	1	0
	—	—	STBAK 15	—	STBAK 13*2	STBAK 12*2	—	STBAK 10
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0.
5	STBAK15	0	R	Standby Acknowledgement from CoreSight 0: The standby acknowledgement from CoreSight is invalid. 1: The standby acknowledgement from CoreSight is valid.*1
4	—	0	R	Reserved This bit is always read as 0.
3	STBAK13*2	0	R	Standby Acknowledgement from JPEG Codec Unit 0: The standby acknowledgement from the JPEG codec unit is invalid. 1: The standby acknowledgement from the JPEG codec unit is valid.*1
2	STBAK12*2	0	R	Standby Acknowledgement from EthernetAVB 0: The standby acknowledgement from the EthernetAVB is invalid. 1: The standby acknowledgement from the EthernetAVB is valid.*1
1	—	0	R	Reserved This bit is always read as 0.
0	STBAK10	0	R	Standby Acknowledgement from Capture Engine Unit 0: The standby acknowledgement from the capture engine unit is invalid. 1: The standby acknowledgement from the capture engine unit is valid.*1

Note 1. The bits in STBACK1 are set to 1 when a standby acknowledgement is transmitted from the module while the MSTP bit for the corresponding module is set to 0.
Note 2. This bit is only present in the RZ/A1LU.

chanhvinh.com

42.2.22 Standby Acknowledge Register 2 (STBACK2)

This register is used to provide notification that a peripheral module is in the standby ready state. The peripheral module returns a standby acknowledgement on reception of a standby notification request if it is ready for standby. This register is a read-only register.

Bit:	7	6	5	4	3	2	1	0
	STBAK 27*2	STBAK 26	STBAK 25	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	STBAK27*2	0	R	Standby Acknowledgement from the media local bus 0: The standby acknowledgement from the media local bus is invalid. 1: The standby acknowledgement from the media local bus is valid.*1
6	STBAK26	0	R	Standby Acknowledgement from Ethernet Controller 0: The standby acknowledgement from the Ethernet controller is invalid. 1: The standby acknowledgement from the Ethernet controller is valid.*1
5	STBAK25	0	R	Standby Acknowledgement from Video Display Controller 5 0: The standby acknowledgement from the video display controller 5 is invalid. 1: The standby acknowledgement from the video display controller 5 is valid.*1
4 to 0	—	All 0	R	Reserved These bits are always read as 0.

Note 1. The bits in STBACK2 are set to 1 when a standby acknowledgement is transmitted from the module while the MSTP bit for the corresponding module is set to 0.
Note 2. This bit is only present in the RZ/A1L.

chanhvinh.com

42.2.23 On-Chip Data-Retention RAM Area Setting Register (RRAMKP)

RRAMKP is an 8-bit readable/writable register that selects whether the contents of the corresponding area of the on-chip data-retention RAM are retained or not in deep standby mode.

When the RRAMKP3 to RRAMKP0 bits are set to 1, the contents of the corresponding area of the on-chip data-retention RAM are retained in deep standby mode. When these bits are cleared to 0, the contents of the corresponding area of the on-chip data-retention RAM are not retained in deep standby mode.

Note: • When writing to this register, see section 42.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	RRAMKP3	RRAMKP2	RRAMKP1	RRAMKP0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	RRAMKP3	0	R/W	On-Chip Data-Retention RAM Storage Area 3 (corresponding area: page 3*1 in on-chip data-retention RAM) 0: The contents of the on-chip data-retention RAM are not retained in deep standby mode. 1: The contents of the on-chip data-retention RAM are retained in deep standby mode.
2	RRAMKP2	0	R/W	On-Chip Data-Retention RAM Storage Area 2 (corresponding area: page 2*1 in on-chip data-retention RAM) 0: The contents of the on-chip data-retention RAM are not retained in deep standby mode. 1: The contents of the on-chip data-retention RAM are retained in deep standby mode.
1	RRAMKP1	0	R/W	On-Chip Data-Retention RAM Storage Area 1 (corresponding area: page 1*1 in on-chip data-retention RAM) 0: The contents of the on-chip data-retention RAM are not retained in deep standby mode. 1: The contents of the on-chip data-retention RAM are retained in deep standby mode.
0	RRAMKP0	0	R/W	On-Chip Data-Retention RAM Storage Area 0 (corresponding area: page 0*1 in on-chip data-retention RAM) 0: The contents of the on-chip data-retention RAM are not retained in deep standby mode. 1: The contents of the on-chip data-retention RAM are retained in deep standby mode.

Note 1. For addresses in each page, see section 40, On-Chip RAM.

chanhvinh.com

42.2.24 Deep Standby Control Register (DSCTR)

DSCTR is an 8-bit readable/writable register that selects whether the states of the external memory control pins are retained or not when returning from deep standby mode and specifies the method to start the LSI.

Note: • When writing to this register, see section 42.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	EBUS KEEPE	RAM BOOT	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	EBUSKEEPE	0	R/W	Retention of External Memory Control Pin State 0: The state of the external memory control pins is not retained when returning from deep standby mode. 1: The state of the external memory control pins is retained when returning from deep standby mode. Note: The following setting is prohibited: (EBUSKEEPE, RAMBOOT) = (1, 0).
6	RAMBOOT	0	R/W	Selection of Method after Returning from Deep Standby Mode Selects an activation method after returning from deep standby mode. 0: Activated according to the boot mode specified for a reset. 1: The program is read from the on-chip data-retention RAM. Instruction fetch from H'20000000 Note: The following setting is prohibited: (EBUSKEEPE, RAMBOOT) = (1, 0).
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

42.2.25 Deep Standby Cancel Source Select Register (DSSSR)

DSSSR is a 16-bit readable/writable register that consists of the bits for selecting a source to cancel deep standby mode. The realtime clock alarm interrupt or change on the pins for canceling (P2_0, P2_2, P2_7, P2_9, P5_8, P5_9, P5_10, P6_7, P7_2, P7_3, P7_6, P7_9, NMI) can be selected as a cancel source. The pins for canceling can be used for canceling deep standby, regardless of pin function settings in the general I/O port.

Note: • When writing to this register, see section 42.4, Usage Notes.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	P2_0	P7_6	P7_9	P5_10	P2_2	P7_2	NMI	—	RTCAR	P5_9	P7_3	P5_8	P2_7	P2_9	P6_7
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	P2_0	0	R/W	Cancel by Change on P2_0 0: Deep standby mode is not canceled by change on the P2_0 pin. 1: Deep standby mode is canceled by change on the P2_0 pin.
13	P7_6	0	R/W	Cancel by Change on P7_6 0: Deep standby mode is not canceled by change on the P7_6 pin. 1: Deep standby mode is canceled by change on the P7_6 pin.
12	P7_9	0	R/W	Cancel by Change on P7_9 0: Deep standby mode is not canceled by change on the P7_9 pin. 1: Deep standby mode is canceled by change on the P7_9 pin.
11	P5_10	0	R/W	Cancel by Change on P5_10 0: Deep standby mode is not canceled by change on the P5_10 pin. 1: Deep standby mode is canceled by change on the P5_10 pin.
10	P2_2	0	R/W	Cancel by Change on P2_2 0: Deep standby mode is not canceled by change on the P2_2 pin. 1: Deep standby mode is canceled by change on the P2_2 pin.
9	P7_2	0	R/W	Cancel by Change on P7_2 0: Deep standby mode is not canceled by change on the P7_2 pin. 1: Deep standby mode is canceled by change on the P7_2 pin.
8	NMI	0	R/W	Cancel by Change on NMI 0: Deep standby mode is not canceled by change on the NMI pin. 1: Deep standby mode is canceled by change on the NMI pin.
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	RTCAR	0	R/W	Cancel by Realtime Clock Alarm Interrupt 0: Deep standby mode is not canceled by a realtime clock alarm interrupt. 1: Deep standby mode is canceled by a realtime clock alarm interrupt.
5	P5_9	0	R/W	Cancel by Change on P5_9 0: Deep standby mode is not canceled by change on the P5_9 pin. 1: Deep standby mode is canceled by change on the P5_9 pin.
4	P7_3	0	R/W	Cancel by Change on P7_3 0: Deep standby mode is not canceled by change on the P7_3 pin. 1: Deep standby mode is canceled by change on the P7_3 pin.
3	P5_8	0	R/W	Cancel by Change on P5_8 0: Deep standby mode is not canceled by change on the P5_8 pin. 1: Deep standby mode is canceled by change on the P5_8 pin.
2	P2_7	0	R/W	Cancel by Change on P2_7 0: Deep standby mode is not canceled by change on the P2_7 pin. 1: Deep standby mode is canceled by change on the P2_7 pin.
1	P2_9	0	R/W	Cancel by Change on P2_9 0: Deep standby mode is not canceled by change on the P2_9 pin. 1: Deep standby mode is canceled by change on the P2_9 pin.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
0	P6_7	0	R/W	Cancel by Change on P6_7 0: Deep standby mode is not canceled by change on the P6_7 pin. 1: Deep standby mode is canceled by change on the P6_7 pin.

chanhvinh.com

42.2.26 Deep Standby Cancel Edge Select Register (DSESR)

DSESR is a 16-bit readable/writable register that consists of the bits for selecting an edge to be detected for the pin specified as a deep standby cancel source with DSSSR. This register setting is always valid for canceling deep standby, regardless of the interrupt controller setting.

Note: • When writing to this register, see section 42.4, Usage Notes.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	P2_0E	P7_6E	P7_9E	P5_10E	P2_2E	P7_2E	NMIE	—	—	P5_9E	P7_3E	P5_8E	P2_7E	P2_9E	P6_7E
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	P2_0E	0	R/W	P2_0 Edge Detection 0: Falling edge of P2_0 is detected. 1: Rising edge of P2_0 is detected.
13	P7_6E	0	R/W	P7_6 Edge Detection 0: Falling edge of P7_6 is detected. 1: Rising edge of P7_6 is detected.
12	P7_9E	0	R/W	P7_9 Edge Detection 0: Falling edge of P7_9 is detected. 1: Rising edge of P7_9 is detected.
11	P5_10E	0	R/W	P5_10 Edge Detection 0: Falling edge of P5_10 is detected. 1: Rising edge of P5_10 is detected.
10	P2_2E	0	R/W	P2_2 Edge Detection 0: Falling edge of P2_2 is detected. 1: Rising edge of P2_2 is detected.
9	P7_2E	0	R/W	P7_2 Edge Detection 0: Falling edge of P7_2 is detected. 1: Rising edge of P7_2 is detected.
8	NMIE	0	R/W	NMI Edge Detection 0: Falling edge of NMI is detected. 1: Rising edge of NMI is detected.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	P5_9E	0	R/W	P5_9 Edge Detection 0: Falling edge of P5_9 is detected. 1: Rising edge of P5_9 is detected.
4	P7_3E	0	R/W	P7_3 Edge Detection 0: Falling edge of P7_3 is detected. 1: Rising edge of P7_3 is detected.
3	P5_8E	0	R/W	P5_8 Edge Detection 0: Falling edge of P5_8 is detected. 1: Rising edge of P5_8 is detected.
2	P2_7E	0	R/W	P2_7 Edge Detection 0: Falling edge of P2_7 is detected. 1: Rising edge of P2_7 is detected.
1	P2_9E	0	R/W	P2_9 Edge Detection 0: Falling edge of P2_9 is detected. 1: Rising edge of P2_9 is detected.
0	P6_7E	0	R/W	P6_7 Edge Detection 0: Falling edge of P6_7 is detected. 1: Rising edge of P6_7 is detected.

chanhvinh.com

42.2.27 Deep Standby Cancel Source Flag Register (DSFR)

DSFR is a 16-bit readable/writable register composed of two types of bits. One is the flag that confirms which source canceled deep standby mode. The other is the bit that releases the state of pins after canceling deep standby mode. When deep standby mode is canceled by an interrupt (NMI or realtime clock alarm interrupt) and changes on the pins for canceling, this register retains the previous data although power-on reset exception handling is executed. When deep standby mode is canceled by a power-on reset, this register is initialized to H'0000.
All flags must be cleared immediately before transition to deep standby mode.

Note: • When writing to this register, see section 42.4, Usage Notes.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IO KEEP	P2_0F	P7_6F	P7_9F	P5_10F	P2_2F	P7_2F	NMIF	-	RTC ARF	P5_9F	P7_3F	P5_8F	P2_7F	P2_9F	P6_7F
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written after reading 1 to clear the flag.

Bit	Bit Name	Initial Value	R/W	Description
15	IOKEEP	0	R/(W)*1	Release of Pin State Retention Releases the retention of the pin state after canceling deep standby mode 0: Pin state not retained [Clearing condition] • Writing 0 after reading 1 1: Pin state retained [Setting condition] • When deep standby mode is entered
14	P2_0F	0	R/(W)*1	P2_0 Flag 0: No change on the P2_0 pin 1: Change on the P2_0 pin
13	P7_6F	0	R/(W)*1	P7_6 Flag 0: No change on the P7_6 pin 1: Change on the P7_6 pin
12	P7_9F	0	R/(W)*1	P7_9 Flag 0: No change on the P7_9 pin 1: Change on the P7_9 pin
11	P5_10F	0	R/(W)*1	P5_10 Flag 0: No change on the P5_10 pin 1: Change on the P5_10 pin
10	P2_2F	0	R/(W)*1	P2_2 Flag 0: No change on the P2_2 pin 1: Change on the P2_2 pin
9	P7_2F	0	R/(W)*1	P7_2 Flag 0: No change on the P7_2 pin 1: Change on the P7_2 pin
8	NMIF	0	R/(W)*1	NMI Flag 0: No interrupt on NMI pin 1: Interrupt on NMI pin
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	RTCARF	0	R/(W)*1	RTCAR Flag 0: No realtime clock alarm interrupt generated 1: Realtime clock alarm Interrupt generated
5	P5_9F	0	R/(W)*1	P5_9 Flag 0: No change on the P5_9 pin 1: Change on the P5_9 pin

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
4	P7_3F	0	R/(W)*1	P7_3 Flag 0: No change on the P7_3 pin 1: Change on the P7_3 pin
3	P5_8F	0	R/(W)*1	P5_8 Flag 0: No change on the P5_8 pin 1: Change on the P5_8 pin
2	P2_7F	0	R/(W)*1	P2_7 Flag 0: No change on the P2_7 pin 1: Change on the P2_7 pin
1	P2_9F	0	R/(W)*1	P2_9 Flag 0: No change on the P2_9 pin 1: Change on the P2_9 pin
0	P6_7F	0	R/(W)*1	P6_7 Flag 0: No change on the P6_7 pin 1: Change on the P6_7 pin

Note 1. Only 0 can be written after reading 1 to clear the flag.

42.2.28 XTAL Crystal Oscillator Gain Control Register (XTALCTR)

XTALCTR is an 8-bit readable/writable register that controls the gain of the crystal oscillator for XTAL.
If the realtime clock uses the EXTAL input, the GAIN0 bit retains the previous value when software standby mode or deep standby mode is canceled by a source other than a power-on reset. If the realtime clock does not use the EXTAL input, this bit is initialized to 0 when software standby or deep standby mode is entered.
This register is initialized to H'00 when software standby or deep standby mode is canceled by a power-on reset.

Note: • When writing to this register, see section 42.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GAIN0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GAIN0	0	R/W	XTAL Crystal Oscillator (EXTAL or XTAL Pin) Gain Select 0: Large gain 1: Small gain

chanhvinh.com

42.3 Operation

42.3.1 Sleep Mode

(1) Transition to Sleep Mode

When causing the device to make a transition from the program execution state to sleep mode, follow procedure (a) or (b) depending on the condition described following the description of the procedures.

(a) Execute the WFI instruction while the STBY bit in STBCR1 is 0.
(b) While the STBY bit in STBCR1 is 0 and bits I and F in the current program status register (CPSR) corresponding to the interrupt source to trigger release from the sleep mode are 0 (mask disabled), execute the following three instructions in the given sequence: SEV, WFE, and WFE.

When the following condition is met, follow procedure (b) and do not use the WFI instruction other than to initiate a transition to software standby or deep standby.

- Software standby is also being used.
- When software standby is not being used, deep standby is being used, and the procedure (a) in 42.3.4 (1) Transition to Deep Standby Mode is followed to initiate the transition to deep standby.

Although the CPU halts immediately after a transition to the sleep mode, the contents of its internal registers remain unchanged. The on-chip peripheral modules continue to run. The clock output from the CKIO pin is continued.

Note: • When writing to STBCR1 register, see section 42.4, Usage Notes.

(2) Canceling Sleep Mode

Sleep mode is canceled by an interrupt (NMI, IRQ, and on-chip peripheral module) or a power-on reset.

- Canceling by an interrupt
  When an NMI, IRQ, or on-chip peripheral module interrupt occurs, sleep mode is canceled and interrupt exception handling is executed. When the priority level of the generated interrupt is lower than the setting of the interrupt controller execution priority register, or the interrupt by the on-chip peripheral module is disabled on the module side, the interrupt request is not accepted and sleep mode is not canceled. When procedure (a) in 42.3.1 (1) Transition to Sleep Mode is followed and an interrupt request occurs with an interrupt masked by bits 6 (the FIQ mask bit) and 7 (the IRQ mask bit) in the CPSR register in the CPU, sleep mode is canceled but interrupt exception handling is not executed.
- Canceling by a reset
  Sleep mode is canceled by a power-on reset.

chanhvinh.com

42.3.2 Software Standby Mode

(1) Transition to Software Standby Mode

The LSI switches from a program execution state to software standby mode by executing the WFI instruction when the STBY bit and DEEP bit in STBCR1 are 1 and 0 respectively. In software standby mode, not only the CPU but also the clock and on-chip peripheral modules halt. The clock output from the CKIO pin also stops.

The contents of the CPU and cache registers remain unchanged. Some registers of on-chip peripheral modules are, however, initialized. As for the states of on-chip peripheral module registers in software standby mode, see section 46.3, Register States.

The CPU takes one cycle to finish writing to STBCR1, and then executes processing for the next instruction. However, it takes one or more cycles to actually write. Therefore, execute a WFI instruction after reading STBCR1 to have the values written to STBCR1 by the CPU to be definitely reflected in the WFI instruction.

After a WFI instruction is issued, the hardware automatically stops the bus master following the wait for completion of the issuing-finished request from the bus master and the transition to software standby proceeds. Since the transition to software standby is not possible if completion of the issuing-finished request is not possible at this time, do not proceed with access to the registers of modules in the module-standby state and so on. Furthermore, as the issuing of unintended requests by the bus master is inhibited, using software to stop all bus masters in preparation for the procedure for the transition to software standby is also effective.

The procedure for switching to software standby mode is as follows:

1. Set the standby_mode_en bit of the power control register in the PL310 to 1. For the details of this register, see CoreLink Level 2 Cache Controller L2C-310 Technical Reference Manual issued by Arm Ltd.
2. Clear the TME bit in the watchdog timer control/status register (WTCSR) of the watchdog timer to 0 to stop the watchdog timer.
3. Set the CKS[2:0] bits of the WTCSR register and the watchdog timer counter (WTCNT) to appropriate values so that the watchdog timer overflow period will be at least the oscillation settling time on return from standby.
4. After setting the STBY and DEEP bits in STBCR1 to 1 and 0 respectively, read STBCR1. Then, execute a WFI instruction.

chanhvinh.com

(2) Canceling Software Standby Mode

Software standby mode is canceled by interrupts (NMI or IRQ) or a reset (power-on reset). Clock signal starts to be output from the CKIO pin.

- Canceling by an interrupt
  When the falling edge or rising edge of the NMI pin (selected by the NMI edge select bit (NMIE) in the interrupt control register 0 (ICR0) of the interrupt controller) or the falling edge or rising edge of an IRQ pin (IRQ7 to IRQ0) (selected by the IRQn sense select bits (IRQn1S and IRQn0S) in interrupt control register 1 (ICR1) of the interrupt controller) is detected, clock oscillation is started. This clock pulse is supplied only to the oscillation settling counter (watchdog timer) used to count the oscillation settling time.
  After the elapse of the time set in the clock select bits (CKS[2:0]) in the watchdog timer control/status register (WTCSR) of the watchdog timer before the transition to software standby mode, the watchdog timer overflow occurs. Since this overflow indicates that the clock has been stabilized, the clock pulse will be supplied to the entire chip after this overflow. Software standby mode is thus cleared and NMI interrupt exception handling (IRQ interrupt exception handling in case of IRQ) is started. If the priority level of the generated interrupt is lower than the setting of the interrupt controller execution priority register, the interrupt request is not accepted and sleep mode is entered. So, cancel software standby mode by an interrupt whose priority level is higher than the setting of the interrupt controller execution priority register.
  When canceling software standby mode by the NMI interrupt or IRQ interrupt, set the CKS[2:0] bits so that the watchdog timer overflow period will be equal to or longer than the oscillation settling time.
  The clock output phase of the CKIO pin may be unstable immediately after detecting an interrupt and until software standby mode is canceled.
- Canceling by a reset
  When the $\overline{\text{RES}}$ pin is driven low, software standby mode is canceled and the LSI enters the power-on reset state. After that, if the $\overline{\text{RES}}$ pin is driven high, the power-on reset exception handling is started.
  Keep the $\overline{\text{RES}}$ pin low until the clock oscillation settles. The internal clock will continue to be output to the CKIO pin.

(3) Note on Transition to Software Standby Mode

Release from software standby mode is triggered by interrupts (NMI or IRQ) or a power-on reset. If, however, a WFI instruction and an interrupt other than NMI and IRQ are generated at the same time, software standby mode may be canceled due to acceptance of the interrupt.
When initiating a transition to software standby mode, make settings so that interrupts other than NMI and IRQ are not generated before execution of the WFI instruction.

(4) Note on Canceling Software Standby Mode

After software standby mode is canceled, unstable clock pulses are output from the CKIO pin during oscillation settling time. To prevent malfunction due to the unstable pulses, the CKOEN[1:0] bits in FRQCR should be modified.

chanhvinh.com

42.3.3 Software Standby Mode Application Example

This example describes a transition to software standby mode on the falling edge of the NMI signal, and cancellation on the rising edge of the NMI signal. The timing is shown in Figure 42.2.

When the NMI pin is changed from high to low level while the NMI edge select bit (NMIE) in the interrupt control register 0 (ICR0) is set to 0 (falling edge detection), the NMI interrupt is accepted. When the NMIE bit is set to 1 (rising edge detection) by the NMI exception service routine, the STBY and DEEP bits in STBCR1 are set to 1 and 0 respectively, and a WFI instruction is executed, software standby mode is entered. Thereafter, software standby mode is canceled when the NMI pin is changed from low to high level.

Note: • When writing to STBCR1 register, see section 42.4, Usage Notes.

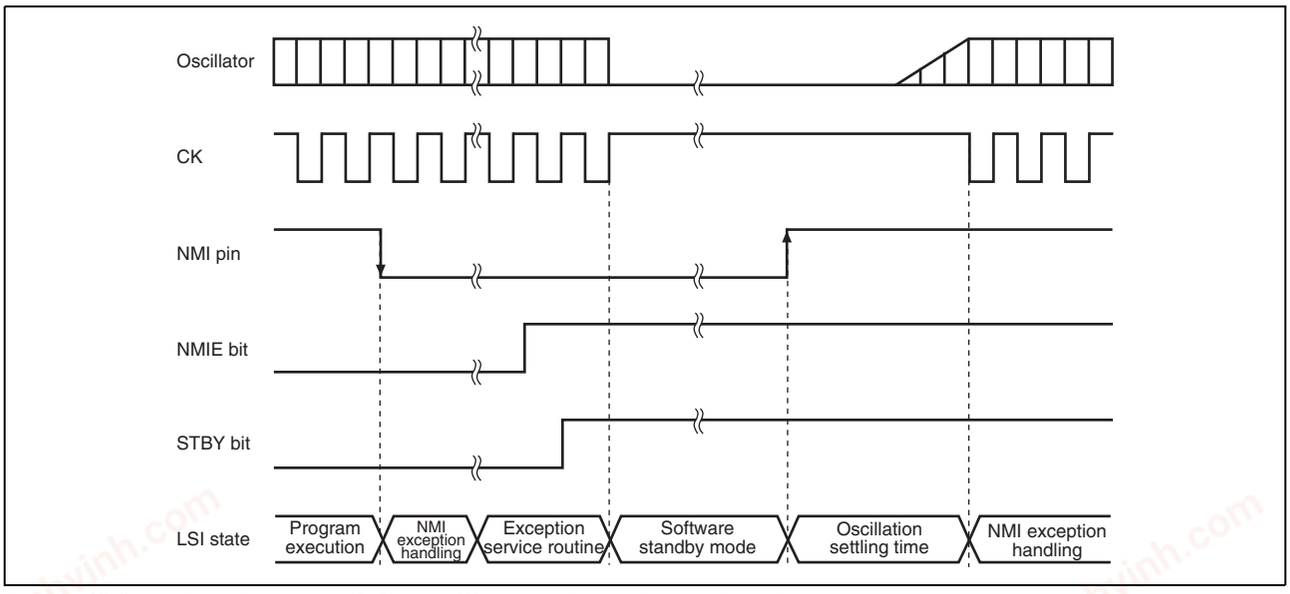


Figure 42.2 NMI Timing in Software Standby Mode (Application Example)

42.3.4 Deep Standby Mode

(1) Transition to Deep Standby Mode

The LSI switches from a program execution state to deep standby mode by executing the WFI instruction when the STBY bit and DEEP bit in STBCR1 are set to 1. In deep standby mode, not only the CPU, clocks, and on-chip peripheral modules but also power supply is turned off excluding the on-chip data-retention RAM area specified by the RRAMKP3 to RRAMKP0 bits in RRAMKP and realtime clock. This can significantly reduce power consumption. Therefore, data in the registers of the CPU, cache, and on-chip peripheral modules are not retained. Pin state values immediately before the transition to deep standby mode are retained.

The CPU takes one cycle to finish writing to STBCR1, and then executes processing for the next instruction. However, it actually takes one or more cycles to write. Therefore, execute a WFI instruction after reading STBCR1 to reflect the values written to STBCR1 by the CPU in the WFI instruction without fail.

After a WFI instruction is issued, the hardware automatically stops the bus master following the wait for completion of the issuing-finished request from the bus master and the transition to deep standby proceeds. Since the transition to deep standby is not possible if completion of the issuing-finished request is not possible at this time, do not proceed with access to the registers of modules in the module-standby state and so on. Furthermore, as the issuing of unintended requests by the bus master is inhibited, using software to stop all bus masters in preparation for the procedure for the transition to deep standby is also effective.

chanhvinh.com

The procedure for switching to deep standby mode is as follows. Figure 42.3 also shows its flowchart.

1. Set the standby_mode_en bit of the power control register in the PL310 to 1. For the details of this register, see CoreLink Level 2 Cache Controller L2C-310 Technical Reference Manual issued by Arm Ltd.
2. Set the RRAMKP3 to RRAMKP0 bits in RRAMKP for the corresponding on-chip data-retention RAM area that must be retained. Transfer the programs to be retained to the specified areas of the on-chip data-retention RAM.
3. Set the RAMBOOT and EBUSKEEPE bits in DSCTR to specify the activation method for returning from deep standby mode and to select whether the external memory control pin status is retained or not.
4. When canceling deep standby mode by an interrupt, set the corresponding bit in DSSSR to select the pin or source to cancel deep standby mode. In this case, specify the input signal detection mode for the selected pin with the corresponding bit in DSESR.
5. Execute read and write of an arbitrary but the same address for each page in the on-chip data-retention RAM area. When this is not executed, data last written may not be written to the on-chip data-retention RAM. If there is a write to the on-chip data-retention RAM after this time, execute this processing after the last write to the on-chip data-retention RAM.

The procedures for steps 6 to 8 are as listed under (a) or (b), and which procedure is correct depends on the condition described after the procedures.

(a)

6. Set the STBY and DEEP bits in the STBCR1 register to 1, and then read this register.
7. Clear the flag in the DSFR register.
8. Set the CPU interface control register (ICCICR) of the interrupt controller to 0 so that the CPU is not notified of interrupts other than NMIs. Then, read the ICCICR register.

(b)

6. Clear the flag in the DSFR register.
7. Set the CPU interface control register (ICCICR) of the interrupt controller to 0 so that the CPU is not notified of interrupts other than NMIs. Then, read the ICCICR register.
8. Set the STBY and DEEP bits in the STBCR1 register to 1, and then read this register.

When the condition below is met, follow procedure (b).

- Software standby is not being used, the sleep mode is being used, and procedure (a) in section 42.3.1, (1) Transition to Sleep Mode is followed to initiate the transition to sleep mode.

9. Execute the WFI instruction.
10. The conflict between the instruction and the NMI interrupt may prevent transition to deep standby mode. After executing the WFI instruction, locate the branch instruction to return to step 9.

chanhvinh.com

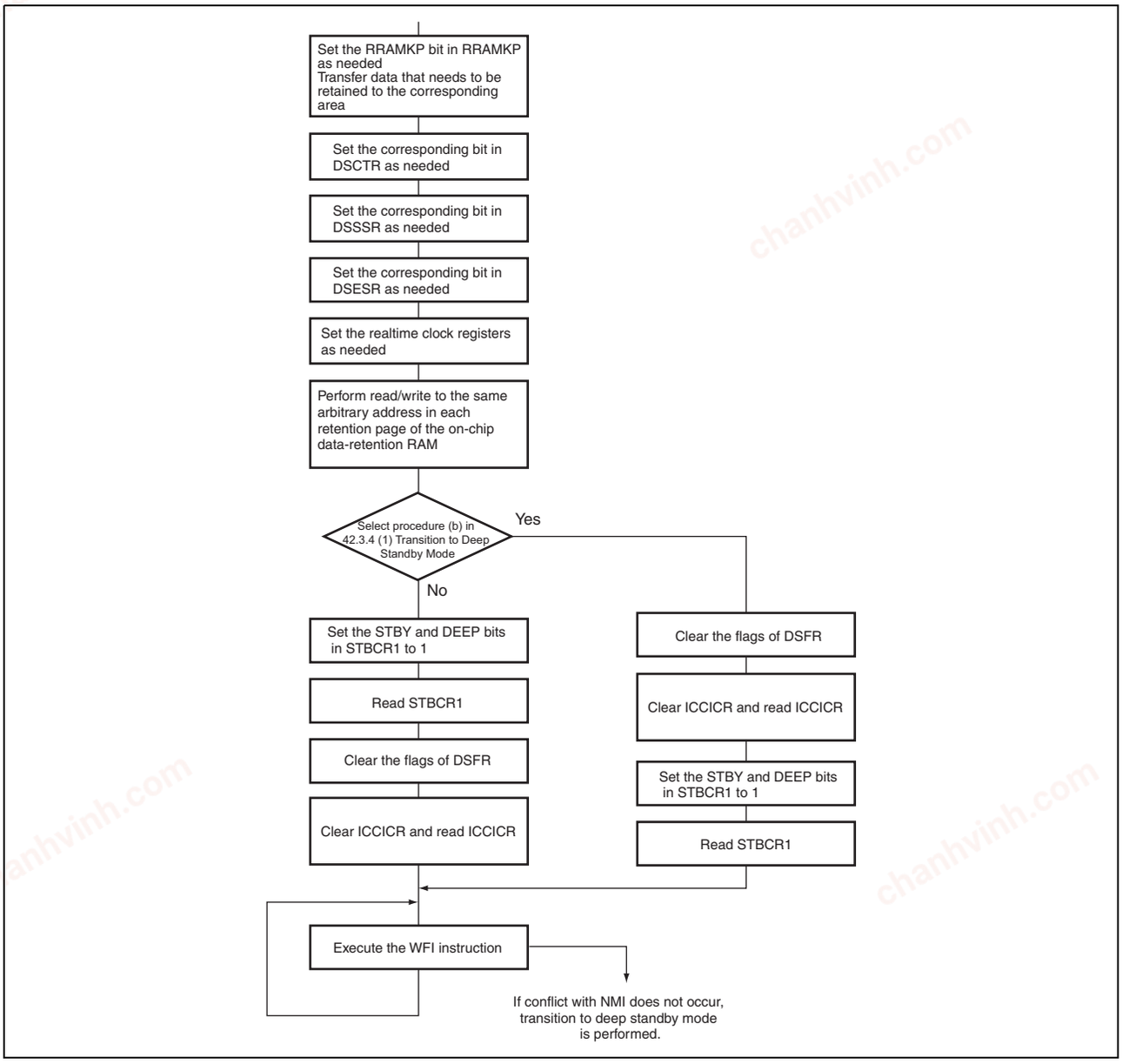


Figure 42.3 Flowchart of Transition to Deep Standby Mode

chanhvinh.com

(2) Canceling Deep Standby Mode

Deep standby mode is canceled by interrupts (NMI or realtime clock alarm interrupt), change on the pins for canceling, or a reset (power-on reset). The realtime clock alarm interrupt can always cancel deep standby mode regardless of the settings of the execution priority register for the interrupt controller and of the alarm interrupt enable flag (RCR1.AIE) . This interrupt can also always cancel this mode do regardless of the states of bit 6 (the FIQ mask bit) and bit 7 (the IRQ mask bit) in the CPSR register of the CPU. When canceling the mode by a source other than a reset, a power-on reset exception handling is executed instead of an interrupt exception handling.
Figure 42.4 shows the flowchart of canceling deep standby mode.

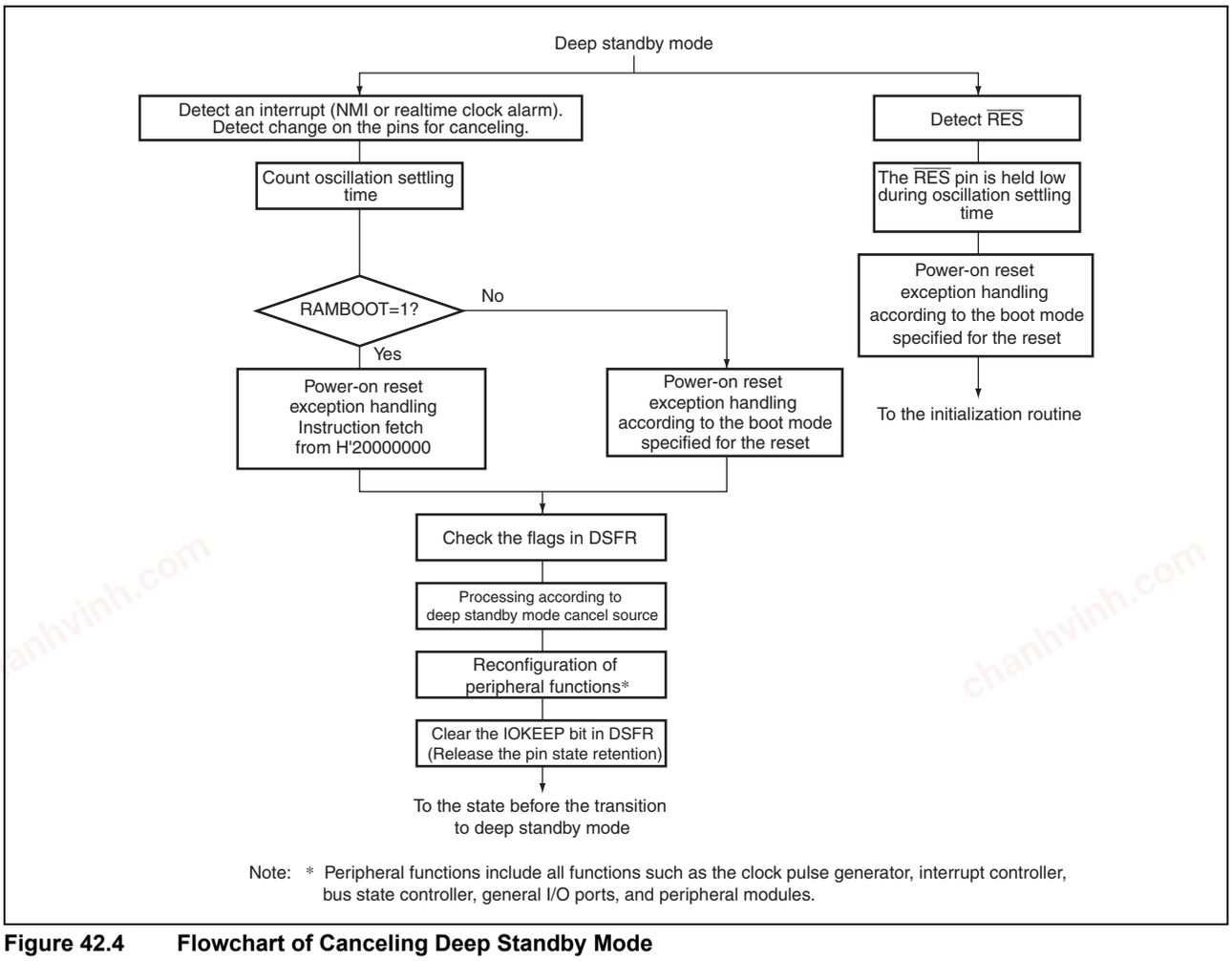


Figure 42.4 Flowchart of Canceling Deep Standby Mode

- Canceling by a source other than a reset
  When the falling or rising edge of the NMI pin (selected by a corresponding bit in DSESR) or falling or rising edge of the pins for canceling (selected by a corresponding bit in DSESR) is detected or the realtime clock alarm interrupt (see section 13.4.4, Alarm Function) is generated, clock oscillation is started after the wait time for the oscillation settling time. After the oscillation settling time has elapsed, deep standby mode is cancelled and the power-on reset exception handling is executed.
  The clock output phase of the CKIO pin may be unstable immediately after detecting a cancel source and until deep standby mode is canceled.
  The detecting of the NMI pin, the pins for canceling, and the realtime clock alarm interrupt becomes enable when the corresponding bits in DSSSR are set. The detected cancel sources are kept, but they are reflected to DSFR after canceling the deep standby mode. When the CPU accepts any interrupts and reads the interrupt response register (ICCIAR), all of the cancel sources that are kept are cleared. When the CPU enters the deep standby mode as the

chanhvinh.com

detected cancel sources are kept, the deep standby mode is canceled immediately after the CPU enters the deep standby mode.

- Canceling with a reset
  Driving the $\overline{\text{RES}}$ pin low cancels deep standby mode and causes a transition to the power-on reset state. After this, driving the $\overline{\text{RES}}$ pin high initiates power-on reset exception handling. Output of the internal clock from the CKIO pin also starts by driving the $\overline{\text{RES}}$ pin low.
  Keep the $\overline{\text{RES}}$ pin low until the clock oscillation has settled.

(3) Operation after Canceling Deep Standby Mode

After canceling deep standby mode, the LSI can be activated through the external memory or from the on-chip data-retention RAM, which can be selected by setting the RAMBOOT bit in DSCTR. By setting the EBUSKEEPE bit, the states of the external memory control pins can be retained even after cancellation of deep standby mode. Table 42.3 shows the pin states after cancellation of deep standby mode according to the setting of each bit. Table 42.4 lists the external memory control pins.

Table 42.3 Pin States after Cancellation of Deep Standby Mode and System Activation Method by the DSCTR Settings

EBUSKEEPE Bit	RAMBOOT Bit	Activation Method	Pin States After Cancellation of Deep Standby Mode
0	0	External memory	The states of the external memory control pins are not retained. For other pins, the retention of their states is cancelled when the IOKEEP bit is cleared.
0	1	On-chip data-retention RAM	The states of the external memory control pins are not retained. After cancellation of deep standby mode, the retention of the external memory control pin states is cancelled. For other pins, the retention of their states is cancelled when the IOKEEP bit is cleared.
1	0	—	Setting prohibited.
1	1	On-chip data-retention RAM	The states of the external memory control pin are retained. The retention of the states of the external memory control pins and other pins is cancelled when the IOKEEP bit is cleared.

Table 42.4 External Memory Control Pins in Different Modes

Boot Mode 0 (CS0 Area: Bus Width: 16 Bits)	Boot Mode 1 (Serial Flash Boot)	Boot Mode 2 (eSD Boot)	Boot Mode 3 (eMMC Boot)
A[20:1] D[15:0] $\overline{\text{CS0}}$, $\overline{\text{RD}}$, CKIO	SPBCLK_0, SPBSSL_0, SPBMO0_0, SPBMI0_0	SD_CLK_0, SD_CMD_0, SD_D[3:0]_0	MMC_CLK, MMC_CMD, MMC_D[3:0]

When deep standby mode is canceled by interrupts (NMI or realtime clock alarm) or changes on the pins for canceling, the deep standby cancel source flag register (DSFR) can be used to confirm which source has canceled the mode.
Pins retain the state immediately before the transition to deep standby mode. However, in system activation through the external memory, the retention of the states of the external memory control pins is cancelled so that programs can be fetched after cancellation of deep standby mode. Other pins, after cancellation of deep standby mode, continue to retain the pin states until writing 0 to the IOKEEP bit in DSFR after reading 1 from the same bit. In system activation from the on-chip data-retention RAM, after cancellation of deep standby mode, both the external memory control pins and other pins continues to retain the pin states until writing 0 to the IOKEEP bit in DSFR after reading 1 from the same bit.
Reconfiguration of peripheral functions is required to return to the previous state of deep standby mode. Peripheral functions include all functions such as the clock pulse generator, interrupt controller, general I/O ports, and peripheral modules. After the reconfiguration, the retention of the pin state can be canceled and the LSI returns to the state prior to the transition to deep standby mode by reading 1 from the IOKEEP bit in DSFR and then writing 0 to it.

chanhvinh.com

(4) Notes on Transition to Deep Standby Mode

If multiple canceling sources have been specified and multiple canceling sources are input, multiple cancel source flags will be set.

42.3.5 Module Standby Function

(1) Transition to Module Standby Function

Setting an MSTP bit in a standby control register to 1 halts supply of the clock signal to the corresponding on-chip peripheral module. This function can be used to reduce power consumption both while a program is running and in sleep mode. Disable a module before placing it in the module standby state. If interrupt requests are enabled, set the respective register in the target module or in the interrupt controller to disable interrupt requests. If DMA transfer requests are enabled, set the respective register in the target module to disable DMA transfer requests from the module and then set the respective register in the direct memory access controller to stop DMA transfers. In addition, do not attempt to access a module's registers while it is in the module standby state. The procedure for transitions to the module standby state depends on whether the STBREQ register has a bit for the corresponding module.

(a) Procedure for transition to module standby of modules for which the STBREQ register does not have a corresponding bit

1. Set the MSTP bit of the corresponding module to 1.

(b) Procedure for transition to module standby of modules for which the STBREQ register has a corresponding bit

1. Set the corresponding bit in the STBREQ register to 1 to generate a request to stop the module.
2. Confirm that the module is ready to be stopped by the corresponding bit in the STBACK register being set to 1.
3. Set the MSTP bit of the corresponding module to 1.

For states of registers in the module standby function, see section 46.3, Register States.

(2) Canceling Module Standby Function

Release from the module standby state can be achieved in two ways: starting the module by a power-on reset while the MSTP bit is set to 1, then clearing the MSTP bit to 0, or setting the MSTP bit to 0 to activate the module, setting the MSTP bit to 1 to place the module in the standby state, and then setting the MSTP bit to 0 again. If you use the latter approach, the procedure for release from module standby depends on whether the STBREQ register has a bit for the corresponding module.

- Release from the module standby state after the activation of the module by a power-on reset while the MSTP bit is set to 1

1. Clear the MSTP bit to 0.
2. After that, dummy-read the same register.

- Release from the module standby state after a transition to standby following activation of the module

(a) Procedure for release from module standby of modules for which the STBREQ register does not have a corresponding bit

1. Clear the MSTP bit to 0, then dummy-read the same register.

(b) Procedure for release from module standby of modules for which the STBREQ register has a corresponding bit

1. Clear the MSTP bit to 0, then dummy-read the same register.
2. Clear the corresponding bit in the STBREQ register to 0 to cancel the request to stop the module.
3. Confirm that the corresponding bit in the STBACK register has been cleared to 0, indicating the completion of release from standby.

chanhvinh.com

42.3.6 Software Reset

Initialization equivalent to a power-on reset is only applied to the selected modules. The procedure for transitions to the software reset state varies according to whether the STBREQ register has a bit for the corresponding module. Follow the procedures below according to whether this is or is not the case. If DMA transfer requests are enabled, make the settings to stop the DMA transaction for the corresponding channel before transitions to the software reset state. For stopping the DMA transaction, see section 9.7.11, Transfer Status.

(1) Transition to Software Reset State

(a) Procedure for transition to the software reset state of modules for which the STBREQ register does not have a corresponding bit

1. Set the SRST bit of the corresponding module to 1, then dummy-read the same register.

(b) Procedure for transition to the software reset state of modules for which the STBREQ register has a corresponding bit

1. Set the corresponding bit in the STBREQ register to 1 to generate a request to stop the module.
2. Confirm that the corresponding bit in the STBACK register has been set to 1.
3. Set the SRST bit of the corresponding module to 1, then dummy-read the same register.

(2) Canceling Software Reset

(a) Procedure for release from the software reset state of modules for which the STBREQ register does not have a corresponding bit

1. Clear the SRST bit of the corresponding module to 0, then dummy-read the same register.

(b) Procedure for release from the software reset state of modules for which the STBREQ register has a corresponding bit

1. Clear the SRST bit of the corresponding module to 0, then dummy-read the same register.
2. Clear the corresponding bit in the STBREQ register to 0 to cancel the request to stop the module.
3. Confirm that the corresponding bit in the STBACK register has been cleared to 0.

42.3.7 Adjustment of XTAL Crystal Oscillator Gain

The gain of the crystal oscillator for XTAL can be adjusted using the GAIN0 bit in XTALCTR. To modify the gain of the signal on the EXTAL or XTAL pin, PLL settling time is needed. The settling time is counted using the on-chip watchdog timer.

After a change to the value of the GAIN0 bit, the hardware automatically stops the bus master following the wait for completion of the issuing-finished request from the bus master and the change to the gain for the crystal oscillator in use with XTAL proceeds. Since the change to the gain for the crystal oscillator in use with XTAL cannot start if completion of the issuing-finished request is not possible at this time, do not proceed with access to the registers of modules in the module-standby state and so on. Furthermore, as the issuing of unintended requests by the bus master is inhibited, using software to stop all bus masters in preparation for the procedure to change the gain for the crystal oscillator in use with XTAL is also effective.

1. The large gain is selected in the initial state.
2. Set the standby_mode_en bit of the power control register in the PL310 to 1. For the details of this register, see CoreLink Level 2 Cache Controller L2C-310 Technical Reference Manual issued by Arm Ltd.
3. Set the watchdog timer so that the specified settling time should be obtained and stop the watchdog timer.
   Specifically, the following settings are necessary:
   TME in WTCSR = 0: Stop the watchdog timer.
   CKS[2:0] in WTCSR: Division ratio for watchdog timer count clock
   WTCNT: Initial counter value
   (The watchdog timer starts counting on the set clock.)

chanhvinh.com

4. Set the GAIN0 bit to the desired value.
5. The LSI is internally stopped and the watchdog timer starts counting. The clock is supplied only to the watchdog timer and other internal clocks are stopped. In this state, the CKIO pin continues to output an unstable clock. To avoid malfunction due to the unstable clock, modify the CKOEN2 bit in FRQCR appropriately. Since this state is equivalent to the software standby mode state, some registers of on-chip peripheral modules are initialized. For details, see section 46.3, Register States.
6. When an overflow occurs on the watchdog timer, the specified clock supply is started and the LSI starts operation. The watchdog timer stops after an overflow.

chanhvinh.com

42.4 Usage Notes

42.4.1 Usage Notes on Setting Registers

When writing to the registers related to power-down modes, note the following.
When writing to the register related to power-down modes, the CPU, after executing a write instruction, executes the next instruction without waiting for the write operation to complete.
Therefore, to reflect the change specified by writing to the register while the next instruction is executed, insert a dummy read of the same register between the register write instruction and the next instruction.

42.4.2 Usage Notes when the Realtime Clock is not Used

When the realtime clock is not used, set the MSTP60 bit in STBCR6 to 1 after setting the bits in registers of the realtime clock. For details, see section 42.2.6, Standby Control Register 6 (STBCR6).

- Set the RTCEN bit in the control register 2 (RCR2) to 0
- Set the RCKSEL[1:0] bits in the control register 5 (RCR5) to 00.

42.4.3 Usage Notes Applying when the USB_X1 Pin is not to be Used

When a 48 MHz clock is not being supplied to the USB_X1 pin, set registers according to the below procedure in the initial settings after release from the power-on reset state or deep standby mode.

(1) When the USB2.0 host/function module is not to be used

1. Set the MSTP71 bit in the STBCR7 register to 0 and dummy-read STBCR7.
2. Set the UCKSEL bit in the SYSCFG0_0 register to 1.
3. Wait for at least 20 cycles of the EXTAL clock.
4. Set the MSTP71 bit in the STBCR7 register to 1 and dummy-read STBCR7.

(2) When the USB2.0 host/function module is to be used

1. Set the MSTP71 bit* in the STBCR7 register to 0 and dummy-read STBCR7.
2. Set the UCKSEL bit in the SYSCFG0_0 register to 1.
3. Follow the procedure of example 1 in section 28.4.1 (5) Setting the Clock Supply for the USB Module.
4. Make the initial settings of the USB2.0 host/function module.

Note: * When channel 1 of the USB2.0 host/function module is to be used, also set the MSTP70 bit to 0.

42.4.4 Notes on Using IRQ Pins as Triggers for Release from Standby when Software Standby is in Use

To use an IRQ pin as the trigger for release from standby when software standby is in use, see section 7.8.4, Notes on Using IRQ Pins as Triggers for Release from Standby when Software Standby is in Use.

chanhvinh.com

43. Debugger Interface

This LSI incorporates a debugger interface to support the boundary scan function and connection to the emulator.

43.1 Features

The debugger interface is a serial input/output interface which has a JTAG (Joint Test Action Group, IEEE Std.1149.1 and IEEE Standard Test Access Port and Boundary-Scan Architecture) interface and CoreSight debug interface.

This module has TAP controllers for the boundary scan and CoreSight debug function.

Driving the BSCANP pin high selects the boundary-scan TAP controller, and driving the BSCANP pin low selects the CoreSight debug TAP controller.

Figure 43.1 is a block diagram of this module and Table 43.1 shows the JTAG pin mode.

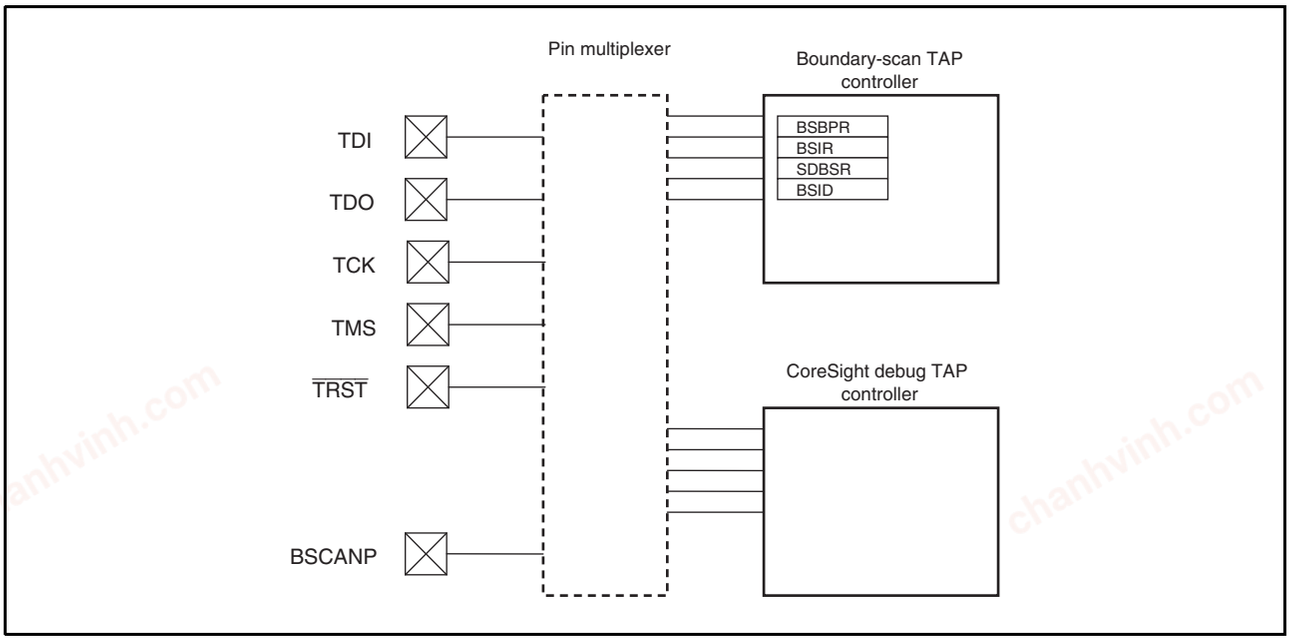


Figure 43.1 Block Diagram

Table 43.1 JTAG Pin Mode

BSCANP	JTAG Pin Mode
0	Normal operation (CoreSight debug mode)
1	Boundary-scan mode

The following lists the features of CoreSight.

- JTAG interface
  Supports JTAG and serial wire debug mode (SWD)
- Trace interface
  Outputs 4-bit × 66-Mbps (33-MHz DDR) trace data
  4-Kbyte Embedded Trace FIFO (ETF)
- Controls by ICE registers
  Control of resetting, debug-enable signal supplied to the CPU, and disabling power shut-off in deep-standby mode during debugging (fake debug mode)

chanhvinh.com

Figure 43.2 is a block diagram of CoreSight, Table 43.2 to Table 43.5 show the input/output pins of the cross-trigger interface (CTI), and Table 43.6 shows the address map for CoreSight. For details of CoreSight other than ICE registers, see the technical reference manual issued by Arm Ltd.

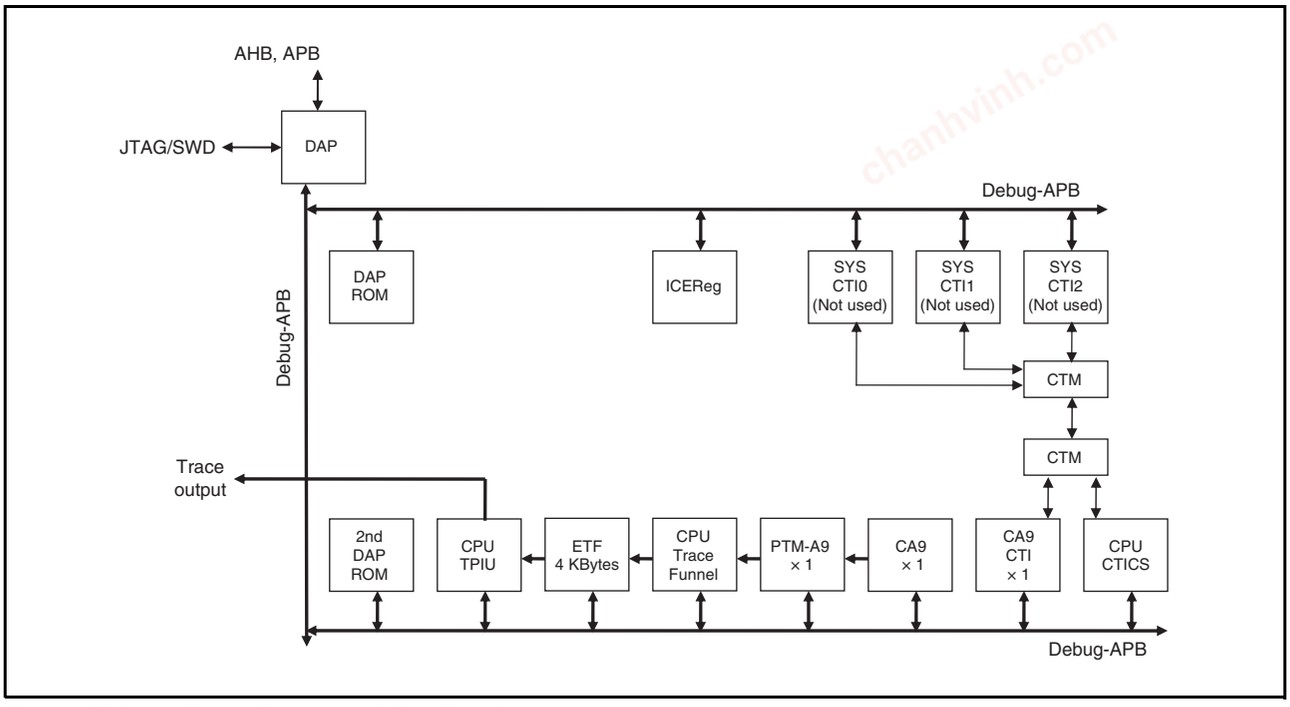


Figure 43.2 Block Diagram of CoreSight

Table 43.2 CA9 CTI Trigger Inputs

Trigger Input Bit	Trigger Signal	Source Device
[7]	Not used	—
[6]	TRIGGER	PTM-A9
[5]	COMMRX	CA9
[4]	COMMTX	CA9
[3]	EXTOUT[1]	PTM-A9
[2]	EXTOUT[0]	PTM-A9
[1]	PMUIRQ	CA9
[0]	DBGACK	CA9

Table 43.3 CA9 CTI Trigger Outputs

Trigger Input Bit	Trigger Signal	Destination Device
[7]	DBGRESTART	CA9
[6]	nCTIIRQ	INTC
[5]	Not used	—
[4]	EXTIN[3]	PTM-A9
[3]	EXTIN[2]	PTM-A9
[2]	EXTIN[1]	PTM-A9
[1]	EXTIN[0]	PTM-A9
[0]	EDBGRQ	CA9

chanhvinh.com

Table 43.4 CPU CTICS Trigger Inputs

Trigger Input Bit	Trigger Signal	Source Device
[7]	Not used	—
[6]	Not used	—
[5]	Not used	—
[4]	Not used	—
[3]	ACQCOMP	ETF
[2]	FULL	ETF
[1]	Not used	—
[0]	Not used	—

Table 43.5 CA9 CTI Trigger Outputs

Trigger Input Bit	Trigger Signal	Destination Device
[7]	Not used	—
[6]	Not used	—
[5]	Not used	—
[4]	Not used	—
[3]	TRIGIN	CPU TPIU
[2]	FLUSHIN	CPU TPIU
[1]	TRIGIN	ETF
[0]	FLUSHIN	ETF

Table 43.6 Address Map for CoreSight

System Address (Viewed from CPU)	Debug-APB Address (Viewed from Debugger)	Module
H'FC000000 to H'FC000FFF	H'80000000 to H'80000FFF	DAP ROM
H'FC001000 to H'FC001FFF	H'80001000 to H'80001FFF	Reserved
H'FC002000 to H'FC002FFF	H'80002000 to H'80002FFF	SYS-CTI0
H'FC003000 to H'FC003FFF	H'80003000 to H'80003FFF	Reserved
H'FC004000 to H'FC004FFF	H'80004000 to H'80004FFF	Reserved
H'FC005000 to H'FC005FFF	H'80005000 to H'80005FFF	Reserved
H'FC006000 to H'FC006FFF	H'80006000 to H'80006FFF	Reserved
H'FC007000 to H'FC007FFF	H'80007000 to H'80007FFF	Reserved
H'FC008000 to H'FC008FFF	H'80008000 to H'80008FFF	SYS-CTI2
H'FC009000 to H'FC009FFF	H'80009000 to H'80009FFF	Reserved
H'FC00A000 to H'FC00AFFF	H'8000A000 to H'8000AFFF	Reserved
H'FC00B000 to H'FC00BFFF	H'8000B000 to H'8000BFFF	Reserved
H'FC00C000 to H'FC00CFFF	H'8000C000 to H'8000CFFF	SYS-CTI1
H'FC00D000 to H'FC00DFFF	H'8000D000 to H'8000DFFF	Reserved
H'FC00E000 to H'FC00EFFF	H'8000E000 to H'8000EFFF	Reserved
H'FC00F000 to H'FC00FFFF	H'8000F000 to H'8000FFFF	ICE registers (ICEReg)
H'FC010000 to H'FC010FFF	H'80010000 to H'80010FFF	Reserved
H'FC011000 to H'FC011FFF	H'80011000 to H'80011FFF	Reserved
H'FC012000 to H'FC012FFF	H'80012000 to H'80012FFF	Reserved
H'FC013000 to H'FC013FFF	H'80013000 to H'80013FFF	Reserved
H'FC014000 to H'FC014FFF	H'80014000 to H'80014FFF	Reserved
H'FC015000 to H'FC015FFF	H'80015000 to H'80015FFF	Reserved

chanhvinh.com

Table 43.6 Address Map for CoreSight

System Address (Viewed from CPU)	Debug-APB Address (Viewed from Debugger)	Module
H'FC016000 to H'FC016FFF	H'80016000 to H'80016FFF	Reserved
H'FC017000 to H'FC017FFF	H'80017000 to H'80017FFF	Reserved
H'FC018000 to H'FC018FFF	H'80018000 to H'80018FFF	Reserved
H'FC019000 to H'FC019FFF	H'80019000 to H'80019FFF	Reserved
H'FC01A000 to H'FC01AFFF	H'8001A000 to H'8001AFFF	Reserved
H'FC01B000 to H'FC01BFFF	H'8001B000 to H'8001BFFF	Reserved
H'FC01C000 to H'FC01CFFF	H'8001C000 to H'8001CFFF	Reserved
H'FC01D000 to H'FC01DFFF	H'8001D000 to H'8001DFFF	Reserved
H'FC01E000 to H'FC01EFFF	H'8001E000 to H'8001EFFF	Reserved
H'FC01F000 to H'FC01FFFF	H'8001F000 to H'8001FFFF	Reserved
H'FC020000 to H'FC020FFF	H'80020000 to H'80020FFF	2nd DAP ROM
H'FC021000 to H'FC021FFF	H'80021000 to H'80021FFF	CPU-ETF
H'FC022000 to H'FC022FFF	H'80022000 to H'80022FFF	CPU-CTICS
H'FC023000 to H'FC023FFF	H'80023000 to H'80023FFF	CPU-TPIU
H'FC024000 to H'FC024FFF	H'80024000 to H'80024FFF	CPU-Trace Funnel
H'FC025000 to H'FC025FFF	H'80025000 to H'80025FFF	Reserved
H'FC026000 to H'FC026FFF	H'80026000 to H'80026FFF	Reserved
H'FC027000 to H'FC027FFF	H'80027000 to H'80027FFF	Reserved
H'FC028000 to H'FC028FFF	H'80028000 to H'80028FFF	Reserved
H'FC029000 to H'FC029FFF	H'80029000 to H'80029FFF	Reserved
H'FC02A000 to H'FC02AFFF	H'8002A000 to H'8002AFFF	Reserved
H'FC02B000 to H'FC02BFFF	H'8002B000 to H'8002BFFF	Reserved
H'FC02C000 to H'FC02CFFF	H'8002C000 to H'8002CFFF	Reserved
H'FC02D000 to H'FC02DFFF	H'8002D000 to H'8002DFFF	Reserved
H'FC02E000 to H'FC02EFFF	H'8002E000 to H'8002EFFF	Reserved
H'FC02F000 to H'FC02FFFF	H'8002F000 to H'8002FFFF	Reserved
H'FC030000 to H'FC030FFF	H'80030000 to H'80030FFF	CA9-DBG (CPU0)
H'FC031000 to H'FC031FFF	H'80031000 to H'80031FFF	CA9-PMU (CPU0)
H'FC032000 to H'FC032FFF	H'80032000 to H'80032FFF	Reserved
H'FC033000 to H'FC033FFF	H'80033000 to H'80033FFF	Reserved
H'FC034000 to H'FC034FFF	H'80034000 to H'80034FFF	Reserved
H'FC035000 to H'FC035FFF	H'80035000 to H'80035FFF	Reserved
H'FC036000 to H'FC036FFF	H'80036000 to H'80036FFF	Reserved
H'FC037000 to H'FC037FFF	H'80037000 to H'80037FFF	Reserved
H'FC038000 to H'FC038FFF	H'80038000 to H'80038FFF	CA9 CTI (CPU0)
H'FC039000 to H'FC039FFF	H'80039000 to H'80039FFF	Reserved
H'FC03A000 to H'FC03AFFF	H'8003A000 to H'8003AFFF	Reserved
H'FC03B000 to H'FC03BFFF	H'8003B000 to H'8003BFFF	Reserved
H'FC03C000 to H'FC03CFFF	H'8003C000 to H'8003CFFF	PTM-A9 (CPU0)
H'FC03D000 to H'FC03DFFF	H'8003D000 to H'8003DFFF	Reserved
H'FC03E000 to H'FC03EFFF	H'8003E000 to H'8003EFFF	Reserved
H'FC03F000 to H'FC03FFFF	H'8003F000 to H'8003FFFF	Reserved

chanhvinh.com

43.2 Input/Output Pins

Table 43.7 shows the pin configuration of the debugger interface.

Table 43.7 Pin Configuration

Name	Pin Name	Input/ Output	Function
Test clock	TCK/SWDCLK	Input	Data is input in serial to this module via the data input pin (TDI) or data is output via the data output pin (TDO) in synchronization with this signal. Functions as the SWDCLK pin in serial wire debug (SWD) mode.
Test mode select	TMS/SWDIO	Input, Input/ Output	Changing this signal in synchronization with the TCK signal determines the state of the TAP controller circuit. Its protocol conforms to the subset of the JTAG standard (IEEE standard 1149.1). Functions as the SWDIO pin in serial wire debug (SWD) mode.
Test reset	$\overline{\text{TRST}}$*1	Input	This signal is received asynchronously with the TCK signal. Asserting this signal resets this module. When power is supplied, this pin should be asserted for a given period regardless of whether or not this module function is used. For details on resetting, see section 43.5.2, Reset Signal Setting.
Test data input	TDI	Input	Data is sent to this module by changing this signal in synchronization with the TCK signal.
Test data output	TDO	Output	Data is read from this module by reading this signal in synchronization with the TCK signal. This pin also functions as an output pin in the serial wire debug (SWD) mode. However, a fixed value (the high level) is output because this product does not support the SWO function.
Boundary scan setting	BSCANP	Input	Input a high-level signal during boundary-scan testing. Input a low-level signal during normal use.
Clock output	TRACECLK	Output	Trace clock output pin
Enable output	TRACECTL	Output	Trace enable output pin
Data output	TRACEDAT3 to TRACEDATA0	Output	Trace data output pins

Note 1. When designing a board that can use an emulator, the $\overline{\text{RES}}$ and $\overline{\text{TRST}}$ pin circuits should be designed so that they can be set low and the $\overline{\text{TRST}}$ pin can be controlled independently at power-on. When the $\overline{\text{TRST}}$ pin is not in use, it should be either fixed to low level or connected to the $\overline{\text{RES}}$ pin (or another pin which operates in the same manner as the $\overline{\text{RES}}$ pin).

chanhvinh.com

43.3 Registers for Boundary-Scan TAP Controller

The boundary-scan TAP controller has the following registers.

Table 43.8 List of Registers of Boundary-Scan TAP Controller

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Bypass register	BSBPR	—	—	—	—
Instruction register	BSIR	—	H'55	—	—
Boundary scan register	SDBSR	—	—	—	—
ID register	BSID	—	H'081A6447*1 H'082F4447*2	—	—

Note 1. RZ/A1L
Note 2. RZ/A1LU and RZ/A1LC

43.3.1 Bypass Register (BSBPR)

BSBPR is a 1-bit register that cannot be accessed by the CPU. When the BSIR register is set to BYPASS mode, BSBPR is connected between the TDI and TDO pins. The initial value is undefined.

43.3.2 Instruction Register (BSIR)

BSIR is an 8-bit register. This register is initialized by TRST assertion or in the TAP test-logic-reset state. BSIR cannot be accessed by the CPU.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	TI[7:0]	01010101	—	Test Instruction The instructions of this module are transferred to BSIR by a serial input. For commands, see Table 43.9.

Table 43.9 Commands Supported for Boundary-Scan TAP Controller

Bits 7 to 0								
TI7	TI6	TI5	TI4	TI3	TI2	TI1	TI0	Description
0	0	0	0	0	0	0	0	EXTEST
0	1	0	0	0	0	0	0	SAMPLE/PRELOAD
0	1	0	1	0	1	0	1	IDCODE (Initial value)
1	1	0	1	0	0	0	0	CLAMP
1	0	0	0	0	0	0	0	HIGHZ
1	1	1	1	1	1	1	1	BYPASS
Other than above								Reserved

chanhvinh.com

43.3.3 Boundary Scan Register (SDBSR)

SDBSR is a shift register, located on the PAD, for controlling the input/output pins of this LSI. SDBSR cannot be accessed by the CPU. The initial value is undefined.

Using the EXTEST, SAMPLE/PRELOAD, CLAMP, and HIGHZ commands, a boundary scan test conforming to the JTAG standard can be carried out. Table 43.10 shows the correspondence between the pins of the RZ/A1L and RZ/A1LC group products and the boundary scan register bits.

Table 43.11 shows the correspondence between the pins of the RZ/A1LU group products and the boundary scan register bits.

Table 43.10 Correspondence between the Pins of the RZ/A1L and RZ/A1LC Group Products and the Boundary Scan Register Bits

208-pin Bit Number	176-pin Bit Number	Pin Name*1	Type
From TDI			
327	327	P3_0	OUTPUT
326	326	P3_0	CONTROL
325	325	P3_0	INPUT
324	324	P3_1	OUTPUT
323	323	P3_1	CONTROL
322	322	P3_1	INPUT
321	321	P3_2	OUTPUT
320	320	P3_2	CONTROL
319	319	P3_2	INPUT
318	318	P3_3	OUTPUT
317	317	P3_3	CONTROL
316	316	P3_3	INPUT
315	315	P3_4	OUTPUT
314	314	P3_4	CONTROL
313	313	P3_4	INPUT
312	-	P9_2	OUTPUT
311	-	P9_2	CONTROL
310	-	P9_2	INPUT
309	-	P9_3	OUTPUT
308	-	P9_3	CONTROL
307	-	P9_3	INPUT
306	306	P3_5	OUTPUT
305	305	P3_5	CONTROL
304	304	P3_5	INPUT
303	303	P3_6	OUTPUT
302	302	P3_6	CONTROL
301	301	P3_6	INPUT
300	300	P3_7	OUTPUT
299	299	P3_7	CONTROL
298	298	P3_7	INPUT
297	-	P9_4	OUTPUT
296	-	P9_4	CONTROL
295	-	P9_4	INPUT
294	-	P9_5	OUTPUT
293	-	P9_5	CONTROL
292	-	P9_5	INPUT
291	291	P3_8	OUTPUT
290	290	P3_8	CONTROL
289	289	P3_8	INPUT
288	288	P3_9	OUTPUT
287	287	P3_9	CONTROL
286	286	P3_9	INPUT
285	285	P3_10	OUTPUT
284	284	P3_10	CONTROL
283	283	P3_10	INPUT
282	282	P3_11	OUTPUT
281	281	P3_11	CONTROL
280	280	P3_11	INPUT
279	279	P3_12	OUTPUT
278	278	P3_12	CONTROL
277	277	P3_12	INPUT
276	276	P3_13	OUTPUT
275	275	P3_13	CONTROL
274	274	P3_13	INPUT
273	273	P3_14	OUTPUT
272	272	P3_14	CONTROL
271	271	P3_14	INPUT
270	270	P1_0	OUTPUT*2
269	269	P1_0	INPUT
268	268	P1_1	OUTPUT*2
267	267	P1_1	INPUT
266	266	P1_2	OUTPUT*2
265	265	P1_2	INPUT
264	264	P1_3	OUTPUT*2
263	263	P1_3	INPUT
262	262	P1_4	OUTPUT*2
261	261	P1_4	INPUT
260	260	P1_5	OUTPUT*2
259	259	P1_5	INPUT

chanhvinh.com

Table 43.10 Correspondence between the Pins of the RZ/A1L and RZ/A1LC Group Products and the Boundary Scan Register Bits

208-pin Bit Number	176-pin Bit Number	Pin Name*1	Type
258	258	P1_6	OUTPUT*2
257	257	P1_6	INPUT
256	256	P1_7	OUTPUT*2
255	255	P1_7	INPUT
254	254	P3_15	OUTPUT
253	253	P3_15	CONTROL
252	252	P3_15	INPUT
251	251	P2_6	OUTPUT
250	250	P2_6	CONTROL
249	249	P2_6	INPUT
248	248	P2_7	OUTPUT
247	247	P2_7	CONTROL
246	246	P2_7	INPUT
245	245	P2_8	OUTPUT
244	244	P2_8	CONTROL
243	243	P2_8	INPUT
242	242	P2_9	OUTPUT
241	241	P2_9	CONTROL
240	240	P2_9	INPUT
239	239	P4_0	OUTPUT
238	238	P4_0	CONTROL
237	237	P4_0	INPUT
236	236	P4_1	OUTPUT
235	235	P4_1	CONTROL
234	234	P4_1	INPUT
233	233	P4_2	OUTPUT
232	232	P4_2	CONTROL
231	231	P4_2	INPUT
230	230	P4_3	OUTPUT
229	229	P4_3	CONTROL
228	228	P4_3	INPUT
227	-	P8_0	OUTPUT
226	-	P8_0	CONTROL
225	-	P8_0	INPUT
224	-	P8_1	OUTPUT
223	-	P8_1	CONTROL
222	-	P8_1	INPUT
221	221	P4_4	OUTPUT
220	220	P4_4	CONTROL
219	219	P4_4	INPUT
218	218	P4_5	OUTPUT
217	217	P4_5	CONTROL
216	216	P4_5	INPUT
215	-	P8_2	OUTPUT
214	-	P8_2	CONTROL
213	-	P8_2	INPUT
212	-	P8_3	OUTPUT
211	-	P8_3	CONTROL
210	-	P8_3	INPUT
209	-	P8_4	OUTPUT
208	-	P8_4	CONTROL
207	-	P8_4	INPUT
206	-	P8_5	OUTPUT
205	-	P8_5	CONTROL
204	-	P8_5	INPUT
203	203	P4_6	OUTPUT
202	202	P4_6	CONTROL
201	201	P4_6	INPUT
200	200	P4_7	OUTPUT
199	199	P4_7	CONTROL
198	198	P4_7	INPUT
197	197	P5_0	OUTPUT
196	196	P5_0	CONTROL
195	195	P5_0	INPUT
194	194	P5_1	OUTPUT
193	193	P5_1	CONTROL
192	192	P5_1	INPUT
191	191	P5_2	OUTPUT
190	190	P5_2	CONTROL
189	189	P5_2	INPUT
188	188	P5_3	OUTPUT
187	187	P5_3	CONTROL
186	186	P5_3	INPUT
185	185	P5_4	OUTPUT
184	184	P5_4	CONTROL
183	183	P5_4	INPUT
182	182	P5_5	OUTPUT
181	181	P5_5	CONTROL
180	180	P5_5	INPUT
179	179	P5_6	OUTPUT
178	178	P5_6	CONTROL
177	177	P5_6	INPUT
176	176	P5_7	OUTPUT
175	175	P5_7	CONTROL
174	174	P5_7	INPUT
173	173	P5_8	OUTPUT
172	172	P5_8	CONTROL
171	171	P5_8	INPUT

chanhvinh.com

Table 43.10 Correspondence between the Pins of the RZ/A1L and RZ/A1LC Group Products and the Boundary Scan Register Bits

208-pin Bit Number	176-pin Bit Number	Pin Name*1	Type
170	170	P5_9	OUTPUT
169	169	P5_9	CONTROL
168	168	P5_9	INPUT
167	167	P5_10	OUTPUT
166	166	P5_10	CONTROL
165	165	P5_10	INPUT
164	164	P5_11	OUTPUT
163	163	P5_11	CONTROL
162	162	P5_11	INPUT
161	161	P5_12	OUTPUT
160	160	P5_12	CONTROL
159	159	P5_12	INPUT
158	158	P5_13	OUTPUT
157	157	P5_13	CONTROL
156	156	P5_13	INPUT
155	155	P5_14	OUTPUT
154	154	P5_14	CONTROL
153	153	P5_14	INPUT
152	152	P5_15	OUTPUT
151	151	P5_15	CONTROL
150	150	P5_15	INPUT
149	149	P6_0	OUTPUT
148	148	P6_0	CONTROL
147	147	P6_0	INPUT
146	146	P6_1	OUTPUT
145	145	P6_1	CONTROL
144	144	P6_1	INPUT
143	143	P6_2	OUTPUT
142	142	P6_2	CONTROL
141	141	P6_2	INPUT
140	140	P6_3	OUTPUT
139	139	P6_3	CONTROL
138	138	P6_3	INPUT
137	137	P6_4	OUTPUT
136	136	P6_4	CONTROL
135	135	P6_4	INPUT
134	134	P6_5	OUTPUT
133	133	P6_5	CONTROL
132	132	P6_5	INPUT
131	-	P8_6	OUTPUT
130	-	P8_6	CONTROL
129	-	P8_6	INPUT
128	-	P8_7	OUTPUT
127	-	P8_7	CONTROL
126	-	P8_7	INPUT
125	-	P8_8	OUTPUT
124	-	P8_8	CONTROL
123	-	P8_8	INPUT
122	-	P8_9	OUTPUT
121	-	P8_9	CONTROL
120	-	P8_9	INPUT
119	119	P6_6	OUTPUT
118	118	P6_6	CONTROL
117	117	P6_6	INPUT
116	116	P6_7	OUTPUT
115	115	P6_7	CONTROL
114	114	P6_7	INPUT
113	113	P6_8	OUTPUT
112	112	P6_8	CONTROL
111	111	P6_8	INPUT
110	-	P8_10	OUTPUT
109	-	P8_10	CONTROL
108	-	P8_10	INPUT
107	-	P8_11	OUTPUT
106	-	P8_11	CONTROL
105	-	P8_11	INPUT
104	104	P6_9	OUTPUT
103	103	P6_9	CONTROL
102	102	P6_9	INPUT
101	101	P6_10	OUTPUT
100	100	P6_10	CONTROL
99	99	P6_10	INPUT
98	98	P6_11	OUTPUT
97	97	P6_11	CONTROL
96	96	P6_11	INPUT
95	95	P6_12	OUTPUT
94	94	P6_12	CONTROL
93	93	P6_12	INPUT
92	92	P6_13	OUTPUT
91	91	P6_13	CONTROL
90	90	P6_13	INPUT
89	89	P6_14	OUTPUT
88	88	P6_14	CONTROL
87	87	P6_14	INPUT
86	86	P6_15	OUTPUT
85	85	P6_15	CONTROL
84	84	P6_15	INPUT
83	83	P7_0	OUTPUT

chanhvinh.com

Table 43.10 Correspondence between the Pins of the RZ/A1L and RZ/A1LC Group Products and the Boundary Scan Register Bits

208-pin Bit Number	176-pin Bit Number	Pin Name*1	Type
82	82	P7_0	CONTROL
81	81	P7_0	INPUT
80	80	P7_1	OUTPUT
79	79	P7_1	CONTROL
78	78	P7_1	INPUT
77	77	P7_2	OUTPUT
76	76	P7_2	CONTROL
75	75	P7_2	INPUT
74	74	P7_3	OUTPUT
73	73	P7_3	CONTROL
72	72	P7_3	INPUT
71	71	P7_4	OUTPUT
70	70	P7_4	CONTROL
69	69	P7_4	INPUT
68	68	P7_5	OUTPUT
67	67	P7_5	CONTROL
66	66	P7_5	INPUT
65	65	P7_6	OUTPUT
64	64	P7_6	CONTROL
63	63	P7_6	INPUT
62	62	P7_7	OUTPUT
61	61	P7_7	CONTROL
60	60	P7_7	INPUT
59	59	P7_8	OUTPUT
58	58	P7_8	CONTROL
57	57	P7_8	INPUT
56	56	P7_9	OUTPUT
55	55	P7_9	CONTROL
54	54	P7_9	INPUT
53	53	P7_10	OUTPUT
52	52	P7_10	CONTROL
51	51	P7_10	INPUT
50	50	P7_11	OUTPUT
49	49	P7_11	CONTROL
48	48	P7_11	INPUT
47	-	P8_12	OUTPUT
46	-	P8_12	CONTROL
45	-	P8_12	INPUT
44	-	P8_13	OUTPUT
43	-	P8_13	CONTROL
42	-	P8_13	INPUT
41	41	P2_0	OUTPUT
40	40	P2_0	CONTROL
39	39	P2_0	INPUT
38	38	P2_1	OUTPUT
37	37	P2_1	CONTROL
36	36	P2_1	INPUT
35	-	P8_14	OUTPUT
34	-	P8_14	CONTROL
33	-	P8_14	INPUT
32	-	P8_15	OUTPUT
31	-	P8_15	CONTROL
30	-	P8_15	INPUT
29	-	P9_0	OUTPUT
28	-	P9_0	CONTROL
27	-	P9_0	INPUT
26	-	P9_1	OUTPUT
25	-	P9_1	CONTROL
24	-	P9_1	INPUT
23	23	P2_2	OUTPUT
22	22	P2_2	CONTROL
21	21	P2_2	INPUT
20	20	P2_3	OUTPUT
19	19	P2_3	CONTROL
18	18	P2_3	INPUT
17	17	P0_0	INPUT
16	16	P0_1	INPUT
15	15	P0_2	INPUT
14	14	NMI	INPUT
13	13	P1_8	INPUT
12	12	P1_9	INPUT
11	11	P1_10	INPUT
10	10	P1_11	INPUT
9	9	P1_12	INPUT
8	8	P1_13	INPUT
7	7	P1_14	INPUT
6	6	P1_15	INPUT
5	5	P2_4	OUTPUT
4	4	P2_4	CONTROL
3	3	P2_4	INPUT
2	2	P2_5	OUTPUT
1	1	P2_5	CONTROL
0	0	P2_5	INPUT

Note 1. The pin names are as listed in the “Port function/dedicated function” column of Table 1.4, List of Pins in section 1.6, List of Pins.
Note 2. Open-drain pin. Setting the data bit for the pin to 0 makes the pin output a low-level signal and setting the data bit for the pin to 1 places the pin in the hi-Z state.

chanhvinh.com

Note 3. Bits of type “control” are active low. When the control bit is at the low level, the corresponding pin becomes an output.

Table 43.11 Correspondence between the Pins of the RZ/A1LU Group Products and the Boundary Scan Register Bits

208-pin 233-pin Bit Number	176-pin Bit Number	Pin Name*1	Type
From TDI			
327	327	P3_0	OUTPUT
326	326	P3_0	CONTROL
325	325	P3_0	INPUT
324	324	P3_1	OUTPUT
323	323	P3_1	CONTROL
322	322	P3_1	INPUT
321	321	P3_2	OUTPUT
320	320	P3_2	CONTROL
319	319	P3_2	INPUT
318	318	P3_3	OUTPUT
317	317	P3_3	CONTROL
316	316	P3_3	INPUT
315	315	P3_4	OUTPUT
314	314	P3_4	CONTROL
313	313	P3_4	INPUT
312	-	P9_2	OUTPUT
311	-	P9_2	CONTROL
310	-	P9_2	INPUT
309	-	P9_3	OUTPUT
308	-	P9_3	CONTROL
307	-	P9_3	INPUT
306	306	P3_5	OUTPUT
305	305	P3_5	CONTROL
304	304	P3_5	INPUT
303	303	P3_6	OUTPUT
302	302	P3_6	CONTROL
301	301	P3_6	INPUT
300	300	P3_7	OUTPUT
299	299	P3_7	CONTROL
298	298	P3_7	INPUT
297	-	P9_4	OUTPUT
296	-	P9_4	CONTROL
295	-	P9_4	INPUT
294	-	P9_5	OUTPUT
293	-	P9_5	CONTROL
292	-	P9_5	INPUT
291	291	P3_8	OUTPUT
290	290	P3_8	CONTROL
289	289	P3_8	INPUT
288	288	P3_9	OUTPUT
287	287	P3_9	CONTROL
286	286	P3_9	INPUT
285	285	P3_10	OUTPUT
284	284	P3_10	CONTROL
283	283	P3_10	INPUT
282	282	P3_11	OUTPUT
281	281	P3_11	CONTROL
280	280	P3_11	INPUT
279	279	P3_12	OUTPUT
278	278	P3_12	CONTROL
277	277	P3_12	INPUT
276	276	P3_13	OUTPUT
275	275	P3_13	CONTROL
274	274	P3_13	INPUT
273	273	P3_14	OUTPUT
272	272	P3_14	CONTROL
271	271	P3_14	INPUT
270	270	P1_0	OUTPUT
269	269	P1_0	INPUT
268	268	P1_1	OUTPUT*2
267	267	P1_1	INPUT
266	266	P1_2	OUTPUT*2
265	265	P1_2	INPUT
264	264	P1_3	OUTPUT*2
263	263	P1_3	INPUT
262	262	P1_4	OUTPUT*2
261	261	P1_4	INPUT
260	260	P1_5	OUTPUT*2
259	259	P1_5	INPUT
258	258	P1_6	OUTPUT*2
257	257	P1_6	INPUT
256	256	P1_7	OUTPUT*2
255	255	P1_7	INPUT
254	254	P3_15	OUTPUT
253	253	P3_15	CONTROL
252	252	P3_15	INPUT
251	251	P2_6	OUTPUT
250	250	P2_6	CONTROL
249	249	P2_6	INPUT
248	248	P2_7	OUTPUT
247	247	P2_7	CONTROL

chanhvinh.com

Table 43.11 Correspondence between the Pins of the RZ/A1LU Group Products and the Boundary Scan Register Bits

208-pin 233-pin Bit Number	176-pin Bit Number	Pin Name*1	Type
246	246	P2_7	INPUT
245	245	P2_8	OUTPUT
244	244	P2_8	CONTROL
243	243	P2_8	INPUT
242	242	P2_9	OUTPUT
241	241	P2_9	CONTROL
240	240	P2_9	INPUT
239	239	P4_0	OUTPUT
238	238	P4_0	CONTROL
237	237	P4_0	INPUT
236	236	P4_1	OUTPUT
235	235	P4_1	CONTROL
234	234	P4_1	INPUT
233	233	P4_2	OUTPUT
232	232	P4_2	CONTROL
231	231	P4_2	INPUT
230	230	P4_3	OUTPUT
229	229	P4_3	CONTROL
228	228	P4_3	INPUT
227	-	P8_0	OUTPUT
226	-	P8_0	CONTROL
225	-	P8_0	INPUT
224	-	P8_1	OUTPUT
223	-	P8_1	CONTROL
222	-	P8_1	INPUT
221	221	P4_4	OUTPUT
220	220	P4_4	CONTROL
219	219	P4_4	INPUT
218	218	P4_5	OUTPUT
217	217	P4_5	CONTROL
216	216	P4_5	INPUT
215	-	P8_2	OUTPUT
214	-	P8_2	CONTROL
213	-	P8_2	INPUT
212	-	P8_3	OUTPUT
211	-	P8_3	CONTROL
210	-	P8_3	INPUT
209	-	P8_4	OUTPUT
208	-	P8_4	CONTROL
207	-	P8_4	INPUT
206	-	P8_5	OUTPUT
205	-	P8_5	CONTROL
204	-	P8_5	INPUT

208-pin 233-pin Bit Number	176-pin Bit Number	Pin Name*1	Type
203	203	P4_6	OUTPUT
202	202	P4_6	CONTROL
201	201	P4_6	INPUT
200	200	P4_7	OUTPUT
199	199	P4_7	CONTROL
198	198	P4_7	INPUT
197	197	P5_0	OUTPUT
196	196	P5_0	CONTROL
195	195	P5_0	INPUT
194	194	P5_1	OUTPUT
193	193	P5_1	CONTROL
192	192	P5_1	INPUT
191	191	P5_2	OUTPUT
190	190	P5_2	CONTROL
189	189	P5_2	INPUT
188	188	P5_3	OUTPUT
187	187	P5_3	CONTROL
186	186	P5_3	INPUT
185	185	P5_4	OUTPUT
184	184	P5_4	CONTROL
183	183	P5_4	INPUT
182	182	P5_5	OUTPUT
181	181	P5_5	CONTROL
180	180	P5_5	INPUT
179	179	P5_6	OUTPUT
178	178	P5_6	CONTROL
177	177	P5_6	INPUT
176	176	P5_7	OUTPUT
175	175	P5_7	CONTROL
174	174	P5_7	INPUT
173	173	P5_8	OUTPUT
172	172	P5_8	CONTROL
171	171	P5_8	INPUT
170	170	P5_9	OUTPUT
169	169	P5_9	CONTROL
168	168	P5_9	INPUT
167	167	P5_10	OUTPUT
166	166	P5_10	CONTROL
165	165	P5_10	INPUT
164	164	P5_11	OUTPUT
163	163	P5_11	CONTROL
162	162	P5_11	INPUT
161	161	P5_12	OUTPUT

chanhvinh.com

Table 43.11 Correspondence between the Pins of the RZ/A1LU Group Products and the Boundary Scan Register Bits

208-pin 233-pin Bit Number	176-pin Bit Number	Pin Name*1	Type
160	160	P5_12	CONTROL
159	159	P5_12	INPUT
158	158	P5_13	OUTPUT
157	157	P5_13	CONTROL
156	156	P5_13	INPUT
155	155	P5_14	OUTPUT
154	154	P5_14	CONTROL
153	153	P5_14	INPUT
152	152	P5_15	OUTPUT
151	151	P5_15	CONTROL
150	150	P5_15	INPUT
149	149	P6_0	OUTPUT
148	148	P6_0	CONTROL
147	147	P6_0	INPUT
146	146	P6_1	OUTPUT
145	145	P6_1	CONTROL
144	144	P6_1	INPUT
143	143	P6_2	OUTPUT
142	142	P6_2	CONTROL
141	141	P6_2	INPUT
140	140	P6_3	OUTPUT
139	139	P6_3	CONTROL
138	138	P6_3	INPUT
137	137	P6_4	OUTPUT
136	136	P6_4	CONTROL
135	135	P6_4	INPUT
134	134	P6_5	OUTPUT
133	133	P6_5	CONTROL
132	132	P6_5	INPUT
131	-	P8_6	OUTPUT
130	-	P8_6	CONTROL
129	-	P8_6	INPUT
128	-	P8_7	OUTPUT
127	-	P8_7	CONTROL
126	-	P8_7	INPUT
125	-	P8_8	OUTPUT
124	-	P8_8	CONTROL
123	-	P8_8	INPUT
122	-	P8_9	OUTPUT
121	-	P8_9	CONTROL
120	-	P8_9	INPUT
119	119	P6_6	OUTPUT
118	118	P6_6	CONTROL
117	117	P6_6	INPUT
116	116	P6_7	OUTPUT
115	115	P6_7	CONTROL
114	114	P6_7	INPUT
113	113	P6_8	OUTPUT
112	112	P6_8	CONTROL
111	111	P6_8	INPUT
110	-	P8_10	OUTPUT
109	-	P8_10	CONTROL
108	-	P8_10	INPUT
107	-	P8_11	OUTPUT
106	-	P8_11	CONTROL
105	-	P8_11	INPUT
104	104	P6_9	OUTPUT
103	103	P6_9	CONTROL
102	102	P6_9	INPUT
101	101	P6_10	OUTPUT
100	100	P6_10	CONTROL
99	99	P6_10	INPUT
98	98	P6_11	OUTPUT
97	97	P6_11	CONTROL
96	96	P6_11	INPUT
95	95	P6_12	OUTPUT
94	94	P6_12	CONTROL
93	93	P6_12	INPUT
92	92	P6_13	OUTPUT
91	91	P6_13	CONTROL
90	90	P6_13	INPUT
89	89	P6_14	OUTPUT
88	88	P6_14	CONTROL
87	87	P6_14	INPUT
86	86	P6_15	OUTPUT
85	85	P6_15	CONTROL
84	84	P6_15	INPUT
83	83	P7_0	OUTPUT
82	82	P7_0	CONTROL
81	81	P7_0	INPUT
80	80	P7_1	OUTPUT
79	79	P7_1	CONTROL
78	78	P7_1	INPUT
77	77	P7_2	OUTPUT
76	76	P7_2	CONTROL
75	75	P7_2	INPUT

chanhvinh.com

Table 43.11 Correspondence between the Pins of the RZ/A1LU Group Products and the Boundary Scan Register Bits

208-pin 233-pin Bit Number	176-pin Bit Number	Pin Name*1	Type
74	74	P7_3	OUTPUT
73	73	P7_3	CONTROL
72	72	P7_3	INPUT
71	71	P7_4	OUTPUT
70	70	P7_4	CONTROL
69	69	P7_4	INPUT
68	68	P7_5	OUTPUT
67	67	P7_5	CONTROL
66	66	P7_5	INPUT
65	65	P7_6	OUTPUT
64	64	P7_6	CONTROL
63	63	P7_6	INPUT
62	62	P7_7	OUTPUT
61	61	P7_7	CONTROL
60	60	P7_7	INPUT
59	59	P7_8	OUTPUT
58	58	P7_8	CONTROL
57	57	P7_8	INPUT
56	56	P7_9	OUTPUT
55	55	P7_9	CONTROL
54	54	P7_9	INPUT
53	-	P8_12	OUTPUT
52	-	P8_12	CONTROL
51	-	P8_12	INPUT
50	-	P8_13	OUTPUT
49	-	P8_13	CONTROL
48	-	P8_13	INPUT
47	47	P7_10	OUTPUT
46	46	P7_10	CONTROL
45	45	P7_10	INPUT
44	44	P7_11	OUTPUT
43	43	P7_11	CONTROL
42	42	P7_11	INPUT
41	41	P2_0	OUTPUT
40	40	P2_0	CONTROL
39	39	P2_0	INPUT
38	38	P2_1	OUTPUT
37	37	P2_1	CONTROL
36	36	P2_1	INPUT
35	-	P8_14	OUTPUT
34	-	P8_14	CONTROL
33	-	P8_14	INPUT
32	-	P8_15	OUTPUT
31	-	P8_15	CONTROL
30	-	P8_15	INPUT
29	-	P9_0	OUTPUT
28	-	P9_0	CONTROL
27	-	P9_0	INPUT
26	-	P9_1	OUTPUT
25	-	P9_1	CONTROL
24	-	P9_1	INPUT
23	23	P2_2	OUTPUT
22	22	P2_2	CONTROL
21	21	P2_2	INPUT
20	20	P2_3	OUTPUT
19	19	P2_3	CONTROL
18	18	P2_3	INPUT
17	17	P0_0	INPUT
16	16	P0_1	INPUT
15	15	P0_2	INPUT
14	14	NMI	INPUT
13	13	P1_8	INPUT
12	12	P1_9	INPUT
11	11	P1_10	INPUT
10	10	P1_11	INPUT
9	9	P1_12	INPUT
8	8	P1_13	INPUT
7	7	P1_14	INPUT
6	6	P1_15	INPUT
5	5	P2_4	OUTPUT
4	4	P2_4	CONTROL
3	3	P2_4	INPUT
2	2	P2_5	OUTPUT
1	1	P2_5	CONTROL
0	0	P2_5	INPUT

Note 1. The pin names are as listed in the “Port function/dedicated function” column of Table 1.4, List of Pins in section 1.6, List of Pins.
Note 2. Open-drain pin. Setting the data bit for the pin to 0 makes the pin output a low-level signal and setting the data bit for the pin to 1 places the pin in the hi-Z state.
Note 3. Bits of type “control” are active low. When the control bit is at the low level, the corresponding pin becomes an output.

chanhvinh.com

43.3.4 ID Register (BSID)

BSID is a 32-bit read-only register that cannot be accessed by the CPU. BSID can be only be read from the pins when the IDCODE command is set.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DID[31:16]															
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DID[15:0]															
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*: RZ/A1L	0	0	0	0	1	0	0	0	0	0	0	1	1	0	1	0
	0	1	1	0	0	1	0	0	0	1	0	0	0	1	1	1
*: RZ/A1LU and RZ/A1LC	0	0	0	0	1	0	0	0	0	0	1	0	1	1	1	1
	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DID[31:0]	H'081A6447*1 H'082F4447*2	—	Device IDCODE This is the ID register stipulated by JTAG. Note that the higher-order four bits may be changed depending on the version of the chip.

Note 1. RZ/A1L
Note 2. RZ/A1LU and RZ/A1LC

chanhvinh.com

43.4 ICE Registers

The debugger interface has the following ICE registers.
ICE registers are exclusively for use in debugging. Do not use these registers other than for debugging.

Table 43.12 Configuration of ICE Registers

Register Name	Abbreviation	R/W	Initial Value	Address		Access Size
				Viewed from CPU	Viewed from Debugger	
Mode reset control register	ICEREGMDRSTCTL	R/W	H'0000111E	H'FC00F000	H'8000F000	32
JTAG trace select register	ICEREGJTTRCSEL	R/W	H'00800000	H'FC00F004	H'8000F004	32
Clock power control register	ICEREGCLKPWRCTRL	R/W	H'00000000	H'FC00F014	H'8000F014	32
Lock access register	ICEREGLOCKACCES	W	—	H'FC00FFB0	H'8000FFB0	32

43.4.1 Mode Reset Control Register (ICEREGMDRSTCTL)

ICEREGMDRSTCTL is used to set the debug mode and to perform software reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	NIDEN_CPU0	—	—	—	DBGEN_CPU0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	RSTRB_CPU0_DERSTZ	—	—	—	RSTRB_CPU0_CPURSTZ	—	—	—	RSTRB_SYS_SYSRSTZ	—	RSTRB_CPU_PRSTDBGZ	RSTRB_CPU_SYSRSTZ	—
Initial value:	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved The read value is always 0.
20	NIDEN_CPU0	0	R/W	CPU Noninvasive Debug Enable Controls the noninvasive debug enable signal (NIDEN/SPNIDEN) supplied to the CPU. This bit is only effective when the PINSETEN bit in ICEREGJTTRCSEL is set to 0.
19 to 17	—	All 0	R	Reserved The read value is always 0.
16	DBGEN_CPU0	0	R/W	CPU Debug Enable Controls the debug enable signal (DBGEN/SPIDEN) supplied to the CPU. This bit is only effective when the PINSETEN bit in ICEREGJTTRCSEL is set to 0.
15 to 13	—	All 0	R	Reserved The read value is always 0.
12	RSTRB_CPU0_DERSTZ	1	R/W	NEON Reset (low-active) Resets the NEON unit in the CPU.
11 to 9	—	All 0	R	Reserved The read value is always 0.
8	RSTRB_CPU0_CPURSTZ	1	R/W	CPU Reset (low-active) Resets all systems on the CPU block other than debugging resources.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved The read value is always 0.
4	RSTRB_SYS_SYSRSTZ	1	R/W	System Reset (low-active) Resets all systems on the chip other than debugging resources.
3	—	1	R/W	Reserved The read value is always 1. The write value should always be 1.
2	RSTRB_CPU_PRSTDBGZ	1	R/W	CPU Subsystem Debug Peripheral Reset (low-active) Resets the debug-APB block in the CPU subsystem.
1	RSTRB_CPU_SYSRSTZ	1	R/W	CPU Subsystem Reset (low-active) Resets the CPU subsystem other than those for debugging.
0	—	0	R	Reserved The read value is always 0.

43.4.2 JTAG Trace Select Register (ICEREGJTTRCSEL)

ICEREGJTTRCSEL is used to control trace data output to the respective pins and debug enable signal supplied to the CPU.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	PINSETEN	—	—	TRCMUX_SEL	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R	R	R/W	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved The read value is always 0.
23	PINSETEN	1	R/W	Selects control method of debug enable signal supplied to the CPU. 0: Supply of the DBGEN and NIDEN signals is determined by the ICEREGMDRSTCTL register. 1: Supply of the DBGEN and NIDEN signals is determined by the BSCANP pin.
22, 21	—	All 0	R	Reserved The read value is always 0.
20	TRCMUX_SEL	0	R/W	Sets the priority of the function multiplexed on the pins to which trace output is assigned. 0: The function of the pins depends on the general I/O port setting. 1: The pins act as trace outputs regardless of the general I/O port setting.
19 to 0	—	All 0	R	Reserved The read value is always 0.

chanhvinh.com

43.4.3 Clock Power Control Register (ICEREGCLKPWRCTRL)

ICEREGCLKPWRCTRL is used to set the fake debug mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	FAKEDBGCTRL	—	—	—	—	—	—	FAKEDBG
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved The read value is always 0.
7	FAKEDBGCTRL	0	R/W	Enables or disables the fake debug mode. 0: The power control is determined by the standby control register 1 (STBCR1). 1: The power control is determined by this register.
6 to 1	—	All 0	R	Reserved The read value is always 0.
0	FAKEDBG	0	R/W	Selects whether or not to stop the power supply in deep standby mode. 0: The power supply is actually stopped in deep standby mode. 1: The power supply is not stopped even in deep standby mode.

43.4.4 Lock Access Register (ICEREGLOCKACCESS)

ICEREGLOCKACCESS is used to enable access to ICE registers by the CPU.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICEREGLOCKACCESS															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICEREGLOCKACCESS															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	ICEREGLOCKACCESS	—	W	To enable access to ICE registers by the CPU, first write H'C5AC CE55 to this register.

chanhvinh.com

43.5 Operation

43.5.1 TAP Controller

Figure 43.3 shows the internal states of the TAP controller. State transitions basically conform to the JTAG standard.

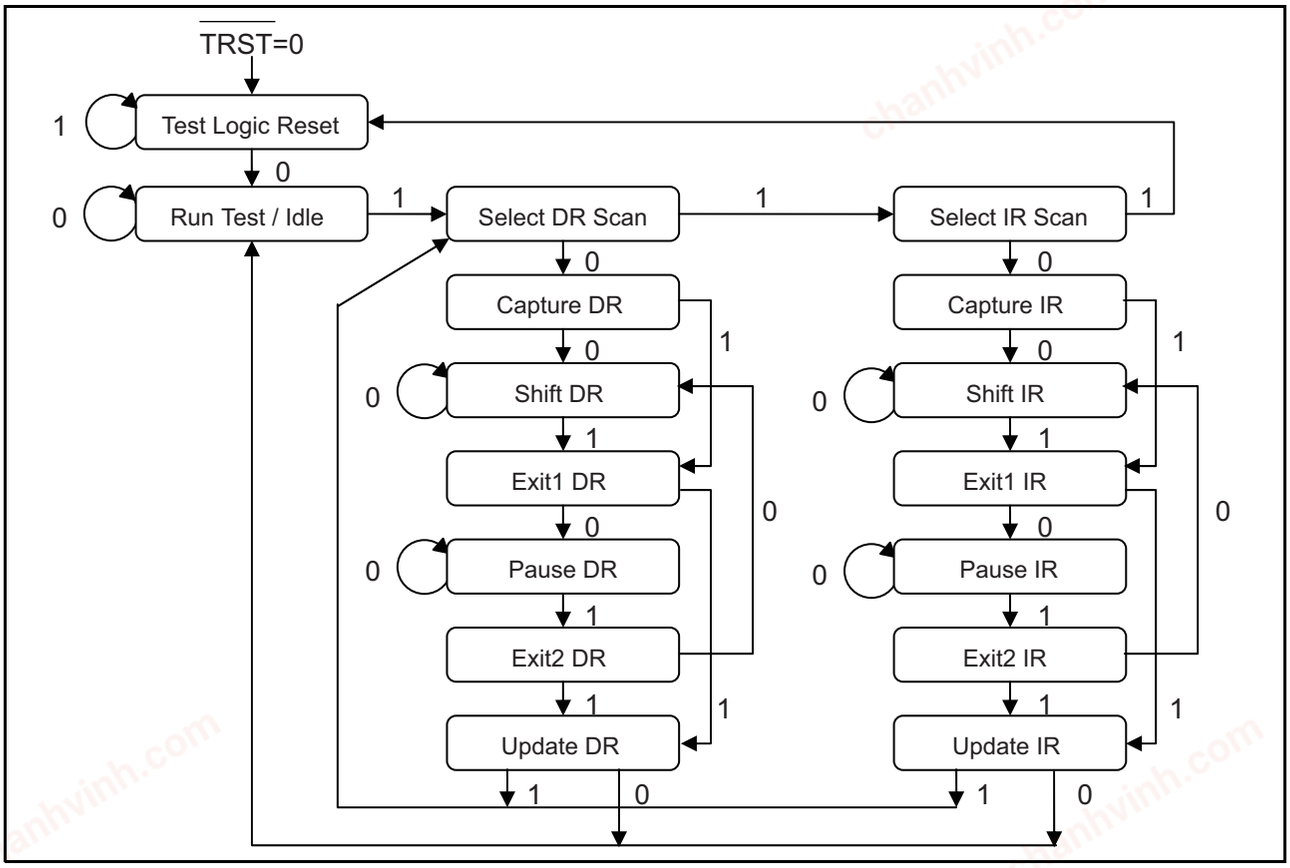


Figure 43.3 State Transitions of TAP Controller

Note: State transition occur according to the TMS value at the rising edge of the TCK signal. The TDI value sampled at the rising edge of the TCK signal and shifted at the falling edge of the TCK signal. TDO value is output at the rising edge of the TCK signal. The TDO signal is in a Hi-Z state for TAP controller states other than Shift-DR and Shift-IR. A transition to the Test-Logic-Reset state by clearing $\overline{\text{TRST}}$ to 0 is performed asynchronously with the TCK signal.

43.5.2 Reset Signal Setting

Table 43.13 Reset Signal Setting

$\overline{\text{RES}}$	$\overline{\text{TRST}}$	Chip State
L	L	The chip is power-on reset and this module is reset.*1
	H	The chip is power-on reset.*1
H	L	This module is only reset.*2
	H	Normal operation*2

Note 1. By asserting the $\overline{\text{RES}}$ and $\overline{\text{TRST}}$ signals, the CPU and CoreSight enter the reset state. Set the debugging function while asserting the $\overline{\text{RES}}$ signal after negating the $\overline{\text{TRST}}$ signal.

Note 2. Do not negate the $\overline{\text{TRST}}$ signal while the $\overline{\text{RES}}$ signal is at the high-level.

Note 3. When the $\overline{\text{TRST}}$ pin is to be negated, make sure that the specification for $\overline{\text{RES}}$ input rise time (t_{RSr}) or $\overline{\text{RES}}$ negating hold time (t_{RSNH}) in Table 47.6, Control Signal Timing, is satisfied.

chanhvinh.com

43.6 Boundary Scan

Commands can be set in BSIR by this module to place this module's pins in boundary scan mode stipulated by JTAG.

43.6.1 Supported Instructions

This LSI supports three mandatory instructions stipulated by JTAG standard (BYPASS, SAMPLE/PRELOAD, and EXTEST) and three option instructions (IDCODE, CLAMP, and HIGHZ).

(1) BYPASS

BYPASS is a mandatory instruction that operates the bypass register. This instruction is used to shorten the shift path to speed up serial data transfer involving other LSI on the printed circuit board. This LSI's system circuits are not affected by execution of this instruction.

(2) SAMPLE/PRELOAD

SAMPLE/PRELOAD is used to input data from this LSI’s internal circuit to the boundary scan register, outputs data from the scan path, and loads data onto the scan path. While this instruction is being executed, signals input to this LSI pins are transmitted directly to the internal circuit, and the values of the internal circuit are directly output externally from the output pins. This LSI's system circuits are not affected by execution of this instruction.

In a SAMPLE operation, a snapshot of a value to be transferred from an input pin to the internal circuit, or a value to be transferred from the internal circuit to an output pin, is latched into the boundary scan register and read from the scan path. Snapshot latching is performed in synchronization with the rising edge of the TCK signal in the Capture-DR state. Snapshot latching does not affect normal operation of this LSI.

In a PRELOAD operation, an initial value is set in the parallel output latch of the boundary scan register from the scan path prior to the EXTEST instruction. Without a PRELOAD operation, when the EXTEST instruction was executed an undefined value would be output from the output pin until completion of the initial scan sequence (transfer to the output latch) (with the EXTEST instruction, the parallel output latch value is constantly output to the output pin).

(3) EXTEST

This instruction is provided to test external circuit when this LSI is mounted on a printed circuit board. When this instruction is executed, output pins are used to output test data (previously set by the SAMPLE/PRELOAD instruction) from the boundary scan register to the printed circuit board, and input pins are used to latch test results into the boundary scan register from the printed circuit board. If testing is carried out by using the EXTEST instruction N times, the Nth test data is scanned-in when test data (N-1) is scanned out. The data loaded into the output pin boundary scan register in the Capture-DR state are not used for external circuit testing (data are replaced by the shift operation).

(4) IDCODE

Commands can be set in BSIR via pins of this module to place the module’s pins in the IDCODE mode stipulated by JTAG. When this module is initialized ($\overline{\text{TRST}}$ is asserted or TAP is in the Test-Logic-Reset state), the IDCODE mode is entered.

(5) CLAMP, HIGHZ

Commands can be set in BSIR via pins of this module to place the module’s pins in the CLAMP or HIGHZ mode stipulated by JTAG.

chanhvinh.com

43.6.2 Points for Attention

1. Boundary scan mode does not cover clock-related signals (EXTAL, XTAL, CKIO, AUDIO_X1, AUDIO_X2, USB_X1, USB_X2, RTC_X1, and RTC_X2).
2. Boundary scan mode does not cover reset-related signal ($\overline{\text{RES}}$).
3. Boundary scan mode does not cover this module's signals (TCK, TDI, TDO, TMS, $\overline{\text{TRST}}$, and BSCANP).
4. Boundary scan mode does not cover USB-related signals (DP0, DP1, DM0, DM1, VBUS0, VBUS1, and REFRIN).
5. Boundary scan mode does not cover P0_3.

43.7 Usage Notes

1. Once a command of this module is set, it will not be modified unless another command is reissued. If the same command is given continuously, the command must be set after a command (BYPASS, etc.) that does not affect LSI operations is once set.
2. Regardless of whether or not this module is to be used, it should be initialized by asserting the $\overline{\text{TRST}}$ signal when power is supplied and on release from deep standby mode (excluding fake debug mode) through assertion of the $\overline{\text{RES}}$ signal.
3. Be sure to wait for at least 200 ns after negating the $\overline{\text{TRST}}$ signal before starting TAP controller operation.
4. Be sure to fix the TMS pin to the high level until 200 ns have elapsed after negation of the $\overline{\text{TRST}}$ signal.

chanhvinh.com

44. JPEG Codec Unit

This module is only incorporated in the RZ/A1LU.
The JPEG codec unit (JCU) incorporates a JPEG codec conforming to the JPEG baseline compression and decompression standard to provide high-speed compression of image data and high-speed decoding of JPEG data.

44.1 Features

The JPEG codec unit has the following features:

- Conforms to the JPEG baseline standard within the range described in this document.
  This module does not support the following basic features:
  Scanning with two elements
  Non-interleave scanning with multiple elements
- Operational precision: Conforming to JPEG Part 2, ISO-IEC10918-2
- Image input/output system: Block interleave method
- Pixel format:
  Compression: YCbCr422 (H = 2:1:1, V = 1:1:1)
  Decompression: YCbCr444 (H = 1:1:1, V = 1:1:1), YCbCr422 (H = 2:1:1, V = 1:1:1), YCbCr411 (H = 4:1:1, V = 1:1:1), YCbCr420 (H = 2:1:1, V = 2:1:1)
  Output pixel format to the buffer: YCbCr422, ARGB8888, RGB565
- Four quantization tables provided
- Four Huffman tables provided (two tables for AC coefficients and two tables for DC coefficients)
- Markers supported: SOI (start of image), SOF0 (start of frame type 0), SOS (start of scan), DQT (define quantization tables), DHT (define Huffman tables), DRI (define restart interval), RSTm (restart marks), and EOI (end of image)
- Image data rate: Max. 133.34 Mbytes/s (at 66.67-MHz operation)
- The buffer size can be reduced by using the mode in which data transfer is temporarily stopped each time the specified number of lines or the specified amount of data is transferred during image data or coded data input/output.
- Processing unit: 8-byte address boundary units can be set
- Image sizes that can be processed: Sizes divisible by the minimum coded unit (MCU): 8 lines by 8 pixels in YCbCr444; 8 lines by 16 pixels in YCbCr422; 8 lines by 32 pixels in YCbCr411; 16 lines by 16 pixels in YCbCr420

Note: • Compression and decompression processing of images in unsupported pixel formats or unsupported image sizes should be avoided.

chanhvinh.com

Figure 44.1 shows a block diagram.

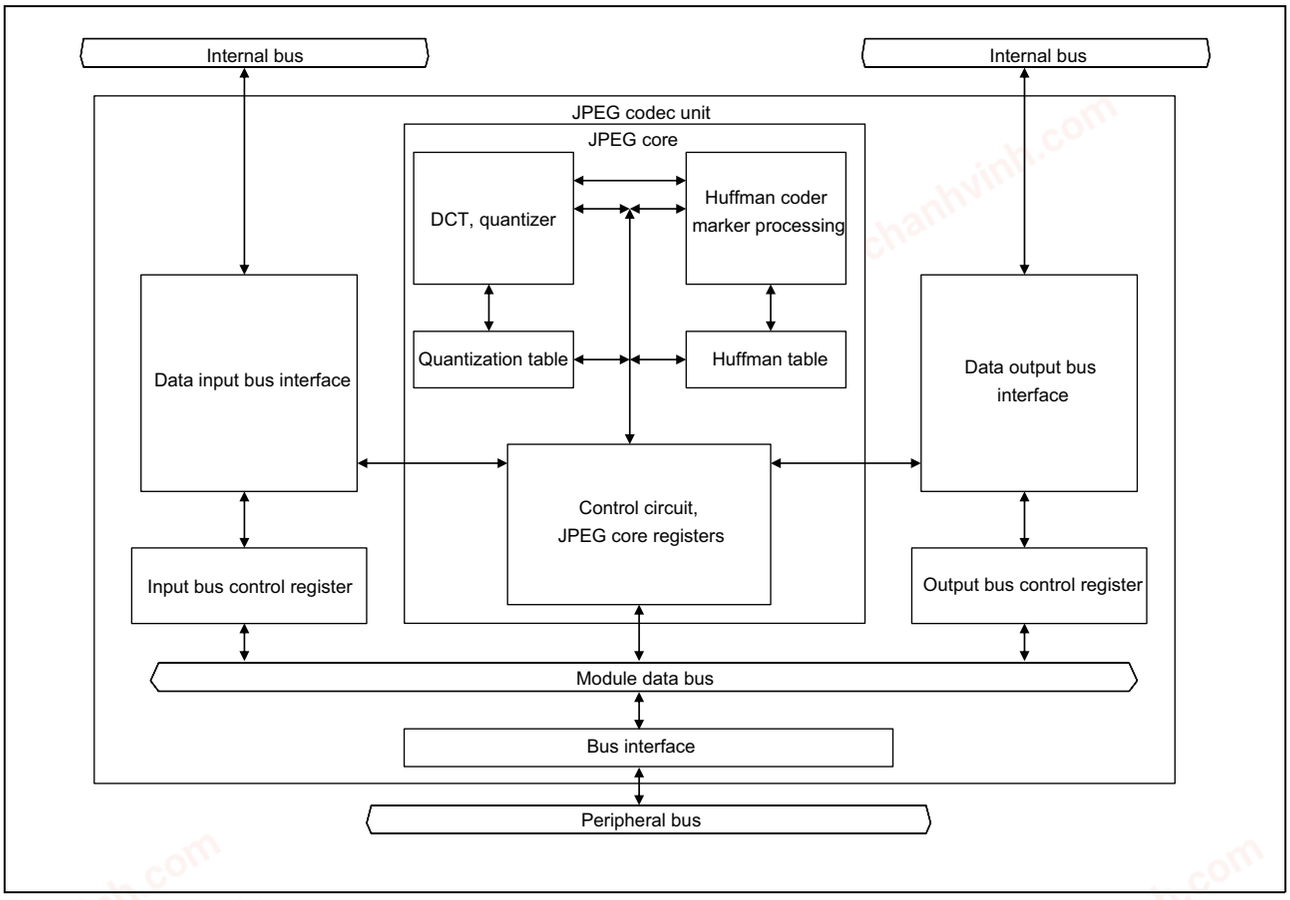


Figure 44.1 Block Diagram

chanhvinh.com

44.2 Register Descriptions

Table 44.1 shows the JCU registers.

Table 44.1 Register Configuration

Register Name	Abbreviation	R/W	Address	Access Size
JPEG code mode register	JCMOD	R/W	H'E801 7000	8
JPEG code command register	JCCMD	R/W	H'E801 7001	8
JPEG code quantization table number register	JCQTN	R/W	H'E801 7003	8
JPEG code Huffman table number register	JCHTN	R/W	H'E801 7004	8
JPEG code DRI upper register	JCDRIU	R/W	H'E801 7005	8
JPEG code DRI lower register	JCDRID	R/W	H'E801 7006	8
JPEG code vertical size upper register	JCVSZU	R/W	H'E801 7007	8
JPEG code vertical size lower register	JCVSZD	R/W	H'E801 7008	8
JPEG code horizontal size upper register	JCHSZU	R/W	H'E801 7009	8
JPEG code horizontal size lower register	JCHSZD	R/W	H'E801 700A	8
JPEG code data count upper register	JCDTCU	R	H'E801 700B	8
JPEG code data count middle register	JCDTCM	R	H'E801 700C	8
JPEG code data count lower register	JCDTCD	R	H'E801 700D	8
JPEG interrupt enable register 0	JINTE0	R/W	H'E801 700E	8
JPEG interrupt status register 0	JINTS0	R/W	H'E801 700F	8
JPEG code decode error register	JCDERR	R/W	H'E801 7010	8
JPEG code reset register	JCRST	R	H'E801 7011	8
JPEG interface compression control register	JIFECNT	R/W	H'E801 7040	32
JPEG interface compression source address register	JIFESA	R/W	H'E801 7044	32
JPEG interface compression line offset register	JIFESOFST	R/W	H'E801 7048	32
JPEG interface compression destination address register	JIFEDA	R/W	H'E801 704C	32
JPEG interface compression source line count register	JIFESLC	R/W	H'E801 7050	32
JPEG interface compression destination register	JIFEDDC	R/W	H'E801 7054	32
JPEG interface decompression control register	JIFDCNT	R/W	H'E801 7058	32
JPEG interface decompression source address register	JIFDSA	R/W	H'E801 705C	32
JPEG interface decompression destination offset register	JIFDDOFST	R/W	H'E801 7060	32
JPEG interface decompression destination address register	JIFDDA	R/W	H'E801 7064	32
JPEG interface decompression source count register	JIFDSDC	R/W	H'E801 7068	32
JPEG interface decompression destination line count register	JIFDDLC	R/W	H'E801 706C	32
JPEG interface decompression α setting register	JIFDADT	R/W	H'E801 7070	32
JPEG interrupt enable register 1	JINTE1	R/W	H'E801 708C	32
JPEG interrupt status register 1	JINTS1	R/W	H'E801 7090	32
JPEG input image data CbCr range setting register	JIFESVSZ	R/W	H'E801 7094	32
JPEG output image data CbCr range setting register	JIFESHSZ	R/W	H'E801 7098	32
JPEG code quantization table 0 register	JCQTBL0	R/W	H'E801 7100 to H'E801 713F	8
JPEG code quantization table 1 register	JCQTBL1	R/W	H'E801 7140 to H'E801 717F	8
JPEG code quantization table 2 register	JCQTBL2	R/W	H'E801 7180 to H'E801 71BF	8
JPEG code quantization table 3 register	JCQTBL3	R/W	H'E801 71C0 to H'E801 71FF	8

chanhvinh.com

Table 44.1 Register Configuration

Register Name	Abbreviation	R/W	Address	Access Size
JPEG code Huffman table DC0 register	JCHTBD0	W	H'E801 7200 to H'E801 721B	8
JPEG code Huffman table AC0 register	JCHTBA0	W	H'E801 7220 to H'E801 72D1	8
JPEG code Huffman table DC1 register	JCHTBD1	W	H'E801 7300 to H'E801 731B	8
JPEG code Huffman table AC1 register	JCHTBA1	W	H'E801 7320 to H'E801 73D1	8

Note: • For the settings of the JPEG code quantization table and JPEG code Huffman table, see section 44.3.1 (4), Table Setting.

44.2.1 JPEG Code Mode Register (JCMOD)

JCMOD sets the operating mode before the JCU starts operation.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	DSP	REDU[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W(compress):	R	R	R	R	R/W	R/W	R/W	R/W
R/W(decompress):	R	R	R	R	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W		Description
			Compression	De-compression	
7 to 4	—	All 0		R	Reserved These bits are always read as 0. The write value should always be 0.
3	DSP	0		R/W	Compression/Decompression Set 0: Compression process 1: Decompression process Note: • When changing between processing for compression and for decompression, be sure to reset this module in advance by setting the SRST21 bit in the software reset control register 2 (SWRSTCR2) of the power-down modes.
2 to 0	REDU[2:0]	000	R/W	R	Pixel Format [Compression] 001: YCbCr422 Other than above: Setting prohibited. [Decompression] 000: YCbCr444 001: YCbCr422 110: YCbCr411 010: YCbCr420 Other than above: Error (JCU cannot process normally.)

chanhvinh.com

44.2.2 JPEG Code Command Register (JCCMD)

JCCMD sets commands. Bits of this register need not be cleared to 0 after setting a command. Multiple commands must not be set simultaneously.

Bit:	7	6	5	4	3	2	1	0
	BRST	—	—	—	—	JEND	JRST	JSRT
Initial value:	0	0	0	0	0	0	0	0
R/W(compress):	R*/W	R	R	R	R	R*/W	Undefined	R*/W
R/W(decompress):	R*/W	R	R	R	R	R*/W	R*/W	R*/W

Note: * Values read from these bits are undefined.

Bit	Bit Name	Initial Value	R/W: Compression	R/W: De-compression	Description
7	BRST	0	R*/W		Bus Reset Setting this bit to 1 resets the internal circuits. While the JCU is in operation (from setting the JPEG core process start command to writing the last output coded/image data), do not set this bit to 1. For the bus reset processing, see section 44.5, Bus Reset Processing.
6 to 3	—	All 0	R		Reserved These bits are always read as 0. The write value should always be 0.
2	JEND	0	R*/W		Interrupt Request Clear Command This bit is valid only for the interrupt sources corresponding to bits INS6, INS5, and INS3 in JINTS0. To clear an interrupt request, set this bit to 1.
1	JRST	0	Invalid	R*/W	JPEG Core Process Stop Clear Command To clear the process-stopped state caused by requests to read the image size and pixel format (enabled by the INT3 bit in JINTE0), set this bit to 1.
0	JSRT	0	R*/W		JPEG Core Process Start Command To start JPEG core processing, set this bit to 1. Do not write this bit to 1 again while the JCU is in operation.

Note: * Values read from these bits are undefined.

chanhvinh.com

44.2.3 JPEG Code Quantization Table Number Register (JCQTN)

JCQTN sets the quantization table number before compression process is started.

- To use quantization table No. 0 (JCQTBL0) as the first color component, set QT1 to B'00
- To use quantization table No. 1 (JCQTBL1) as the first color component, set QT1 to B'01
- To use quantization table No. 2 (JCQTBL2) as the first color component, set QT1 to B'10
- To use quantization table No. 3 (JCQTBL3) as the first color component, set QT1 to B'11

Bit:	7	6	5	4	3	2	1	0
	—	—	QT3[1:0]		QT2[1:0]		QT1[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W(compress):	R	R	R/W	R/W	R/W	R/W	R/W	R/W
R/W(decompress):	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W		Description
			Compression	De-compression	
7, 6	—	All 0	R		Reserved These bits are always read as 0. The write value should always be 0.
5, 4	QT3[1:0]	00	R/W	R	Quantization table number for the third color component
3, 2	QT2[1:0]	00	R/W	R	Quantization table number for the second color component
1, 0	QT1[1:0]	00	R/W	R	Quantization table number for the first color component

chanhvinh.com

44.2.4 JPEG Code Huffman Table Number Register (JCHTN)

JCHTN sets the Huffman table number (AC/DC) before compression process is started.

- To use DC/AC Huffman table No. 0 (JCHTBD0 and JCHTBA0) as the first color component, set bits HTA1 and HTD1 to B'0
- To use DC/AC Huffman table No. 1 (JCHTBD1 and JCHTBA1) as the first color component, set bits HTA1 and HTD1 to B'1

Bit:	7	6	5	4	3	2	1	0
	—	—	HTA3	HTD3	HTA2	HTD2	HTA1	HTD1
Initial value:	0	0	0	0	0	0	0	0
R/W(compress):	R	R	R/W	R/W	R/W	R/W	R/W	R/W
R/W(decompress):	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W		Description
			Compression	De-compression	
7, 6	—	All 0	R		Reserved These bits are always read as 0. The write value should always be 0.
5	HTA3	0	R/W	R	Huffman table number (AC) for the third color component
4	HTD3	0	R/W	R	Huffman table number (DC) for the third color component
3	HTA2	0	R/W	R	Huffman table number (AC) for the second color component
2	HTD2	0	R/W	R	Huffman table number (DC) for the second color component
1	HTA1	0	R/W	R	Huffman table number (AC) for the first color component
0	HTD1	0	R/W	R	Huffman table number (DC) for the first color component

44.2.5 JPEG Code DRI Upper Register (JCDRIU)

JCDRIU sets the upper bytes of the minimum coded units (MCUs) preceding an RST marker.

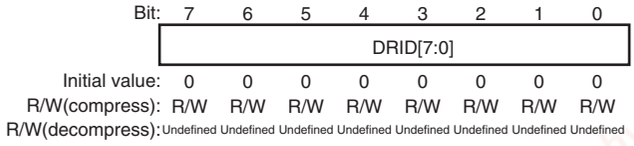
Bit:	7	6	5	4	3	2	1	0
	DRIU[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W(compress):	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
R/W(decompress):	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Name	Initial Value	R/W		Description
			Compression	De-compression	
7 to 0	DRIU[7:0]	H'00	R/W	Invalid	Upper Bytes of MCUs Preceding RST Marker When both upper and lower bytes are set to H'00, neither a DRI nor an RST marker is placed.

chanhvinh.com

44.2.6 JPEG Code DRI Lower Register (JCDRID)

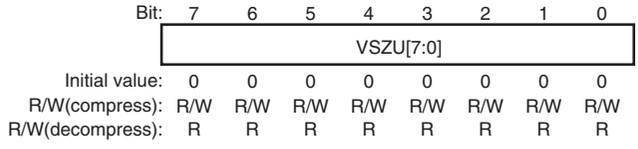
JCDRID sets the lower bytes of MCUs preceding an RST marker.



Bit	Bit Name	Initial Value	R/W		Description
			Compression	De-compression	
7 to 0	DRID[7:0]	H'00	R/W	Invalid	Lower Bytes of MCUs Preceding RST Marker When both upper and lower bytes are set to H'00, neither a DRI nor an RST marker is placed.

44.2.7 JPEG Code Vertical Size Upper Register (JCVSZU)

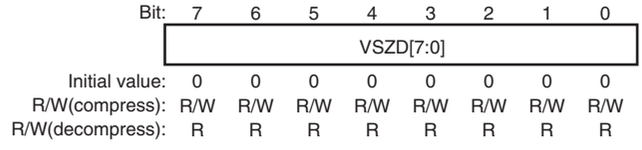
JCVSZU sets the upper bytes of the vertical image size.



Bit	Bit Name	Initial Value	R/W		Description
			Compression	De-compression	
7 to 0	VSZU[7:0]	H'00	R/W	R	Upper Bytes of Vertical Image Size In decompression process, a downloaded value from the JPEG coded data is set.

44.2.8 JPEG Code Vertical Size Lower Register (JCVSZD)

JCVSZD sets the lower bytes of the vertical image size.



Bit	Bit Name	Initial Value	R/W		Description
			Compression	De-compression	
7 to 0	VSZD[7:0]	H'00	R/W	R	Lower Bytes of Vertical Image Size In decompression process, a downloaded value from the JPEG coded data is set.

chanhvinh.com

44.2.9 JPEG Code Horizontal Size Upper Register (JCHSZU)

JCHSZU sets the upper bytes of the horizontal image size.

Bit:	7	6	5	4	3	2	1	0
	HSZU[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W(compress):	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
R/W(decompress):	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W		Description
			Compression	De-compression	
7 to 0	HSZU[7:0]	H'00	R/W	R	Upper Bytes of Horizontal Image Size In decompression process, a downloaded value from the JPEG coded data is set.

44.2.10 JPEG Coded Horizontal Size Lower Register (JCHSZD)

JCHSZD sets the lower bytes of the horizontal image size.

Bit:	7	6	5	4	3	2	1	0
	HSZD[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W(compress):	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
R/W(decompress):	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W		Description
			Compression	De-compression	
7 to 0	HSZD[7:0]	H'00	R/W	R	Lower Bytes of Horizontal Image Size In decompression process, a downloaded value from the JPEG coded data is set.

44.2.11 JPEG Code Data Count Upper Register (JCDTCU)

The upper bytes for the counted amount of data to be compressed are set to JCDTCU. The values of this register are reset before compression starts.

Bit:	7	6	5	4	3	2	1	0
	DCU[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W(compress):	R	R	R	R	R	R	R	R
R/W(decompress):	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Name	Initial Value	R/W		Description
			Compression	De-compression	
7 to 0	DCU[7:0]	H'00	R	Invalid	Upper bytes of the counted amount of data to be compressed

chanhvinh.com

44.2.12 JPEG Code Data Count Middle Register (JCDTCM)

The middle bytes for the counted amount of data to be compressed are set to JCDTCM. The values of this register are reset before compression starts.

Bit:	7	6	5	4	3	2	1	0
	DCM[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W(compress):	R	R	R	R	R	R	R	R
R/W(decompress):	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Name	Initial Value	R/W		Description
			Compression	De-compression	
7 to 0	DCM[7:0]	H'00	R	Invalid	Middle bytes of the counted amount of data to be compressed

44.2.13 JPEG Code Data Count Lower Register (JCDTCD)

The lower bytes for the counted amount of data to be compressed are set to JCDTCD. The values of this register are reset before compression starts.

Bit:	7	6	5	4	3	2	1	0
	DCD[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W(compress):	R	R	R	R	R	R	R	R
R/W(decompress):	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Name	Initial Value	R/W		Description
			Compression	De-compression	
7 to 0	DCD[7:0]	H'00	R	Invalid	Lower bytes of the counted amount of data to be compressed

chanhvinh.com

44.2.14 JPEG Interrupt Enable Register 0 (JINTE0)

JINTE0 enables interrupts.

When any of bits INT7 to INT5 is set to B'1, the INS5 bit in JINTS0 indicates B'1 as the error status upon occurrence of the compression data error, and the ERR bit in JCDERR indicates the particular error code.

Bit:	7	6	5	4	3	2	1	0
	INT7	INT6	INT5	—	INT3	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W(compress):	Undefined	Undefined	Undefined	R	Undefined	R	R	R
R/W(decompress):	R/W	R/W	R/W	R	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W		Description
			Compression	De-compression	
7	INT7	0	Invalid	R/W	This bit enables an interrupt to be generated when the number of data in the restart interval of the Huffman-coding segment is not correct in decompression. When this bit is not set to enable interrupt generation, an error code is not returned.
6	INT6	0	Invalid	R/W	This bit enables an interrupt to be generated when the total number of data in the Huffman-coding segment is not correct in decompression. When this bit is not set to enable interrupt generation, an error code is not returned.
5	INT5	0	Invalid	R/W	This bit enables an interrupt to be generated when the final number of MCU data in the Huffman-coding segment is not correct in decompression. When this bit is not set to enable interrupt generation, an error code is not returned.
4	—	0		R	Reserved This bit is always read as 0. The write value should always be 0.
3	INT3	0	Invalid	R/W	This bit enables an interrupt to be generated when it has been determined that the image size and the subsampling setting of the compressed data can be read through analyzing the data.
2 to 0	—	All 0		R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

44.2.15 JPEG Interrupt Status Register 0 (JINTS0)

JINTS0 identifies the interrupt sources.

The interrupt sources of this register should be cleared by clearing the corresponding interrupt status bits to 0 and setting the relevant bit in JCCMD appropriately.

Bit:	7	6	5	4	3	2	1	0
	—	INS6	INS5	—	INS3	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W(compress):	R	R/W*	Undefined	R	Undefined	R	R	R
R/W(decompress):	R	R/W*	R/W*	R	R/W*	R	R	R

Note: * Clear this bit by writing 0 to it.
Do not write 1 to this bit.

Bit	Bit Name	Initial Value	R/W: Compression	R/W: De-compression	Description
7	—	0	R		Reserved This bit is always read as 0. The write value should always be 0.
6	INS6	0	R/W*		This bit is set to 1 when the JCU completes compression process normally.
5	INS5	0	Invalid	R/W*	This bit is set to 1 when a compressed data error occurs.
4	—	0	R		Reserved This bit is always read as 0. The write value should always be 0.
3	INS3	0	Invalid	R/W*	This bit is set to 1 when the image size and pixel format can be read. When an interrupt occurs, this module stops processing and the state is indicated by the JCRST register. To make the JCU resume processing, set the JPEG core process stop clear command bit (JRST) in JCCMD.
2 to 0	—	All 0	R		Reserved These bits are always read as 0. The write value should always be 0.

Note: * Clear this bit by writing 0 to it. Do not write 1 to this bit.

44.2.16 JPEG Code Decode Error Register (JCDERR)

JCDERR indicates the error code to identify the type of the error which has occurred in the compressed data analysis for decompression. The values of this register are reset before the JCU starts decompression.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	ERR[3:0]			
Initial value:	0	0	0	0	1	0	1	0
R/W(compress):	R	R	R	R	Undefined	Undefined	Undefined	Undefined
R/W(decompress):	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W: Compression	R/W: De-compression	Description
7 to 4	—	All 0	R		Reserved These bits are always read as 0.
3 to 0	ERR[3:0]	1010	Invalid	R/W	Error Code (See tables 44.3 and 44.4.)

chanhvinh.com

44.2.17 JPEG Code Reset Register (JCRST)

JCRST indicates a processing-stopped state caused by requests to read the image size and pixel format (enabled by the INT3 bit in JINTE0). To resume processing, set the JPEG core process stop clear command bit (JRST) in JCCMD.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	RST
Initial value:	0	0	0	0	0	0	0	0
R/W(compress):	R	R	R	R	R	R	R	Undefined
R/W(decompress):	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W		Description
			Compression	De-compression	
7 to 1	—	All 0	R		Reserved These bits are always read as 0.
0	RST	0	Invalid	R/W	Operating State 0: State other than below 1: Suspended state caused by interrupt sources of JINTE0

chanhvinh.com

44.2.18 JPEG Interface Compression Control Register (JIFECNT)

JIFECNT controls the compression process.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W(compress):	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
R/W(decompress):	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	JOUT RINI	JOUT RCMD	JOUTC	—	JOUTSWAP[2:0]			—	DIN RINI	DIN RCMD	DIN LC	—	DINSWAP[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W(compress):	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
R/W(decompress):	R	Undefined	Undefined	Undefined	R	Undefined	Undefined	Undefined	R	Undefined	Undefined	Undefined	R	Undefined	Undefined	Undefined

Bit	Bit Name	Initial Value	R/W: Compression	R/W: De-compression	Description
31 to 15	—	All 0		R	Reserved These bits are always read as 0. The write value should always be 0.
14	JOUTRINI	0	R/W	Invalid	Address Initialization when Output Coded Data is Resumed This bit is only valid when the count mode for stopping the output of coded data is on. Set this bit before writing 1 to the data resume command bit. 0: The transfer address is not initialized when the output of coded data is restarted. 1: The transfer address is initialized when the output of coded data is restarted.
13	JOUTR CMD	0	R/W	Invalid	Output Coded Data Resume Command This bit is only valid when the count mode for stopping the output of coded data is on. Setting this bit to 1 resumes writing output coded data. This bit is always read as 0.
12	JOUTC	0	R/W	Invalid	Count Mode Setting for Stopping Output Coded Data 0: Count mode for stopping the output of coded data is off. 1: Count mode for stopping the output of coded data is on.
11	—	0		R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	JOUT SWAP[2:0]	000	R/W	Invalid	Byte/Word/Longword Swap Output coded data in compression is swapped. 000: (1) (2) (3) (4) (5) (6) (7) (8) 001: (2) (1) (4) (3) (6) (5) (8) (7) [Byte swap] 010: (3) (4) (1) (2) (7) (8) (5) (6) [Word swap] 011: (4) (3) (2) (1) (8) (7) (6) (5) [Word - byte swap] 100: (5) (6) (7) (8) (1) (2) (3) (4) [Longword swap] 101: (6) (5) (8) (7) (2) (1) (4) (3) [Longword - byte swap] 110: (7) (8) (5) (6) (3) (4) (1) (2) [Longword - word swap] 111: (8) (7) (6) (5) (4) (3) (2) (1) [Longword - word - byte swap]
7	—	0		R	Reserved This bit is always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W		Description
			Compression	De-compression	
6	DINRINI	0	R/W	Invalid	Address Initialization when Resuming Input of Image Data Lines This bit is only valid when the count mode for stopping the input of image data lines is on. Set this bit before writing 1 to the data-line resume command bit. 0: The transfer address is not initialized when the input of image data lines is restarted. 1: The transfer address is initialized when the input of image data lines is restarted.
5	DINRCMD	0	R/W	Invalid	Input Image Data Lines Resume Command This bit is valid only when the count mode for stopping the input of image data lines is on. Setting this bit to 1 resumes reading input image data. This bit is always read as 0.
4	DINLC	0	R/W	Invalid	Count Mode Setting for Stopping Input Image Data Lines 0: Count mode for stopping the input of image data lines is off. 1: Count mode for stopping the input of image data lines is on.
3	—	0	R		Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	DINSWAP [2:0]	000	R/W	Invalid	Byte/Word Swap Input image data in compression is swapped. 000: (1) (2) (3) (4) (5) (6) (7) (8) 001: (2) (1) (4) (3) (6) (5) (8) (7) [Byte swap] 010: (3) (4) (1) (2) (7) (8) (5) (6) [Word swap] 011: (4) (3) (2) (1) (8) (7) (6) (5) [Word - byte swap] 100: (5) (6) (7) (8) (1) (2) (3) (4) [Longword swap] 101: (6) (5) (8) (7) (2) (1) (4) (3) [Longword - byte swap] 110: (7) (8) (5) (6) (3) (4) (1) (2) [Longword - word swap] 111: (8) (7) (6) (5) (4) (3) (2) (1) [Longword - word - byte swap]

44.2.19 JPEG Interface Compression Source Address Register (JIFESA)

JIFESA sets the source address of the input image data. This register should be set in 8-byte units.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ESA[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W(compress):	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
R/W(decompress):	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ESA[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W(compress):	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R
R/W(decompress):	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Name	Initial Value	R/W		Description
			Compression	De-compression	
31 to 3	ESA[31:3]	H'0000	R/W	Invalid	Input Image Data Source Address (in 8-byte units)
2 to 0	ESA[2:0]	0000	R		The lower three bits should be set to 0.

chanhvinh.com

44.2.20 JPEG Interface Compression Line Offset Register (JIFESOFST)

JIFESOFST sets the line offset of the input image data (refer to section 44.3.4, Storing Image Data). This register should be set in 8-byte units.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W(compress):	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
R/W(decompress):	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ESMW[14:0]														
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W(compress):	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R
R/W(decompress):	R	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Name	Initial Value	R/W: Compression	R/W: De-compression	Description
31 to 15	—	All 0	R		Reserved These bits are always read as 0. The write value should always be 0.
14 to 3	ESMW [14:3]	H'0000	R/W	Invalid	Input Image Data Lines Offset (in 8-byte units) The lower three bits should be set to 0.
2 to 0	ESMW[2:0]		R		

44.2.21 JPEG Interface Compression Destination Address Register (JIFEDA)

JIFEDA sets the destination address of the output coded data. This register should be set in 8-byte units.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EDA[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W(compress):	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
R/W(decompress):	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EDA[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W(compress):	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R
R/W(decompress):	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Name	Initial Value	R/W: Compression	R/W: De-compression	Description
31 to 3	EDA[31:3]	H'0000	R/W	Invalid	Output Coded Data Destination Address (in 8-byte units) The lower three bits should be set to 0.
2 to 0	EDA[2:0]	0000	R		

chanhvinh.com

44.2.22 JPEG Interface Compression Source Line Count Register (JIFESLC)

JIFESLC sets the number of input image data lines when the count mode for stopping the input of image data lines is on (the DINLC bit in JIFECNT is set to 1). This register should be set in 8-line units.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
R/W(compress):	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
R/W(decompress):	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LINES[15:0]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
R/W(compress):	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R
R/W(decompress):	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Name	Initial Value	R/W: Compression	R/W: De-compression	Description
31 to 16	—	H'FFF8	R		Reserved Values read from these bits are undefined. The write value should always be 0.
15 to 3	LINES[15:3]	H'FFF8	R/W	Invalid	Number of Input Image Data Lines to be Read (in 8-line units)
2 to 0	LINES[2:0]		R		The lower three bits should be set to 0.

44.2.23 JPEG Interface Compression Destination Count Register (JIFEDDC)

JIFEDDC sets the amount of output coded data when the count mode for stopping the output of coded data is on (the JOUTC bit in JIFECNT is set to 1). This register should be set in 8-byte units.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W(compress):	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
R/W(decompress):	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	JDATAS[15:0]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
R/W(compress):	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R
R/W(decompress):	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Name	Initial Value	R/W: Compression	R/W: De-compression	Description
31 to 16	—	H'FFF8	R	Invalid	Reserved Values read from these bits are undefined. The write value should always be 0.
15 to 3	JDATAS[15:3]	H'FFF8	R/W	Invalid	Amount of Output Coded Data to be Written (in 8-byte units)
2 to 0	JDATAS[2:0]		R		The lower three bits should be set to 0.

chanhvinh.com

44.2.24 JPEG Interface Decompression Control Register (JIFDCNT)

JIFDCNT controls the decompression process.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	VINTER[1:0]		HINTER[1:0]		OPF[1:0]		—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W(compress):	R	R	Undefined		Undefined		Undefined		R	R	R	R	R	R	R	R
R/W(decompress):	R	R	R/W		R/W		R/W		R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	JINRINI	JINRCMD	JINC	—	JINSWAP[2:0]			—	DOUTRINI	DOUTRCMD	DOUTLD	—	DOUTSWAP[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W(compress):	R	Undefined			R	Undefined			R	Undefined			R	Undefined		
R/W(decompress):	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W: Compression	R/W: De-compression	Description
31, 30	—	All 0		R	Reserved These bits are always read as 0. The write value should always be 0.
29, 28	VINTER[1:0]	00	Invalid	R/W	Vertical Subsampling Subsamples vertical output image data. 00: No subsampling 01: Subsamples output data into 1/2. 10: Subsamples output data into 1/4. 11: Subsamples output data into 1/8.
27, 26	HINTER[1:0]	00	Invalid	R/W	Horizontal Subsampling Subsamples horizontal output image data. 00: No subsampling 01: Subsamples output data into 1/2. 10: Subsamples output data into 1/4. 11: Subsamples output data into 1/8.
25, 24	OPF[1:0]	00	Invalid	R/W	Specifies output image data pixel format. 00: YCbCr422 01: ARGB8888 10: RGB565 11: Setting prohibited
23 to 15	—	All 0		R	Reserved These bits are always read as 0. The write value should always be 0.
14	JINRINI	0	Invalid	R/W	Address Initialization when Input Coded Data is Resumed This bit is only valid when the count mode for stopping the input of coded data is on. Set this bit before writing 1 to the data resume command bit. 0: The transfer address is not initialized when the input of coded data is restarted. 1: The transfer address is initialized when the input of coded data is restarted.
13	JINRCMD	0	Invalid	R/W	Input Coded Data Resume Command This bit is valid only when the count mode for stopping the input of coded data is on. Setting this bit to 1 resumes reading input coded data. This bit is always read as 0.
12	JINC	0	Invalid	R/W	Count Mode Setting for Stopping Input Coded Data 0: Count mode for stopping the input of coded data is off. 1: Count mode for stopping the input of coded data is on.
11	—	0		R	Reserved This bit is always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W		Description
			Compression	De-compression	
10 to 8	JINSWAP [2:0]	000	Invalid	R/W	Byte/Word/Longword Swap Input coded data in decompression is swapped. 000: (1) (2) (3) (4) (5) (6) (7) (8) 001: (2) (1) (4) (3) (6) (5) (8) (7) [Byte swap] 010: (3) (4) (1) (2) (7) (8) (5) (6) [Word swap] 011: (4) (3) (2) (1) (8) (7) (6) (5) [Word - byte swap] 100: (5) (6) (7) (8) (1) (2) (3) (4) [Longword swap] 101: (6) (5) (8) (7) (2) (1) (4) (3) [Longword - byte swap] 110: (7) (8) (5) (6) (3) (4) (1) (2) [Longword - word swap] 111: (8) (7) (6) (5) (4) (3) (2) (1) [Longword - word - byte swap]
7	—	0	R		Reserved This bit is always read as 0. The write value should always be 0.
6	DOUTRINI	0	Invalid	R/W	Address Initialization when Resuming Output of Image Data Lines This bit is only valid when the count mode for stopping the output of image data lines is on. Set this bit before writing 1 to the data-line resume command bit. 0: The transfer address is not initialized when the output of lines of image data is restarted. 1: The transfer address is initialized when the output of lines of image data is restarted.
5	DOUTRCMD	0	Invalid	R/W	Output Image Data Lines Resume Command This bit is valid only when the count mode for stopping the output of image data lines is on. Setting this bit to 1 resumes writing image data. This bit is always read as 0.
4	DOUTLC	0	Invalid	R/W	Count Mode for Stopping Output Image Data Lines 0: Count mode for stopping the output of image data lines is off. 1: Count mode for stopping the output of image data lines is on.
3	—	0	R		Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	DOUTSWAP [2:0]	000	Invalid	R/W	Byte/Word Swap Output image data in decompression is swapped. 000: (1) (2) (3) (4) (5) (6) (7) (8) 001: (2) (1) (4) (3) (6) (5) (8) (7) [Byte swap] 010: (3) (4) (1) (2) (7) (8) (5) (6) [Word swap] 011: (4) (3) (2) (1) (8) (7) (6) (5) [Word - byte swap] 100: (5) (6) (7) (8) (1) (2) (3) (4) [Longword swap] 101: (6) (5) (8) (7) (2) (1) (4) (3) [Longword - byte swap] 110: (7) (8) (5) (6) (3) (4) (1) (2) [Longword - word swap] 111: (8) (7) (6) (5) (4) (3) (2) (1) [Longword - word - byte swap]

chanhvinh.com

44.2.25 JPEG Interface Decompression Source Address Register (JIFDSA)

JIFDSA sets the source address of the input coded data. This register should be set in 8-byte units.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DSA[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W(compress):	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W(decompress):	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DSA[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W(compress):	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W(decompress):	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W: Compression	R/W: De-compression	Description
31 to 3	DSA[31:3]	H'0000	Invalid	R/W	Input Coded Data Source Address (in 8-byte units)
2 to 0	DSA[2:0]	0000		R	The lower three bits should be set to 0.

44.2.26 JPEG Interface Decompression Line Offset Register (JIFDDOFST)

JIFDDOFST sets the line offset of the output image data to be transferred to the external buffer (refer to section 44.3.4, Storing Image Data). This register should be set in 8-byte units.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W(compress):	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
R/W(decompress):	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	DDMW[14:0]														
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W(compress):	R	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W(decompress):	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W: Compression	R/W: De-compression	Description
31 to 16	—	All 0	R		Reserved These bits are always read as 0. The write value should always be 0.
15 to 3	DDMW[14:3]	H'0000	Invalid	R/W	Output Image Data Lines Offset (in 8-byte units)
2 to 0	DDMW[2:0]			R	The lower three bits should be set to 0.

chanhvinh.com

44.2.27 JPEG Interface Decompression Destination Address Register (JIFDDA)

JIFDDA sets the destination address of the output image data. This register should be set in 8-byte units.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DDA[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W(compress):	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W(decompress):	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DDA[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W(compress):	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W(decompress):	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W		Description
			Compression	De-compression	
31 to 3	DDA[31:3]	H'0000	Invalid	R/W	Output Image Data Destination Address (in 8-byte units)
2 to 0	DDA[2:0]	0000		R	The lower three bits should be set to 0.

chanhvinh.com

44.2.28 JPEG Interface Decompression Source Data Count Register (JIFDSDC)

JIFDSDC sets the amount of input coded data when the count mode for stopping the input of coded data is on (the JINC bit in JIFDCNT is set to 1). This register should be set in 8-byte units.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
R/W(compress):	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
R/W(decompress):	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	JDATAS[15:0]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
R/W(compress):	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W(decompress):	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W: Compression	R/W: De-compression	Description
31 to 16	—	H'FFF8	R		Reserved Values read from these bits are undefined. The write value should always be 0.
15 to 3	JDATAS[15:3]	H'FFF8	Invalid	R/W	Amount of Input Coded Data to be Read (in 8-byte units)The lower three bits should be set to 0.
2 to 0	JDATAS[2:0]			R	

44.2.29 JPEG Interface Decompression Destination Line Count Register (JIFDDLC)

JIFDDLC sets the number of lines of output image data when the count mode for stopping the output of image data lines is on (the DOUTLC bit in JIFECNT is set to 1). This register is used to set the number of lines of output image data in MCU units.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
R/W(compress):	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
R/W(decompress):	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LINES[15:0]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
R/W(compress):	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W(decompress):	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W: Compression	R/W: De-compression	Description
31 to 16	—	H'FFF8	R		Reserved Values read from these bits are undefined. The write value should always be 0.
15 to 3	LINES[15:3]	H'FFF8	Invalid	R/W	Specify the number of lines of output image data to be written. The setting is in MCU units. When data are to be output in YCbCr444, YCbCr422 or YCbCr411 format, the number of lines of output image data is <this setting> x 1. When data are to be output in YCbCr420 format, the number of lines of output image data is <this setting> x 2. The lower three bits should be set to 0.
2 to 0	LINES[2:0]			R	

chanhvinh.com

44.2.30 JPEG Interface Decompression α Set Register (JIFDADT)

JIFDADT is used to set the α value when output is in ARGB8888 format.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W(compress):	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
R/W(decompress):	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ALPHA[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W(compress):	R	R	R	R	R	R	R	R	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W(decompress):	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W		Description
			Compression	De-compression	
31 to 8	—	All 0	R		Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	ALPHA[7:0]	H'00	Invalid	R/W	Setting of the α value for output in ARGB8888 format.

chanhvinh.com

44.2.31 JPEG Interrupt Enable Register 1 (JINTE1)

JINTE1 enables interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W(compress):	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
R/W(decompress):	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	CBTEN	DIN LEN	JOU TEN	—	DBTEN	JINEN	DOU TLEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W(compress):	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	Undefined	Undefined	Undefined
R/W(decompress):	R	R	R	R	R	R	R	R	R	Undefined	Undefined	Undefined	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W: Compression	R/W: De-compression	Description
31 to 7	—	All 0	R		Reserved These bits are always read as 0. The write value should always be 0.
6	CBTEN	0	R/W	Invalid	Enables or disables a data transfer processing interrupt request (JDTI) when the CBTF bit in JINTS1 is set to 1. 0: Disables an interrupt request. 1: Enables an interrupt request.
5	DINLEN	0	R/W	Invalid	Enables or disables a data transfer processing interrupt request (JDTI) when the DINLF bit in JINTS1 is set to 1. 0: Disables an interrupt request. 1: Enables an interrupt request.
4	JOUTEN	0	R/W	Invalid	Enables or disables a data transfer processing interrupt request (JDTI) when the JOUTF bit in JINTS1 is set to 1. 0: Disables an interrupt request. 1: Enables an interrupt request.
3	—	0	R		Reserved This bit is always read as 0. The write value should always be 0.
2	DBTEN	0	Invalid	R/W	Enables or disables a data transfer processing interrupt request (JDTI) when the DBTF bit in JINTS1 is set to 1. 0: Disables an interrupt request. 1: Enables an interrupt request.
1	JINEN	0	Invalid	R/W	Enables or disables a data transfer processing interrupt request (JDTI) when the JINF bit in JINTS1 is set to 1. 0: Disables an interrupt request. 1: Enables an interrupt request.
0	DOUTLEN	0	Invalid	R/W	Enables or disables a data transfer processing interrupt request (JDTI) when the DOUTLF bit in JINTS1 is set to 1. 0: Disables an interrupt request. 1: Enables an interrupt request.

chanhvinh.com

44.2.32 JPEG Interrupt Status Register 1 (JINTS1)

JINTS1 indicates the interrupt sources.

The interrupt sources of this register should be cleared by writing 0 to this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W(compress):	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
R/W(decompress):	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	CBTF	DINLF	JOUTF	—	DBTF	JINF	DOUTLF
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W(compress):	R	R	R	R	R	R	R	R	R	R/W*	R/W*	R/W*	R	Undefined	Undefined	Undefined
R/W(decompress):	R	R	R	R	R	R	R	R	R	Undefined	Undefined	Undefined	R	R/W*	R/W*	R/W*

Note: * When the bit is read as 1, write 0 to clear it.
When the bit is read as 0, write 1 to it.

Bit	Bit Name	Initial Value	R/W		Description
			Compression	De-compression	
31 to 7	—	All 0	R		Reserved These bits are always read as 0. The write value should always be 0.
6	CBTF	0	R/W*	Invalid	This bit is set to 1 when the last output coded data is written in compression.
5	DINLF	0	R/W*	Invalid	This bit is set to 1 when the number of input image data lines indicated by JIFESLC is read in compression. This bit is valid only when the DINLC bit in JIFECNT is set to 1.
4	JOUTF	0	R/W*	Invalid	This bit is set to 1 when the amount of output coded data indicated by JIFEDDC is written in compression. This bit is valid only when the JOUTC bit in JIFECNT is set to 1.
3	—	0	R		Reserved This bit is always read as 0. The write value should always be 0.
2	DBTF	0	Invalid	R/W*	This bit is set to 1 when the last output image data is written in decompression.
1	JINF	0	Invalid	R/W*	This bit is set to 1 when the amount of input coded data indicated by JIFDSDC is read in decompression. This bit is valid only when the JINC bit in JIFDCNT is set to 1.
0	DOUTLF	0	Invalid	R/W*	In decompression, this bit is set to 1 when the number of lines of output image data indicated by JIFDDLC have been written. This bit is only valid when the DOUTLC bit in JIFDCNT is set to 1.

Note: * When the bit is read as 1, write 0 to clear it.
When the bit is read as 0, write 1 to it.

chanhvinh.com

44.2.33 JPEG Input Image Data CbCr Range Setting Register (JIFESVSZ)

JIFESVSZ sets the CbCr range of input image data.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W(compress):	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
R/W(decompress):	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DINY CHG	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W(compress):	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
R/W(decompress):	Undefined	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W: Compression	R/W: De-compression	Description
31 to 16	—	All 0	R		Reserved These bits are always read as 0. The write value should always be 0.
15	DINYCHG	0	R/W	Invalid	Input Image Data CbCr Range Setting 0: Range from -128 to 127 1: Range from 0 to 255
14 to 0	—	All 0	R		Reserved These bits are always read as 0. The write value should always be 0.

44.2.34 JPEG Output Image Data CbCr Range Setting Register (JIFESHSZ)

JIFESHSZ sets the CbCr range of output image data.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W(compress):	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
R/W(decompress):	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DOUTY CHG	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W(compress):	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
R/W(decompress):	Undefined	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W: Compression	R/W: De-compression	Description
31 to 16	—	All 0	R		Reserved These bits are always read as 0. The write value should always be 0.
15	DOUTYCHG	0	Invalid	R/W	Output Image Data CbCr Range Setting 0: Range from -128 to 127 1: Range from 0 to 255
14 to 0	—	All 0	R		Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

44.3 Operation

44.3.1 Compression

(1) Overview of Processing

The compression process flows are described below.

1. The JPEG core is activated.
   A marker is output. (After a marker is output, image data can be input.)
   Approximately 30,000 cycles (necessary for making SOI to SOS markers)
2. Image data is transferred in MCUs from the external buffer to the JCU.
   If the count mode for stopping the input of image data lines is on, reading is stopped each time the number of lines set in JIFESLC is read. Reading is resumed by setting the DINRCMD bit in JIFECNT to 1.
   When the DINRINI bit in JIFECNT is zero, the addresses for reading on resumption are continued from the addresses in the previous round of transfer.
   When the DINRINI bit is one, the address set in JIFESA is used on resumption.
   Reading is also stopped when one frame of image data is completely transferred.
   If the count mode for stopping the input of image data lines is off, reading is continued until one frame of image data is completely transferred.
3. Image data is input to the JPEG core.
   The input data is processed in MCUs at any time in the JPEG core.
4. Coded data is transferred from the JCU to the external buffer.
   When the count mode for stopping the output of coded data is on, writing is stopped each time the amount of coded data set in JIFEDDC is written. Writing is resumed by setting the JOUTRCMD bit in JIFECNT to 1.
   When the JOUTRINI bit in JIFECNT is zero, the addresses for writing on resumption are continued from the addresses in the previous round of transfer.
   When the JOUTRINI bit is one, the address set in JIFEDA is used on resumption.
   Writing is also stopped when one frame of coded data is completely transferred.
   If the count mode for stopping the output of coded data is off, writing is continued until one frame of coded data is completely transferred.
5. Compression is completed after one frame of data is processed completely.

chanhvinh.com

(2) Flowchart (Compression)

(a) Initial Settings

After completing the JPEG core settings and input/output buffer settings and transferring image data to the external buffer, activate this module by setting the JSRT bit in JCCMD to 1. After the JCU has been activated, the JPEG markers (SOI to SOS) are generated and output. It takes approximately 30,000 cycles to generate the markers.

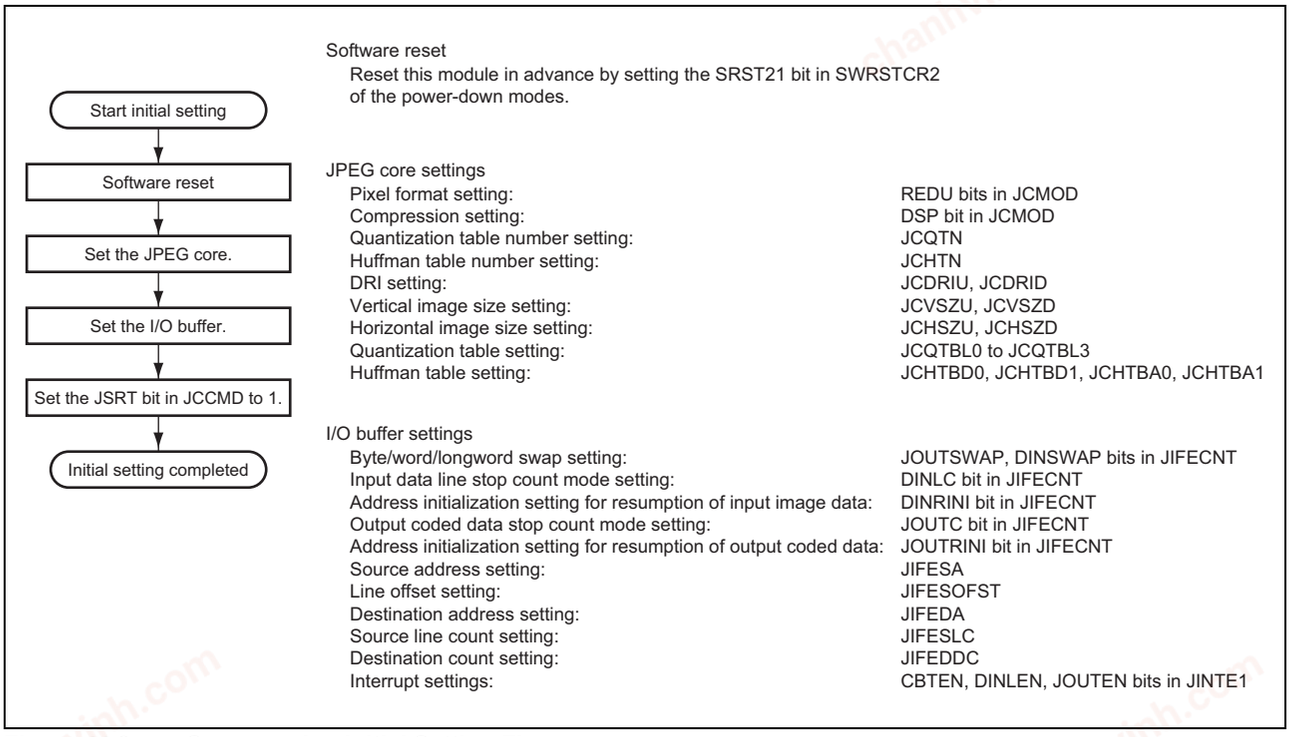


Figure 44.2 Compression Initial Setting Flow

(b) Compression Process

The compression process flows are described below.

- When JPEG compression process has been completed, the INS6 bit in JINTS0 is set to 1. However, the JCU continues processing since the coded data remains to be transferred. The CBTF bit in JINTS1 is set to 1 when the last coded data is transferred. The interrupt source is cleared by writing 0 to the INS6 bit. However, the interrupt request asserted by this interrupt source cannot be cleared by writing 0 to the INS6 bit. Set an interrupt request clear command (by setting the JEND bit in JCCMD to 1) to clear the interrupt request.
- When the JCU has completed compression and all coded data has been transferred, the CBTF flag in JINTS1 is set to 1. When the CBTEN bit in JINTE1 is 1 here, an interrupt is generated. The interrupt source is cleared by writing 0 to the CBTF flag.
- If the count mode for stopping image data lines is on, when the specified number of image data lines set in JIFESLC has been read, the DINLF flag in JINTS1 is set to 1, and reading is stopped. When the DINLEN bit in JINTE1 is 1 here, an interrupt is generated. An interrupt source is cleared by writing 0 to the DINLEN bit. Setting the DINRCMD bit in JIFECNT to 1 resumes reading.
  When the DINRINI bit in JIFECNT is zero, the addresses for reading on resumption are continued from the addresses in the previous round of transfer.
  When the DINRINI bit is one, the address set in JIFESA is used on resumption.
- If the count mode for stopping the output of coded data is on, when the specified amount of coded data set in JIFEDDC has been written, the JOUTF flag in JINTS1 is set to 1, and writing is stopped. When the JOUTEN bit in JINTE1 is 1 here, an interrupt is generated. An interrupt source is cleared by writing 0 to the JOUTF bit. Setting the JOUTRCMD bit in JIFECNT to 1 resumes writing.

chanhvinh.com

When the JOUTRINI bit in JIFECNT is zero, the addresses for writing on resumption are continued from the addresses in the previous round of transfer.
When the JOUTRINI bit is one, the address set in JIFEDA is used on resumption.

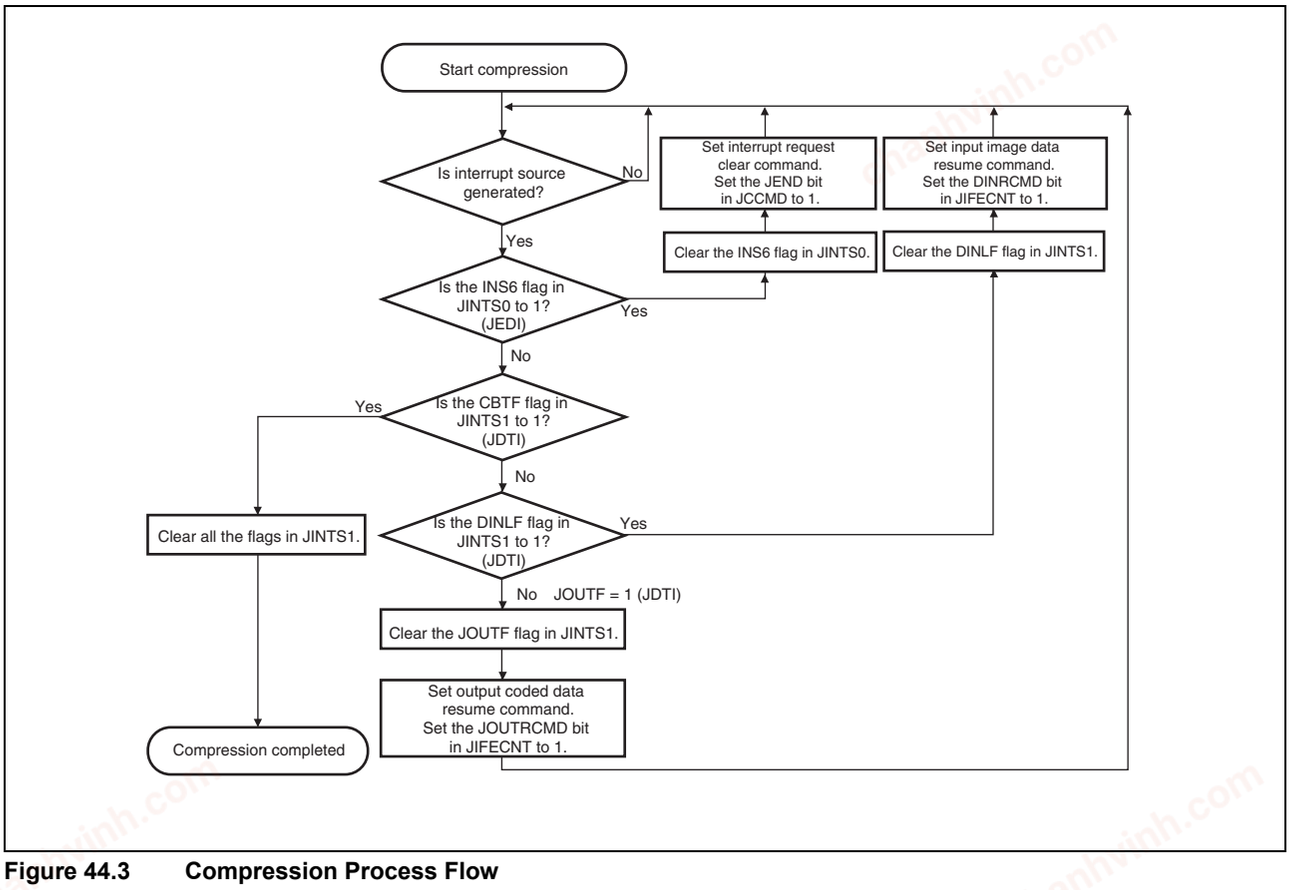


Figure 44.3 Compression Process Flow

(3) JPEG Coded Data Format

Figure 44.4 shows the data output stream in compression. The amount of coded data from SOI to EOI is indicated by JCDTCU, JCDTCM, and JCDTCD. When both JCDRIU and JCDRID are set to H'0000 0000, the following markers are not output.

- DRI marker
- RST marker (in compressed image data)

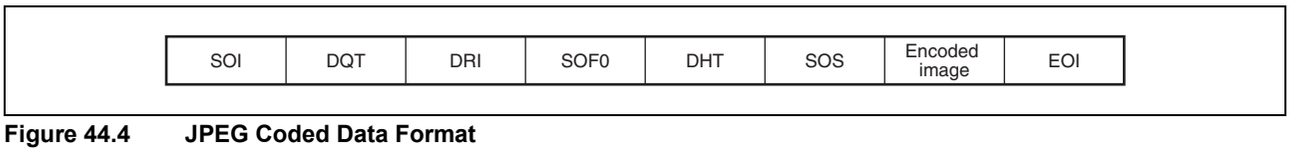


Figure 44.4 JPEG Coded Data Format

DQT: Not output for unused table.
DHT: Output in order DC0, AC0, DC1, and AC1. Not output for unused table.
SOF0: Component identifiers are C1 = first color component, C2 = second color component, and C3 = third color component.
SOS: Scan component selectors are CS1 = first color component, CS2 = second color component, and CS3 = third color component.

chanhvinh.com

Header Volume (Reference):

- SOI: 2 bytes (FFD8)
- DQT: 134 bytes when two quantization tables are used, 199 bytes when three quantization tables are used (±65 bytes/table increase or decrease)
- DRI: 6 bytes
- SOF0: 19 bytes (4:2:2)
- DHT: 420 bytes (two tables are used)
- SOS: 14 bytes (4:2:2)
- EOI: 2 bytes (FFD9)

(4) Table Setting

(a) Quantization Table Specification

The order of addresses shown in 8 × 8 blocks corresponds to that of the register addresses. Do not access this table while the JCU is in processing.

Table 44.2 Quantization Table

00	01	02	03	04	05	06	07
08	09	0A	0B	0C	0D	0E	0F
10	11	12	13	14	15	16	17
18	19	1A	1B	1C	1D	1E	1F
20	21	22	23	24	25	26	27
28	29	2A	2B	2C	2D	2E	2F
30	31	32	33	34	35	36	37
38	39	3A	3B	3C	3D	3E	3F

JCQTBL0 (H'E801 7100) = H'00
JCQTBL0 (H'E801 7101) = H'01
JCQTBL0 (H'E801 7102) = H'02
JCQTBL0 (H'E801 7103) = H'03
:
JCQTBL0 (H'E801 713F) = H'3F

(b) Huffman Table Specification

Examples of the Huffman table specification given in the ITU-T T81 Annex K.3.3 recommended by JPEG are shown below. In compression, the following settings must be specified for all the codes so that Huffman codes can be generated for all the group numbers.

- DC Huffman table: The number of codes for each code length is 12.
  The group numbers in order of frequency of occurrence are 12.
- AC Huffman table: The number of codes for each code length is 162.
  The zero run length/the group numbers in order of frequency of occurrence are 162.

chanhvinh.com

Do not access the following tables while the JCU is in processing. In particular, read access is prohibited.

- able K.3/T81
  JCHTBD0 (H'E801 7200) = H'00
  JCHTBD0 (H'E801 7201) = H'01
  JCHTBD0 (H'E801 7202) = H'05
  JCHTBD0 (H'E801 7203) = H'01
  :
  JCHTBD0 (H'E801 721B) = H'0B

- Table K.4/T81
  JCHTBD1 (H'E801 7300) = H'00
  JCHTBD1 (H'E801 7301) = H'03
  JCHTBD1 (H'E801 7302) = H'01
  JCHTBD1 (H'E801 7303) = H'01
  :
  JCHTBD1 (H'E801 731B) = H'0B

- Table K.5/T81
  JCHTBA0 (H'E801 7220) = H'00
  JCHTBA0 (H'E801 7221) = H'02
  JCHTBA0 (H'E801 7222) = H'01
  JCHTBA0 (H'E801 7223) = H'03
  :
  JCHTBA0 (H'E801 72D1) = H'FA

- Table K.6/T81
  JCHTBA1 (H'E801 7320) = H'00
  JCHTBA1 (H'E801 7321) = H'02
  JCHTBA1 (H'E801 7322) = H'01
  JCHTBA1 (H'E801 7323) = H'02
  :
  JCHTBA1 (H'E801 73D1) = H'FA

chanhvinh.com

(5) Input Pixel Format

Image data in the YCbCr422 format can be input to this module. Allocation of data in the YCbCr422 format can be changed by the DINSWAP bits in JIFECNT as shown below.

- When the DINSWAP bits = 000

b63 b56	b55 b48	b47 b40	b39 b32	b31 b24	b23 b16	b15 b8	b7 b0
Y0 8 bits	Cb0 8 bits	Y1 8 bits	Cr0 8 bits	Y2 8 bits	Cb1 8 bits	Y3 8 bits	Cr1 8 bits

- When the DINSWAP bits = 001

b63 b56	b55 b48	b47 b40	b39 b32	b31 b24	b23 b16	b15 b8	b7 b0
Cb0 8 bits	Y0 8 bits	Cr0 8 bits	Y1 8 bits	Cb1 8 bits	Y2 8 bits	Cr1 8 bits	Y3 8 bits

- When the DINSWAP bits = 010

b63 b56	b55 b48	b47 b40	b39 b32	b31 b24	b23 b16	b15 b8	b7 b0
Y1 8 bits	Cr0 8 bits	Y0 8 bits	Cb0 8 bits	Y3 8 bits	Cr1 8 bits	Y2 8 bits	Cb1 8 bits

- When the DINSWAP bits = 100

b63 b56	b55 b48	b47 b40	b39 b32	b31 b24	b23 b16	b15 b8	b7 b0
Y2 8 bits	Cb1 8 bits	Y3 8 bits	Cr1 8 bits	Y0 8 bits	Cb0 8 bits	Y1 8 bits	Cr0 8 bits

- When the DINSWAP bits = 101

b63 b56	b55 b48	b47 b40	b39 b32	b31 b24	b23 b16	b15 b8	b7 b0
Cb1 8 bits	Y2 8 bits	Cr1 8 bits	Y3 8 bits	Cb0 8 bits	Y0 8 bits	Cr0 8 bits	Y1 8 bits

- When the DINSWAP bits = 110

b63 b56	b55 b48	b47 b40	b39 b32	b31 b24	b23 b16	b15 b8	b7 b0
Y3 8 bits	Cr1 8 bits	Y2 8 bits	Cb1 8 bits	Y1 8 bits	Cr0 8 bits	Y0 8 bits	Cb0 8 bits

- When the DINSWAP bits = 111

b63 b56	b55 b48	b47 b40	b39 b32	b31 b24	b23 b16	b15 b8	b7 b0
Cr1 8 bits	Y3 8 bits	Cb1 8 bits	Y2 8 bits	Cr0 8 bits	Y1 8 bits	Cb0 8 bits	Y0 8 bits

(6) Output Coded Data

In the case of compression, coded data are output. This module handles the output of coded data in 16-bit units. For this reason, if the coded data have an odd code length (are fractional), the final code for output will be H'D9FF.
The JOUTSWAP bits in JIFECNT can be used to alter the arrangement of coded data in the output.

chanhvinh.com

44.3.2 Decompression

(1) Overview of Processing

The decompression process flows are described below.

1. The JPEG core is activated.
2. Coded data is transferred from the external buffer to the JCU.
   If the count mode for stopping the input of coded data is on, reading is stopped each time the amount of coded data set in JIFDSLC is read. Reading is resumed by setting the JINRCMD bit in JIFDCNT to 1. When the JINRINI bit in JIFDCNT is zero, the addresses for reading on resumption are continued from the addresses in the previous round of transfer.
   When the JINRINI bit is one, the address set in JIFDSA is used on resumption. Reading is stopped when the end of the coded data is detected.
   If the count mode for stopping the input of coded data is off, reading is continued until the end of code is detected. With this module, more coded data may be read than the coded data size since coded data reading is continued until the end of code is detected.
3. Coded data is input to the JPEG core.
   The input data is processed in MCUs at any time in the JPEG core.
4. Image data is transferred in MCUs from the JCU to the external buffer.
   When the count mode for stopping the output of image data lines is on, writing is stopped each time the number of image data lines set in JIFDDLC is written. Writing is resumed by setting the DOUTRCMD bit in JIFECNT to 1. When the DOUTRINI bit in JIFDCNT is zero, the addresses for writing on resumption are continued from the addresses in the previous round of transfer.
   When the DOUTRINI bit is one, the address set in JIFDDA is used on resumption. Writing is stopped when one frame of image data is completely transferred.
   If the count mode for stopping the output of image data lines is off, writing is continued until one frame of image data is completely transferred.
5. Decompression is completed after one frame of data is processed completely.

chanhvinh.com

(a) Initial Settings

- When the INT3 bit in JINTE0 is set to 0:
  After completing the JPEG core settings and input/output buffer settings and transferring coded data to the external buffer, activate this module by setting the JSRT bit in JCCMD to 1.
- When the INT3 bit in JINTE0 is set to 1:
  After completing the JPEG core settings and input buffer settings and transferring coded data to the external buffer, activate this module by setting the JSRT bit in JCCMD to 1.
  When the image size and pixel format become readable after the coded data has been decompressed, the INS3 bit in JINTS0 is set. At this time, decompression is temporarily stopped.
  After the image size and pixel format have been read, set the output buffer.
  Setting the JRST bit in JCCMD to 1 after interrupt handling resumes decompression.

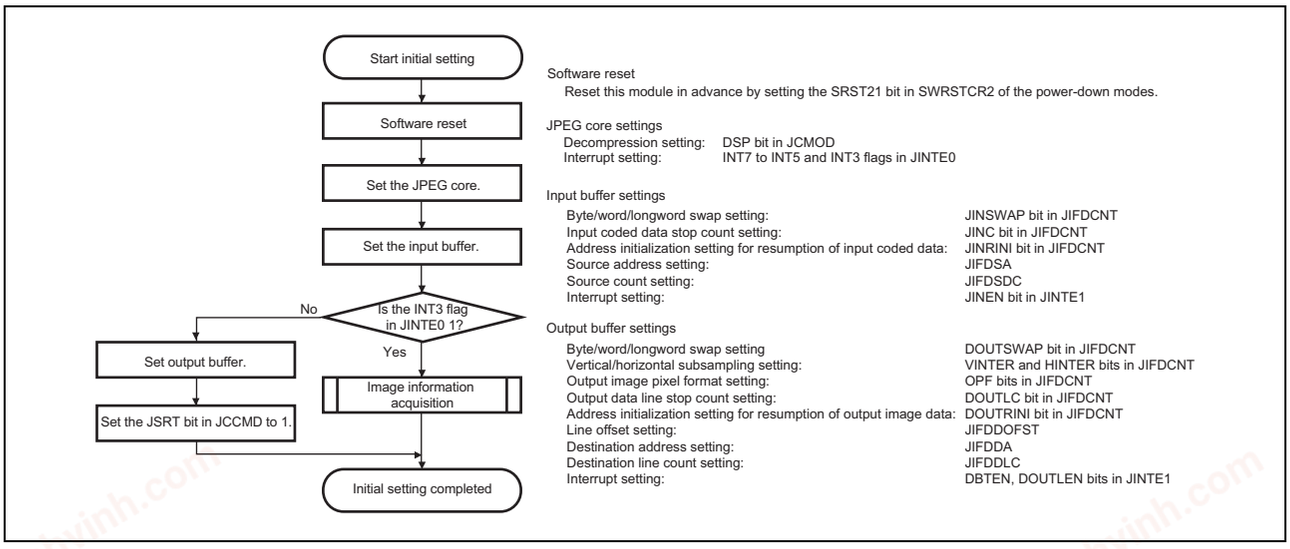


Figure 44.5 Decompression Initial Setting Flow

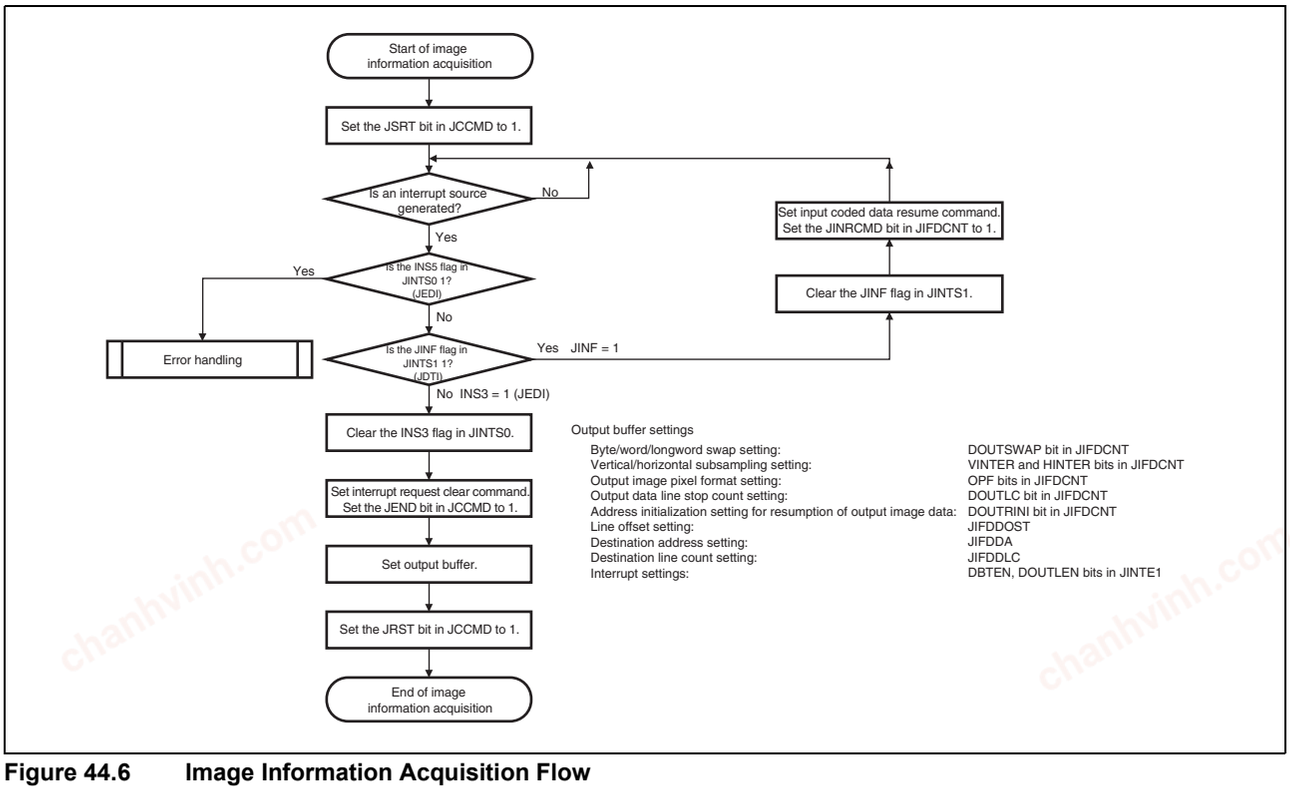


Figure 44.6 Image Information Acquisition Flow

chanhvinh.com

(b) Decompression Process

The decompression process flows are described below.

- When JPEG decompression process has been completed, the INS6 bit in JINTS0 is set to 1. However, the JCU continues processing since the image data remains to be transferred. The DBTF bit in JINTS1 is set to 1 when the last image data is transferred. The interrupt source is cleared by writing 0 to the INS6 bit. However, the interrupt request asserted by this interrupt source cannot be cleared by writing 0 to the INS6 bit. Set an interrupt request clear command (by setting the JEND bit in JCCMD to 1) to clear the interrupt request.
- When the JCU has completed decompression process and all image data has been transferred, the DBTF flag in JINTS1 is set to 1. When the DBTEN bit in JINTE1 is 1 here, an interrupt is generated. The interrupt source is cleared by writing 0 to the DBTF flag.
- If the count mode for stopping input coded data is on, when the specified amount of coded data set in JIFDSDC have been read, the JINF flag in JINTS1 is set to 1, and reading is stopped. When the JINEN bit in JINTE1 is 1 here, an interrupt is generated. An interrupt source is cleared by writing 0 to the JINF bit. Setting the JINRCMD bit in JIFDCNT to 1 resumes reading.
  When the JINRINI bit in JIFDCNT is zero, the addresses for reading on resumption are continued from the addresses in the previous round of transfer.
  When the JINRINI bit is one, the address set in JIFDSA is used on resumption.
- If the count mode for stopping the output image data is on, when the specified number of image data lines set in JIFDDLC have been written, the DOUTLF flag in JINT1 is set to 1, and writing is stopped. When the DOUTLEN bit in JINTE1 is 1 here, an interrupt is generated. An interrupt source is cleared by writing 0 to the DOUTLF bit. Setting the DOUTRCMD bit in JIFDCNT to 1 resumes writing.
  When the DOUTRINI bit in JIFDCNT is zero, the addresses for writing on resumption are continued from the addresses in the previous round of transfer.
  When the DOUTRINI bit is one, the address set in JIFDDA is used on resumption.

chanhvinh.com

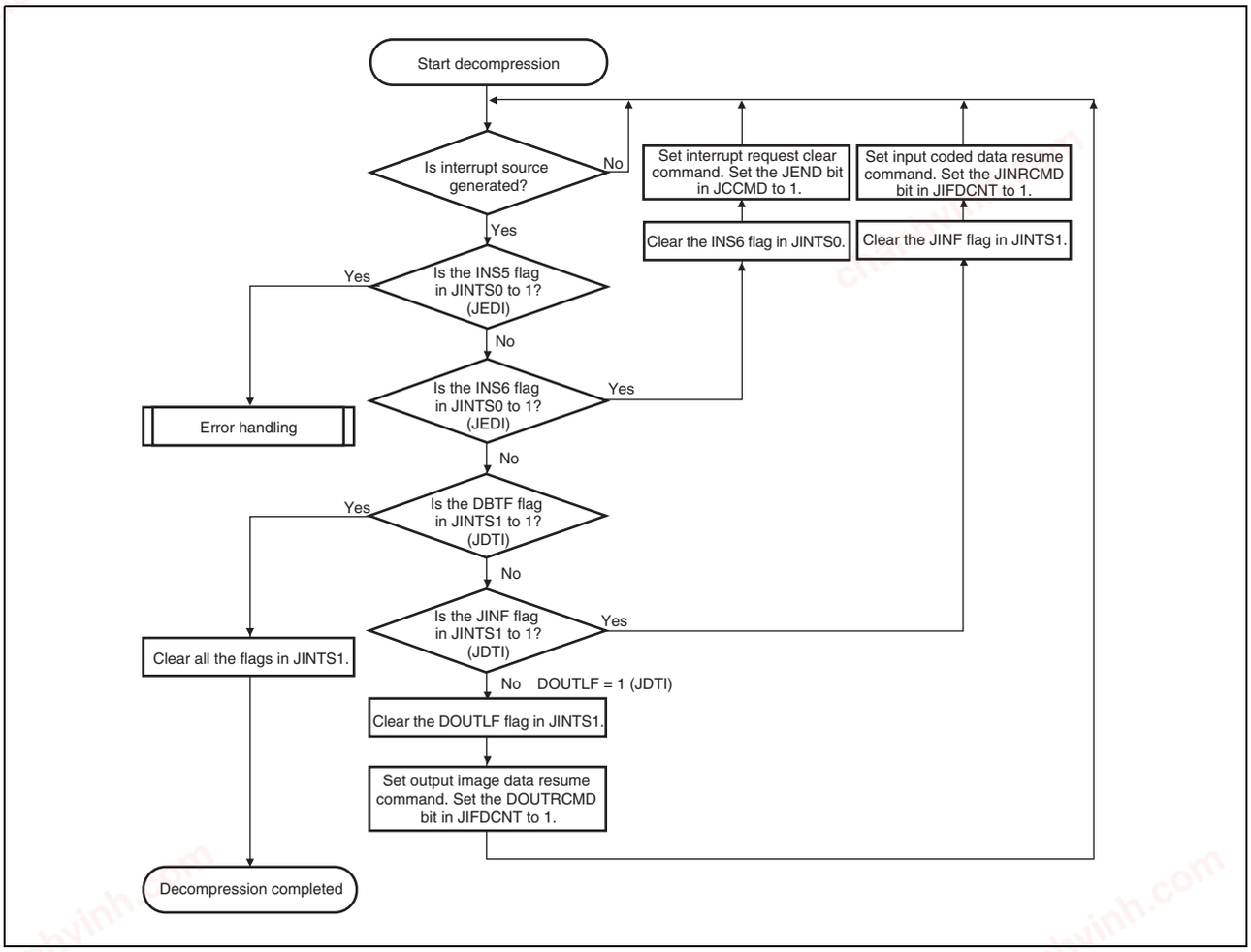


Figure 44.7 Decompression Process Flow

chanhvinh.com

(c) Error Handling

If the INS5 bit in JINTS0 is 1, it indicates that there is an error in the input JPEG coded data and that the decompression process by this module has been ended. Read the ERR bits in JCDERR to determine the cause of the error. The interrupt signal asserted due to the interrupt source indicated by the INS5 bit cannot be negated by clearing the interrupt status through 0-writing. To clear the interrupt request, set the interrupt request clear command (by setting the JEND bit in JCCMD to 1).
If decompression or compression is to proceed after error handling is completed, start by making the initial settings.

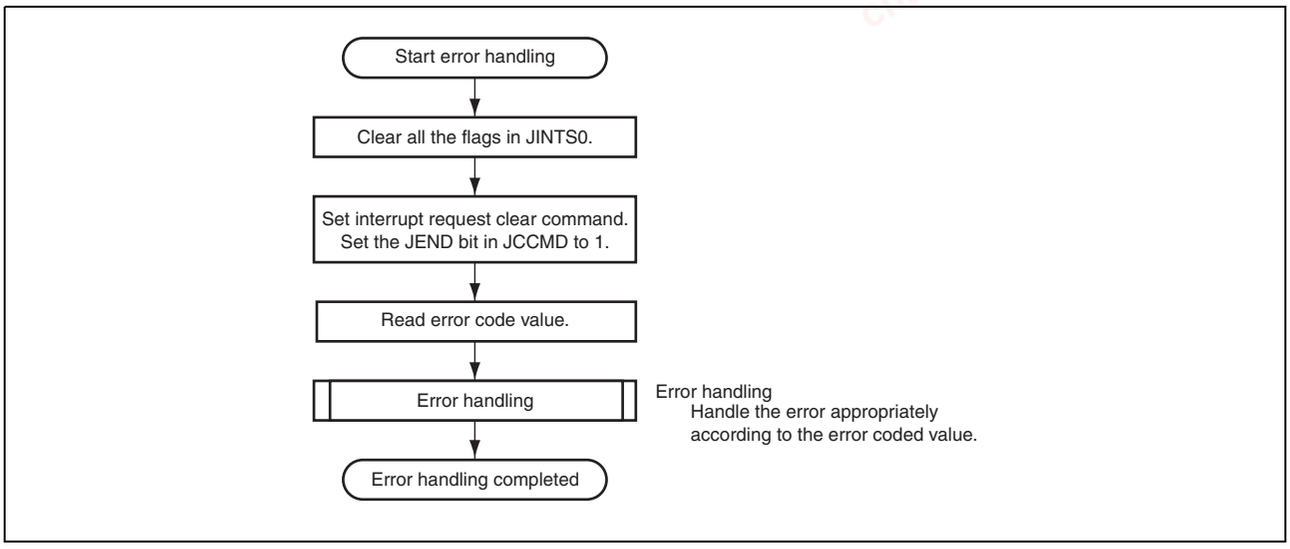


Figure 44.8 Error Handling Flow

(2) Input JPEG Coded Data

Markers to be processed in decompression are SOI, SOF0, SOS, DQT, DHT, DRI, RSTm, and EOI. Other markers except for the error markers shown below are ignored even if they are read.
The JINSWAP bits in JIFDCNT can be used to alter the arrangement for the input of coded data.

chanhvinh.com

(3) JPEG Decompression Errors

(a) Error Marker

If a marker error is found while analyzing compressed data for decompression, the code to identify the error type (shown in Table 44.3) is set to ERR bits in JCDERR. When an error is detected, the JCU generates an interrupt signal and terminates decoding. The stored code value will be set to B'1010 (default value) at the start of processing of the next frame or after a bus reset.

Table 44.3 Decompression Error Codes

Code	Description
B'0000	Normal
B'0001	SOI not detected: SOI not detected until EOI detected
B'0010	SOF1 to SOFF detected
B'0011	Unprovided pixel format detected
B'0100	SOF accuracy error: Other than 8 detected
B'0101	DQT accuracy error: Other than 0 detected
B'0110	Component error 1: The number of SOF0 header components detected is other than 1, 3, or 4
B'0111	Component error 2: The number of components differs between SOF0 header and SOS
B'1000	SOF0, DQT, and DHT not detected when SOS detected
B'1001	SOS not detected: SOS not detected until EOI detected
B'1010	EOI not detected (default)
B'1011	Restart interval data number error detected
B'1100	Image size error detected
B'1101	Last MCU data number error detected
B'1110	Block data number error detected

(b) Huffman Coded Segment Error

During the compressed data analysis in decompression operation, if there is an increase or decrease in the decoded data count due to an error resulting from bit reversal or missing data in the Huffman-coded segment, determine the error type, and set the error code in the ERR bit in JCDERR. Table 44.4 lists the segment error codes. The error code is set, interrupt signal is issued, and the process is ended only if the bits INT7 to INT5 in JINTE0 corresponding to the detected error is set to 1. The set code value will turn to the default value (B'1010) at the start of processing of the next frame or after a bus reset.

However, in this error detection, if an error in the Huffman-coded segment does not result in an alteration in the decoded data count, the error will go undetected.

[Example]

The number of data in a Huffman coded segment with pixel format setting YCbCr422, DRI = 2, X = 80 pixels, and Y = 8 pixels

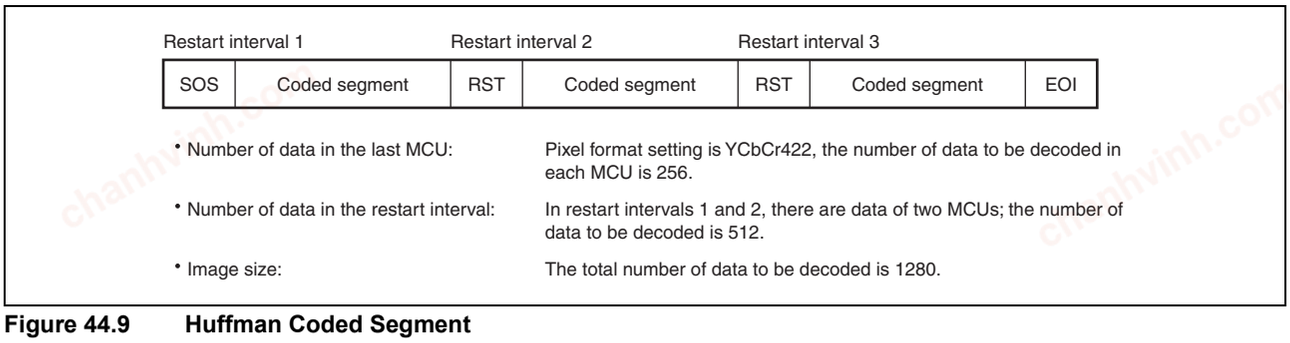


Figure 44.9 Huffman Coded Segment

chanhvinh.com

Table 44.4 Segment Error Codes

Code	Description
B'0000	Normal
B'1011	Restart interval data number error: The number of data in each interval is compared with the number of data specified by the DRI marker. If an interval has more or less data that is specified by the DRI marker, the decompression error code (1011) is set. The last interval which is shorter than the restart interval is not compared. If the DRI marker segment is not placed or the specified number is 00, an error is not detected even if the RSTm marker is placed. Also an m which indicates the order of RSTm marker modulo 8 (m = 0 to 7) is exempt from the error detection analysis. When the INT7 bit in JINTE0 is set to 0, this error is not detected.
B'1100	Image size error: The data number of an image which is calculated from the number of lines specified by the frame parameter and the number of samples per line is compared with the total number of data from SOS to EOI (in pixel units). If the numbers of data do not match, the decompression error code (1100) is set. When the INT6 bit in JINTE0 is set to 0, this error is not detected. The data number of an image is shown in MCU units. Thus the number of lines and the number of samples per line for calculation need to be shown in MCU units.
B'1101	Last MCU data number error: Whether the number of data in the MCUs at the EOI detection is shown in MCU units is checked and fractions are detected. If error (1100) occurs simultaneously, error (1100) has priority. When the INT5 bit in JINTE0 is set to 0, this error is not detected.
B'1110	Block data number error: Whether a block is an 8 × 8 array is checked; the check is performed for fractions. When bits INT7 to INT5 in JINTE0 are all set to 0, this error is not detected.

44.3.3 Output Pixel Format in Decompression

This module is capable of decompressing JPEG encoded data created in the YCbCr444, YCbCr422, YCbCr411 and YCbCr420 formats. The pixel format of the output image will be YCbCr422, ARGB8888, or RGB565. The flow of conversion of decompressed data to the given output pixel format is shown below.

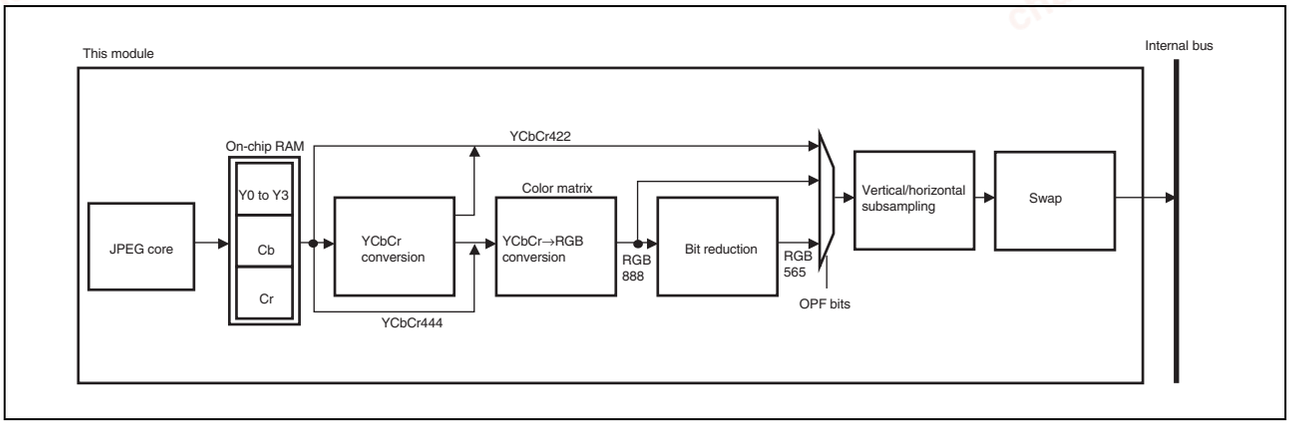


Figure 44.10 Block Diagram of Output Pixel Format Conversion in Decompression

(1) On-chip RAM

Data decoded by the JPEG core are stored in MCUs on RAM in this module.

(2) YCbCr Conversion

When data are to be output in the ARGB8888 or RGB565 format, data in the YCbCr422, YCbCr411 or YCbCr420 format are first converted to the YCbCr444 format.

When data are to be output in the YCbCr422 format, data in the YCbCr444, YCbCr411 or YCbCr420 format are converted to the YCbCr422 format.

Conversion is performed using simple interpolation.

chanhvinh.com

(3) YCbCr → RGB Conversion

Data in the YCbCr444 format are converted to the RGB888 format. The following formulae are used.

R = 1.000Y + 1.402Cr

G = 1.000Y - 0.344Cb - 0.714Cr

B = 1.000Y + 1.772Cb

(4) Bit Reduction

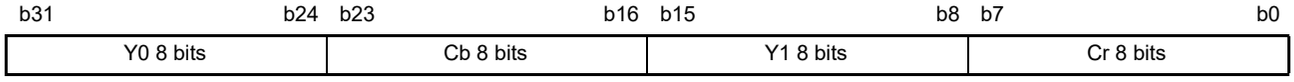
RGB888 data is reduced to RGB565 data. The lower three bits of red and blue, and lower two bits of green are removed.

(5) Output Pixel Format Selection

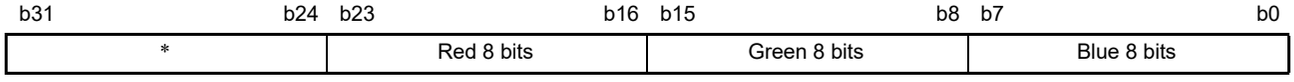
The pixel format to be output is selected by the OPF bit in JIFDCNT.

Allocation of data (while the DOUTSWAP bits in JIFDCNT = 000) in the pixel format is shown below.

- YCbCr422 (32 bits/pixel)

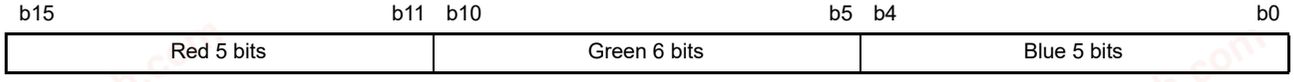


- ARGB8888 (32 bits/pixel)



Note: * This value is determined by the ALPHA[7:0] bits in JIFDADT.

- RGB565 (16 bits/pixel)



(6) Vertical/Horizontal Subsampling

The output data can be horizontally and vertically subsampled according to the VINTER and HINTER bit setting in JIFDCNT.

Figure 44.11 to Figure 44.13 show line subsampling modes.

For the output formats ARGB8888 and RGB565, one cell represents one pixel in the figures.

For the output format YCbCr422, one cell represents one set of Y0Cb0Y1Cr0 in the figures.

As subsampling is carried out by minimum coded units (MCU), the numbers of the horizontal and vertical block units will vary according to the decompressed pixel format.

Table 44.5 and Table 44.6 show the values of n and m in the figures.

Horizontal:

Table 44.5 Number of Horizontal Blocks

Compression Format	Output Format	n
YCbCr444	YCbCr422	1/2
YCbCr444	ARGB8888, RGB565	1
YCbCr422	YCbCr422	1
YCbCr422	ARGB8888, RGB565	2
YCbCr411	YCbCr422	2
YCbCr411	ARGB8888, RGB565	4
YCbCr420	YCbCr422	1
YCbCr420	ARGB8888, RGB565	2

chanhvinh.com

Vertical:

Table 44.6 Number of Vertical Blocks

Compression Format	Output Format	m
YCbCr444	YCbCr422	1
YCbCr444	ARGB8888, RGB565	1
YCbCr422	YCbCr422	1
YCbCr422	ARGB8888, RGB565	1
YCbCr411	YCbCr422	1
YCbCr411	ARGB8888, RGB565	1
YCbCr420	YCbCr422	2
YCbCr420	ARGB8888, RGB565	2

- Subsampling into 1/2
  Even lines are skipped by subsampling.

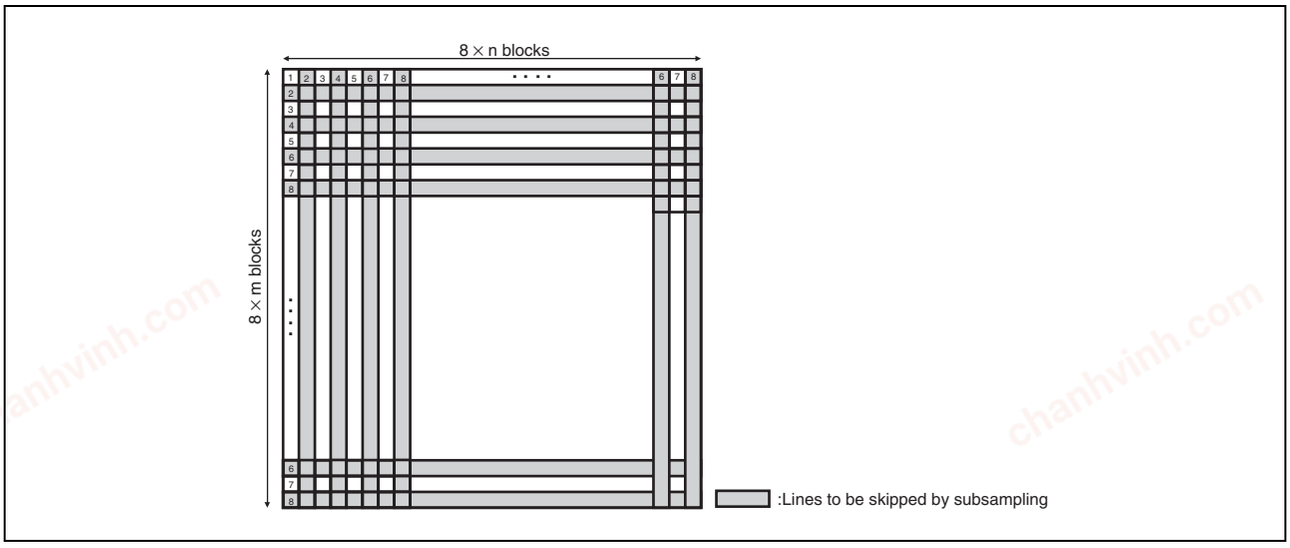


Figure 44.11 MCU when subsampling into 1/2 is selected

chanhvinh.com

- Subsampling into 1/4
  The second, third, and fourth lines are skipped by subsampling.

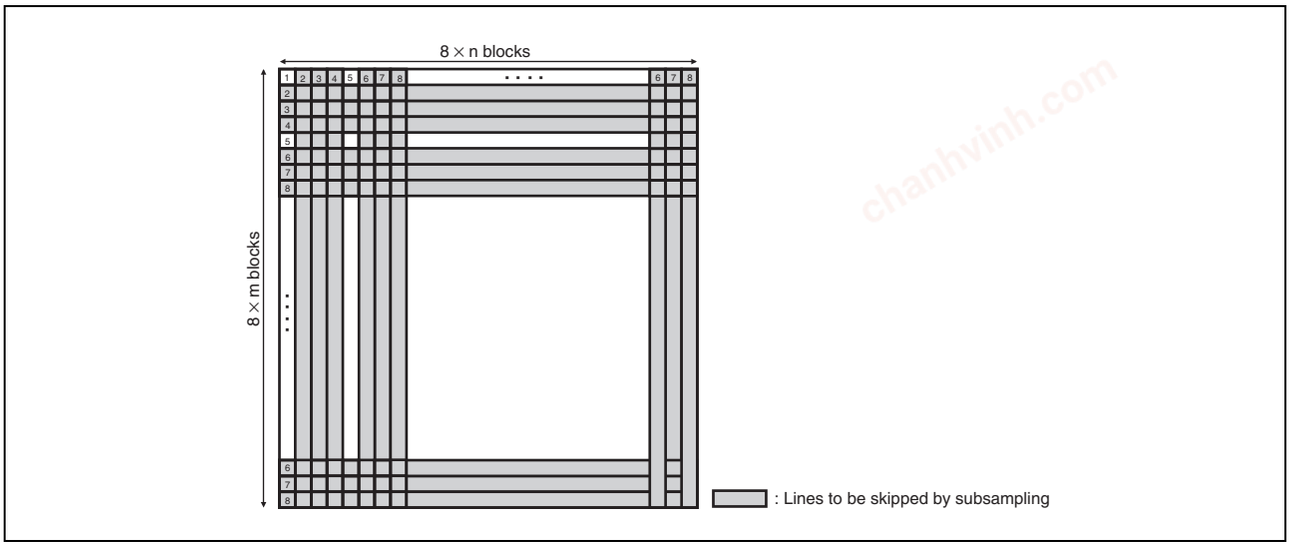


Figure 44.12 MCU when subsampling into 1/4 is selected

- Subsampling into 1/8
  The second, third, fourth, fifth, sixth, seventh, and eighth lines are skipped by subsampling.

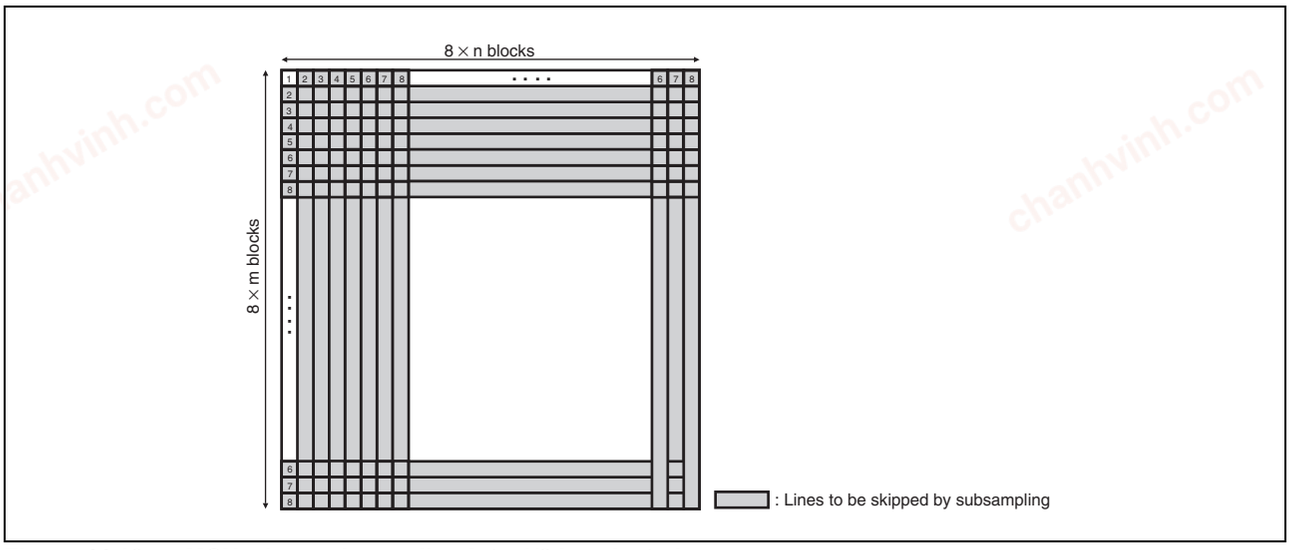


Figure 44.13 MCU when subsampling into 1/8 is selected

(7) Swap

Allocation of data can be changed by the DOUTSWAP bits in JIFECNT.

chanhvinh.com

44.3.4 Storing Image Data

Figure 44.14 shows the buffer area for storing the image data.

- Start address
  Compression: JIFESA
  Decompression: JIFDDA
- Horizontal size
  Compression, decompression: JCHSZU, JCHSZD
- Vertical size
  Compression, decompression: JCVSAU, JCVSZD
- Offset
  Compression: JIFESOFST
  Decompression: JIFDDOFST

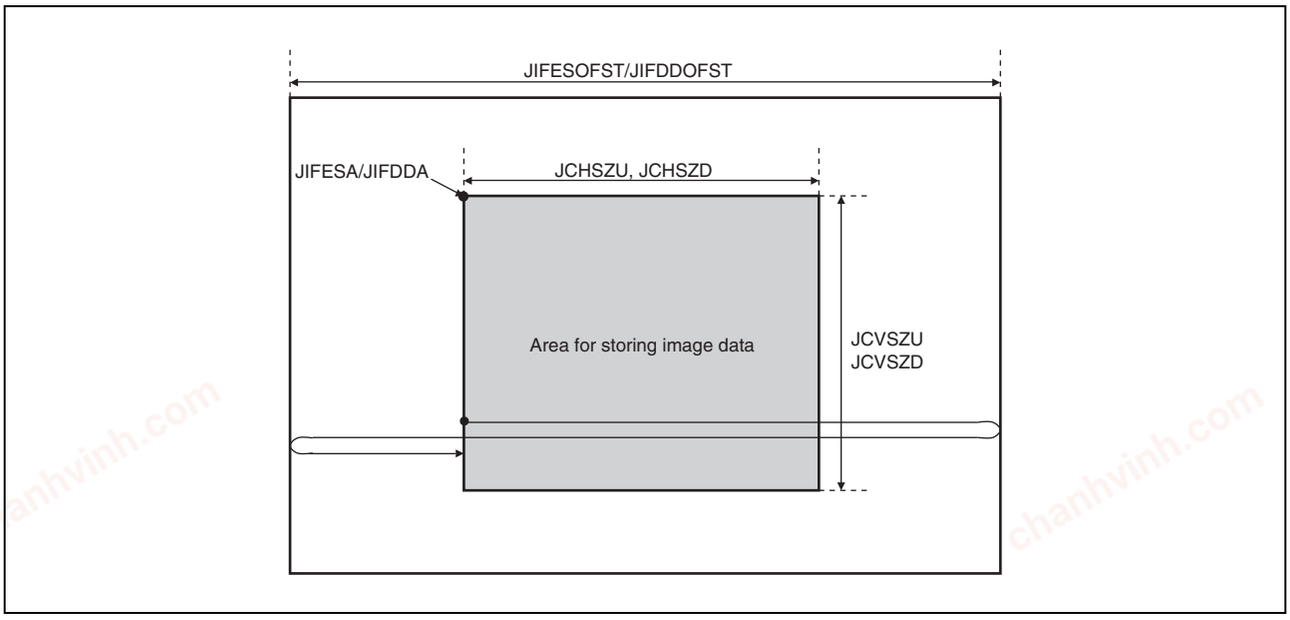


Figure 44.14 Image of Storing Image Data

chanhvinh.com

44.4 Interrupts

Two types of interrupt requests, namely compression/decompression process interrupt request (JEDI) and data transfer interrupt request (JDTI), are available in this module. The two types of interrupt requests are each related to multiple sources. The interrupt request cancellation methods differ depending on the source of the interrupt request.

44.4.1 Compression/Decompression Process Interrupt Request (JEDI)

The flags in JINTS0 indicate compression/decompression-related sources. The interrupt requests asserted by these interrupt sources cannot be negated by clearing the corresponding interrupt status bits to 0. Issue an interrupt request clear command (by setting the JEND bit in JCCMD to 1) to clear the interrupt request. When a flag in JINTS0 is set to 1, a compression/decompression process interrupt request is sent to the interrupt controller.

(1) Compression

- JPEG compression process end
  When the INS6 bit in JINTS0 is 1, the JPEG compression process has been successfully completed. After all of the coded data is transferred, the JCU completes compression.

(2) Decompression

- JPEG decompression process end
  When the INS6 bit in JINTS0 is 1, the JPEG decompression process has been successfully completed. After all of the image data is transferred, JCU completes decompression.
- JPEG decompression error occurrence
  When the INS5 bit in JINTS0 is 1, the input JPEG coded data has an error and the JCU has stopped the decompression process. Read the error code (ERR bits in JCDERR) and identify the error source. This interrupt occurs when any of the INT7 to INT5 bits in JINTE0 is 1.
- Request for reading the image size and pixel format
  When the INS3 bit in JINTS0 is 1, JPEG coded data has been input and information regarding the image size and pixel format can be read. Since the JPEG decompression process is suspended, resume the JPEG decompression process by setting the process stop clear command after accessing the necessary registers. This interrupt occurs when the INT3 bit in JINTE0 is 1.

44.4.2 Data Transfer Interrupt Request (JDTI)

The flags in JINTS1 are the interrupt sources for transferring the image data and coded data. The interrupt requests asserted by these interrupt sources can be negated by clearing the corresponding interrupt status bits to 0.

(1) Compression

- Interrupt request generated after the specified number of input image data lines has been read
  When the DINLF bit in JINTS1 is 1, the number of image data lines specified by JIFESLC has been transferred; transfer the rest of the image data to the external buffer and resume transferring the data from the external buffer. A data transfer interrupt request is sent when the DINLEN bit in JINTE1 is 1.
- Interrupt request generated after the specified amount of output coded data have been written to
  When the JOUTF bit in JINTS1 is 1, the amount of coded data specified by JIFEDDC has been transferred. Secure a space for the next coded data in the external buffer, and resume transfer process. The data transfer interrupt request is sent when the JOUTEN bit in JINTE1 is 1.
- Interrupt request generated after all processes are completed
  When the CBTF bit in JINTS1 is 1, the JCU has completed compression and transferred all of the coded data. The data transfer interrupt request is sent when the CBTEN bit in JINTE1 is set to 1.

chanhvinh.com

(2) Decompression

- Interrupt request generated after the specified number of output image data lines has been written to
  When the DOUTLF bit in the JINTS1 is 1, the number of image data lines specified by JIFDDLC has been transferred. Secure a space for the next coded data in the external buffer, and resume transfer process. A data transfer interrupt request is sent when the DOUTLEN bit in JINTE1 is 1.
- Interrupt request generated after the specified amount of input coded data has been read
  The JINF bit in JINTS1 becomes 1 when the amount of coded data specified by JIFDSDC has been transferred. Secure the next coded data in the external buffer, and resume transfer process. A data transfer interrupt is also sent at this time if the JINEN bit in JINTE1 is 1.
- Interrupt request generated after all processes are completed
  The DBTF bit in JINTS1 becomes 1 when the JCU has completed decompression and transferred all of the coded data. A data transfer interrupt request is also sent at this time if the DBTEN bit in JINTE1 is set to 1.

chanhvinh.com

44.5 Bus Reset Processing

Issuing the bus reset command (setting the BRST bit in JCCMD to 1) causes a bus reset.
When the JCU is in operation, the bus reset command should not be issued. Registers below are initialized by a bus reset.

- JPEG code data count upper register (JCDTCU)
- JPEG code data count middle register (JCDTCM)
- JPEG code data count lower register (JCDTCD)
- JPEG interrupt status register 0 (JINTS0)
- JPEG code decode error register (JCDERR)
- JPEG code reset register (JCRST)

chanhvinh.com

45. EthernetAVB

This module is only incorporated in the RZ/A1LU.

45.1 Overview

The EthernetAVB incorporates an Ethernet controller (E-MAC) that conforms to the definition of the MAC (Media Access Control) layer for Ethernet in the IEEE 802.3 standard.

When connected with a physical-layer LSI chip (PHY-LSI) that complies with the standard, the E-MAC is able to transmit and receive Ethernet (IEEE 802.3) frames. The E-MAC has a single MAC layer interface.

The EthernetAVB has a dedicated direct memory access controller (AVB-DMAC) for transferring transmitted Ethernet frames to and received Ethernet frames from respective storage areas in the on-chip RAM at high speed.

The AVB-DMAC is compliant with the following three standards formulated for IEEE 802.1BA: the IEEE 802.1AS timing and synchronization protocol, the IEEE 802.1Qav real-time transfer, and the IEEE 802.1Qat stream reservation protocol.

For the on-chip RAM, see section 40, On-Chip RAM.

45.1.1 Specifications (Functions)

Table 45.1 lists the specifications of the EthernetAVB module.

Table 45.1 Specifications (Functions)

Item	Description
Protocol	Flow control conforming with the IEEE 802.3x standard
Data transmission and reception	Transmission and reception of Ethernet (IEEE 802.3) frames
Transfer speed	Supports transfer at 10 and 100 Mbps
Mode	Full-duplex mode
Interface	Supports the IEEE 802.3 standard MII (Media Independent Interface)
Summary of the EthernetAVB function	An intelligent frame separation DMAC (AVB-DMAC) conforming with the following standards stipulated for IEEE 802.1BA: • IEEE 802.1AS (time synchronization protocol) • IEEE 802.1Qav (real-time transfer) • IEEE 1722 (AVTP presentation timestamp) IEEE 802.1Qat is supported by software. • Descriptor management system • Identification and sorting of frame data, and extraction and gathering of valid video data • Controllable interrupt frequency (reducing the load on the CPU)

chanhvinh.com

45.1.2 Block Diagram

Figure 45.1 is a block diagram of the EthernetAVB.

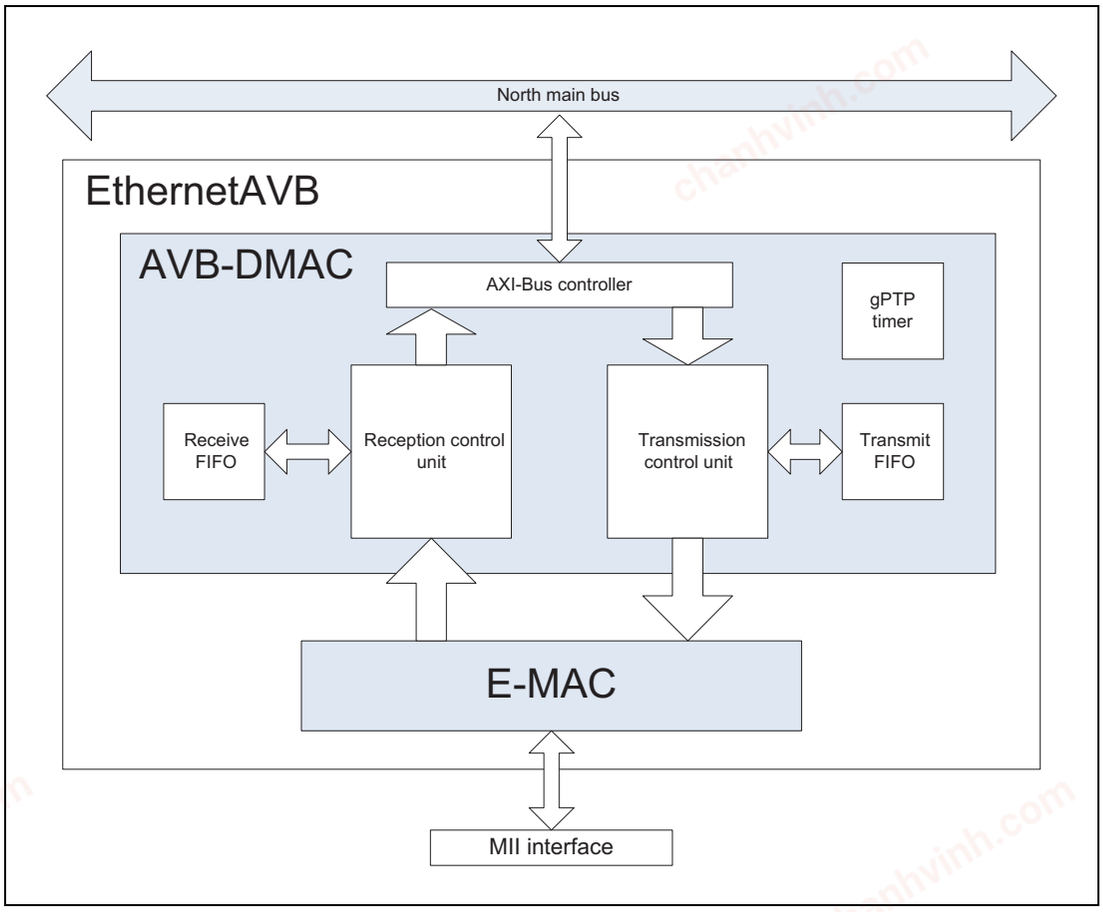


Figure 45.1 Block Diagram of EthernetAVB

chanhvinh.com

45.1.3 I/O Pins

Table 45.2 lists the pins for use by EthernetAVB.

Table 45.2 Pin Configuration

Pin Name	I/O	Function
ET_TXD[3:0]	O	Transmit data signal
ET_TXEN	O	Transmit data enable signal
ET_RXD[3:0]	I	Receive data signal
ET_RXDV	I	Receive data enable signal
ET_TXCLK	I	Transmit clock signal
ET_RXCLK	I	Receive clock signal
ET_TXER	O	Transmit error signal
ET_RXER	I	Reception error signal
ET_MDC	O	Management information transfer clock signal
ET_MDIO	I/O	Management information transmit/receive data
ET_CRS	I	Carrier detection signal
ET_COL	I	Collision detection signal
AVB_CAPTURE	I	Signal for capturing AVTP presentation times
AVB_GPTP_EXTERN	I	External clock signal for gPTP timer

chanhvinh.com

45.2 Register Descriptions

Table 45.3 and Table 45.4 list the EthernetAVB related registers and their configurations.

Table 45.3 Configuration of AVB-DMAC-related Registers (1/2)

Register Name	Symbol	Address	Access size
AVB-DMAC mode register	CCC	H'E821 5000	32
Descriptor base address table register	DBAT	H'E821 5004	32
Descriptor base address load request register	DLR	H'E821 5008	32
AVB-DMAC status register	CSR	H'E821 500C	32
Current descriptor address register q (q = 0 to 21)	CDARq	H'E821 5010+q*4	32
Error status register	ESR	H'E821 5088	32
Receive configuration register	RCR	H'E821 5090	32
Receive queue configuration register i (i = 0 to 4)	RQCi	H'E821 5094+i*4	32
Receive padding configuration register	RPC	H'E821 50B0	32
Unread frame counter stop level register	UFCS	H'E821 50C0	32
Unread frame counter register i (i = 0 to 4)	UFCVi	H'E821 50C4+i*4	32
Unread frame counter decrement register i (i = 0 to 4)	UFCDi	H'E821 50E0+i*4	32
Separation filter offset register	SFO	H'E821 50FC	32
Separation filter pattern register i (i = 0 to 31)	SFPi	H'E821 5100+i*4	32
Separation filter mask register i (i = 0, 1)	SFMi	H'E821 51C0+i*4	32
Transmit configuration register	TGC	H'E821 5300	32
Transmit configuration control register	TCCR	H'E821 5304	32
Transmit status register	TSR	H'E821 5308	32
Time stamp FIFO access register 0	TFA0	H'E821 5310	32
Time stamp FIFO access register 1	TFA1	H'E821 5314	32
Time stamp FIFO access register 2	TFA2	H'E821 5318	32
CBS increment value register c (c= 0, 1)	CIVRc	H'E821 5320+c*4	32
CBS decrement value register c (c = 0, 1)	CDVRc	H'E821 5328+c*4	32
CBS upper limit register c (c = 0, 1)	CULc	H'E821 5330+c*4	32
CBS lower limit register c (c = 0, 1)	CLLc	H'E821 5338+c*4	32
Descriptor interrupt control register	DIC	H'E821 5350	32
Descriptor interrupt status register	DIS	H'E821 5354	32
Error interrupt control register	EIC	H'E821 5358	32
Error interrupt status register	EIS	H'E821 535C	32
Receive interrupt control register 0	RIC0	H'E821 5360	32
Receive interrupt status register 0	RIS0	H'E821 5364	32
Receive interrupt control register 1	RIC1	H'E821 5368	32
Receive interrupt status register 1	RIS1	H'E821 536C	32
Receive interrupt control register 2	RIC2	H'E821 5370	32
Receive interrupt status register 2	RIS2	H'E821 5374	32
Transmit interrupt control register	TIC	H'E821 5378	32
Transmit interrupt status register	TIS	H'E821 537C	32
Interrupt summary status register	ISS	H'E821 5380	32
gPTP configuration control register	GCCR	H'E821 5390	32
gPTP maximum transit time register	GMTT	H'E821 5394	32

chanhvinh.com

Table 45.3 Configuration of AVB-DMAC-related Registers (2/2)

Register Name	Symbol	Address	Access size
gPTP presentation time comparison register	GPTC	H'E821 5398	32
gPTP timer increment register	GTI	H'E821 539C	32
gPTP timer offset configuration register i (i = 0 to 2)	GTOi	H'E821 53A0+i*4	32
gPTP interrupt control register	GIC	H'E821 53AC	32
gPTP interrupt status register	GIS	H'E821 53B0	32
gPTP presentation time capture register	GCPT	H'E821 53B4	32
gPTP timer capture register i (i = 0 to 2)	GCTi	H'E821 53B8+i*4	32
gPTP capture event control register	GCEC	H'E821 53D8	32

Table 45.4 Configuration of E-MAC-related Registers

Register Name	Symbol	Address	Access size
E-MAC mode register	ECMR	H'E821 5500	32
Receive frame length register	RFLR	H'E821 5508	32
E-MAC status register	ECSR	H'E821 5510	32
E-MAC interrupt permission register	ECSIPR	H'E821 5518	32
PHY interface register	PIR	H'E821 5520	32
Automatic PAUSE frame register	APR	H'E821 5554	32
Manual PAUSE frame register	MPR	H'E821 5558	32
PAUSE frame transmit counter	PFTCR	H'E821 555C	32
PAUSE frame receive counter	PFRCR	H'E821 5560	32
Automatic PAUSE frame retransmission count register	TPAUSER	H'E821 5564	32
MAC address high register	MAHR	H'E821 55C0	32
MAC address low register	MALR	H'E821 55C8	32
CRC error frame receive counter register	CEFCR	H'E821 5740	32
Frame receive error counter register	FRECR	H'E821 5748	32
Too-short frame receive counter register	TSFRCR	H'E821 5750	32
Too-long frame receive counter register	TLFRCR	H'E821 5758	32
Residual-bit frame receive counter register	RFCR	H'E821 5760	32
Multicast address frame receive counter register	MAFCR	H'E821 5778	32

chanhvinh.com

45.2.1 AVB-DMAC Mode Register (CCC)

The CCC register specifies the operating mode of the AVB-DMAC.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	FCE	LBME	—	—	—	BOC	—	—	CSEL[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R	R	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DTSR	—	—	—	—	—	—	OPC[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R/W	R/W

Table 45.5 CCC register contents

Bit Position	Bit Name	Function
b31 to b26	—	Reserved These bits are read as 0. The write value should be 0.
b25	FCE	Flow Control Enable 0: Normal operation 1: Flow control is enabled.
b24	LBME	Loopback Mode Enable 0: Normal operation 1: Loopback mode is enabled.
b23 to b21	—	Reserved These bits are read as 0. The write value should be 0.
b20	BOC	First Byte Specification 0: The first byte is the 8 lower-order bits (on-chip RAM[7:0]) 1: The first byte is the 8 higher-order bits (on-chip RAM[31:24])
b19, b18	—	Reserved These bits are read as 0. The write value should be 0.
b17, b16	CSEL[1:0]	gPTP Clock Select B'00: gPTP is not in use. B'01: Internal bus clock (Bϕ) B'10: Ethernet transmission clock (ET_TXCLK) B'11: External clock (AVB_GPTP_EXTERN)
b15 to b9	—	Reserved These bits are read as 0. The write value should be 0.
b8	DTSR	Data Transfer Suspend Request 0: Normal operation 1: Requests suspension.
b7 to b2	—	Reserved These bits are read as 0. The write value should be 0.
b1, b0	OPC[1:0]	Operating Mode Configuration B'00: Reset mode B'01: Configuration mode B'10: Operation mode B'11: Standby mode

FCE (Flow Control Enable) Bit

This bit enables flow control by the E-MAC.

Writing to this bit is only possible when the current operating mode is configuration mode.

chanhvinh.com

LBME (Loopback Mode Enable) Bit

This bit enables loopback mode.

In loopback mode, the transmission lines are internally connected to the reception lines. When loopback mode is to be used, the Ethernet transmission clock must be supplied to the MII interface. A received clock signal is not required. Writing to this bit is only possible when the current operating mode is configuration mode.

CAUTION

Data for transmission are still output normally. To eliminate effects on external modules, pin control should be applied to block the output of data. For pin control, see section 41., Ports.

BOC (First Byte Specification) Bit

Specifies the allocation of the first byte from a received Ethernet frame to the on-chip RAM.

This configuration setting does not affect the format and filter parameters of the descriptor in the on-chip RAM.

Writing to this bit is only possible when the current operating mode is configuration mode.

Figure 45.2 to Figure 45.4 show how data from frames received via the Ethernet connection are stored in the on-chip RAM.

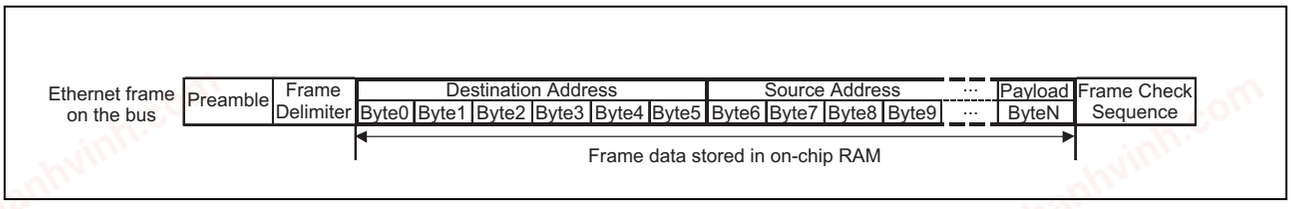


Figure 45.2 Data for Reception in an Ethernet Frame

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPTR+0	Byte3								Byte2								Byte1								Byte0							
DPTR+4	Byte7								Byte6								Byte5								Byte4							
DPTR+8	Byte11								Byte10								Byte9								Byte8							

Figure 45.3 When CCC.BOC = 0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPTR+0	Byte0								Byte1								Byte2								Byte3							
DPTR+4	Byte4								Byte5								Byte6								Byte7							
DPTR+8	Byte8								Byte9								Byte10								Byte11							

Figure 45.4 When CCC.BOC = 1

chanhvinh.com

CSEL[1:0] (gPTP Clock Select) Bits

These bits select the clock source for the gPTP timer.

Writing to these bits is only possible when the current operating mode is configuration mode.

DTSR (Data Transfer Suspend Request) Bit

This bit can suspend access to the on-chip RAM.

The access is suspended on completion of the transfer of the frame currently being transferred.

This function disables access to the on-chip RAM without affecting normal operation of the AVB-DMAC. Use this bit when exclusive control over the contents of the on-chip RAM is necessary, for example, in checking its integrity.

Note that the transmission and reception queues are not processed while access is suspended.

Change neither the AVB-DMAC settings nor the mode from when this bit is set to 1 to complete suspension until when the data transfer suspend status bit (CSR.DTS) is set to 1.

OPC[1:0] (Operating Mode Configuration) Bits

These bits specify the operating mode.

For the operating modes, see section 45.3.1.1, Operating Modes.

Writing to this bit is possible in any of the operating modes, but should not be done after the standby request has been issued to the EthernetAVB. For issuance of the standby request to the EthernetAVB, see section 42., Power-Down Modes.

chanhvinh.com

45.2.2 Descriptor Base Address Table Register (DBAT)

The DBAT register is used to set the base address of the descriptor table.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TA[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TA[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.6 DBAT register contents

Bit Position	Bit Name	Function
b31 to b0	TA[31:0]	Table Address Base address of the descriptor table in the on-chip RAM

CAUTION

The setting of this bit must be a multiple of four (i.e. b0 and b1 must be set to 0).

TAL3[1:0] (Table Address) Bits

These bits specify the base address of the descriptor table in the on-chip RAM.

For the structure of this table, see section 45.3.3, Descriptors.

Writing to this bit is only possible when the current operating mode is configuration mode.

chanhvinh.com

45.2.3 Descriptor Base Address Load Request Register (DLR)

The DLR register is used to issue a request to load the values from the current descriptor address register q (CDARq) for each queue to the descriptor base address table register (DBAT).

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	LBA21	LBA20	LBA19	LBA18	LBA17	LBA16
Initial value	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LBA15	LBA14	LBA13	LBA12	LBA11	LBA10	LBA9	LBA8	LBA7	LBA6	LBA5	LBA4	LBA3	LBA2	LBA1	LBA0
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.7 DLR register contents (1/2)

Bit Position	Bit Name	Function
b31 to 22	—	Reserved These bits are read as 0. The write value should be 0.
b21	LBA21	Base Address Load Request (Rx17: Stream 15) 0(R): No load request is issued. 1(W): A request for loading the corresponding base address is issued. 1(R): The given base address is being loaded.
b20	LBA20	Base Address Load Request (Rx16: Stream 14) 0(R): No load request is issued. 1(W): A request for loading the corresponding base address is issued. 1(R): The given base address is being loaded.
b19	LBA19	Base Address Load Request (Rx15: Stream 13) 0(R): No load request is issued. 1(W): A request for loading the corresponding base address is issued. 1(R): The given base address is being loaded.
b18	LBA18	Base Address Load Request (Rx14: Stream 12) 0(R): No load request is issued. 1(W): A request for loading the corresponding base address is issued. 1(R): The given base address is being loaded.
b17	LBA17	Base Address Load Request (Rx13: Stream 11) 0(R): No load request is issued. 1(W): A request for loading the corresponding base address is issued. 1(R): The given base address is being loaded.
b16	LBA16	Base Address Load Request (Rx12: Stream 10) 0(R): No load request is issued. 1(W): A request for loading the corresponding base address is issued. 1(R): The given base address is being loaded.
b15	LBA15	Base Address Load Request (Rx11: Stream 9) 0(R): No load request is issued. 1(W): A request for loading the corresponding base address is issued. 1(R): The given base address is being loaded.
b14	LBA14	Base Address Load Request (Rx10: Stream 8) 0(R): No load request is issued. 1(W): A request for loading the corresponding base address is issued. 1(R): The given base address is being loaded.
b13	LBA13	Base Address Load Request (Rx9: Stream 7) 0(R): No load request is issued. 1(W): A request for loading the corresponding base address is issued. 1(R): The given base address is being loaded.

chanhvinh.com

Table 45.7 DLR register contents (2/2)

Bit Position	Bit Name	Function
b12	LBA12	Base Address Load Request (Rx8: Stream 6) 0(R): No load request is issued. 1(W): A request for loading the corresponding base address is issued. 1(R): The given base address is being loaded.
b11	LBA11	Base Address Load Request (Rx7: Stream 5) 0(R): No load request is issued. 1(W): A request for loading the corresponding base address is issued. 1(R): The given base address is being loaded.
b10	LBA10	Base Address Load Request (Rx6: Stream 4) 0(R): No load request is issued. 1(W): A request for loading the corresponding base address is issued. 1(R): The given base address is being loaded.
b9	LBA9	Base Address Load Request (Rx5: Stream 3) 0(R): No load request is issued. 1(W): A request for loading the corresponding base address is issued. 1(R): The given base address is being loaded.
b8	LBA8	Base Address Load Request (Rx4: Stream 2) 0(R): No load request is issued. 1(W): A request for loading the corresponding base address is issued. 1(R): The given base address is being loaded.
b7	LBA7	Base Address Load Request (Rx3: Stream 1) 0(R): No load request is issued. 1(W): A request for loading the corresponding base address is issued. 1(R): The given base address is being loaded.
b6	LBA6	Base Address Load Request (Rx2: Stream 0) 0(R): No load request is issued. 1(W): A request for loading the corresponding base address is issued. 1(R): The given base address is being loaded.
b5	LBA5	Base Address Load Request (Rx1: Network Control) 0(R): No load request is issued. 1(W): A request for loading the corresponding base address is issued. 1(R): The given base address is being loaded.
b4	LBA4	Base Address Load Request (Rx0: Best Effort) 0(R): No load request is issued. 1(W): A request for loading the corresponding base address is issued. 1(R): The given base address is being loaded.
b3	LBA3	Base Address Load Request (Tx3: Stream Class A) 0(R): No load request is issued. 1(W): A request for loading the corresponding base address is issued. 1(R): The given base address is being loaded.
b2	LBA2	Base Address Load Request (Tx2: Stream Class B) 0(R): No load request is issued. 1(W): A request for loading the corresponding base address is issued. 1(R): The given base address is being loaded.
b1	LBA1	Base Address Load Request (Tx1: Network Control) 0(R): No load request is issued. 1(W): A request for loading the corresponding base address is issued. 1(R): The given base address is being loaded.
b0	LBA0	Base Address Load Request (Tx0: Best Effort) 0(R): No load request is issued. 1(W): A request for loading the corresponding base address is issued. 1(R): The given base address is being loaded.

chanhvinh.com

LBAq (Base Address Load Request) Bits

Each bit is used to issue requests to load base addresses and to indicate that a base address is currently being loaded.

Setting a bit to 1 issues a request for loading the descriptor base address for the queue q (q = 0 to 21), to change the current descriptor address bit (CDARq.CDA) to DBAT.TA + 8*q.

If transfer is currently in progress, loading is executed on completion of transfer for the current frame.

Completion of loading leads to automatic setting of the corresponding bit to 0.

For the transmission queues, base address load requests are executed even while fetching is in progress (the transmit start request bit in the transmit configuration control register (TCCR.TSRQt) is 1). Therefore, be sure to check that fetching is not in progress before issuing a request.

Writing to a bit of this register is only possible when the current operating mode is configuration mode.

Only 1 can be written to this bit.

chanhvinh.com

45.2.4 AVB-DMAC Status Register (CSR)

The CSR register is used to indicate the operating mode in which the AVB-DMAC is running and the individual communications states.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	RPO	TPO3	TPO2	TPO1	TPO0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DTS	—	—	—	—	OPS[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.8 CSR register contents

Bit Position	Bit Name	Function
b31 to b21	—	Reserved These bits are read as 0. The write value should be 0.
b20	RPO	Receive Process Status 0: Reception completed. 1: Reception is in progress.
b19	TPO3	Transmit Process Status 3 (Stream Class A) 0: Transmission completed. 1: Transmission is in progress.
b18	TPO2	Transmit Process Status 2 (Stream Class B) 0: Transmission completed. 1: Transmission is in progress.
b17	TPO1	Transmit Process Status 1 (Network Control) 0: Transmission completed. 1: Transmission is in progress.
b16	TPO0	Transmit Process Status 0 (Best Effort) 0: Transmission completed. 1: Transmission is in progress.
b16 to b9	—	Reserved These bits are read as 0. The write value should be 0.
b8	DTS	Data Transfer Suspend Status 0: Normal operation 1: Transmission is suspended.
b7 to b4	—	Reserved These bits are read as 0. The write value should be 0.
b3 to b0	OPS[3:0]	Operating Mode Status B'0001: Reset mode B'0010: Configuration mode B'0100: Operation mode B'1000: Standby mode Other settings are reserved.

chanhvinh.com

RPO (Receive Process Status) Bit

This bit indicates whether a reception queue contains an unread received frame.

This bit being set to 1 indicates that a received frame is yet to be stored in the on-chip RAM.

- [Clearing conditions]
  The current operating mode is not operation mode.
  Received frames in the reception FIFO all being stored in the on-chip RAM.
- [Setting condition]
  A received frame being stored in the reception FIFO (but not yet in the on-chip RAM)

TPO3 (Transmit Process Status 3) Bit

This bit indicates whether a class A stream is being transmitted.

This bit being set to 1 indicates that the AVB-DMAC is fetching data for transmission from the on-chip RAM, or the E-MAC is transmitting data.

- [Clearing conditions]
  The current operating mode is not operation mode.
  Completion of transfer of all frames for transmission from the transmission FIFO (the transmit start request bit in the transmit configuration control register (TCCR.TSRQ3) is 0)
- [Setting condition]
  Transmission being started (by writing 1 to the transmit start request bit in the transmit configuration control register (TCCR.TSRQ3))

TPO2 (Transmit Process Status 2) Bit

This bit indicates whether a class B stream is being transmitted.

This bit being set to 1 indicates that the AVB-DMAC is fetching data for transmission from the on-chip RAM, or the E-MAC is transmitting data.

- [Clearing conditions]
  The current operating mode is not operation mode.
  Completion of transfer of all frames for transmission from the transmission FIFO (the transmit start request bit in the transmit configuration control register (TCCR.TSRQ2) is 0)
- [Setting condition]
  Transmission being started (by writing 1 to the transmit start request bit in the transmit configuration control register (TCCR.TSRQ2))

TPO1 (Transmit Process Status 1) Bit

This bit indicates whether a network control is being transmitted.

This bit being set to 1 indicates that the AVB-DMAC is fetching data for transmission from the on-chip RAM, or the E-MAC is transmitting data.

- [Clearing conditions]
  The current operating mode is not operation mode.
  Completion of transfer of all frames for transmission from the transmission FIFO (the transmit start request bit in the transmit configuration control register (TCCR.TSRQ1) is 0)
- [Setting condition]
  Transmission being started (by writing 1 to the transmit start request bit in the transmit configuration control register (TCCR.TSRQ1))

chanhvinh.com

TPO0 (Transmit Process Status 0) Bit

This bit indicates whether a best effort is being transmitted.

This bit being set to 1 indicates that the AVB-DMAC is fetching data for transmission from the on-chip RAM, or the E-MAC is transmitting data.

- [Clearing conditions]
  The current operating mode is not operation mode.
  Completion of transfer of all frames for transmission from the transmission FIFO (the transmit start request bit in the transmit configuration control register (TCCR.TSRQ0) is 0)
- [Setting condition]
  Transmission being started (by writing 1 to the transmit start request bit in the transmit configuration control register (TCCR.TSRQ0))

DTS (Data Transfer Suspend Status) Bit

This bit indicates whether access to the on-chip RAM is enabled.

- [Clearing condition]
  The current operating mode is not operation mode.
  The data transfer suspend request bit in the AVB-DMAC mode register (CCC.DTSR) being 0.
- [Setting condition]
  Access to the on-chip RAM not proceeding while the data transfer suspend request bit (CCC.DTSR) in the AVB-DMAC mode register (CCC) is 1 (if the on-chip RAM is being accessed, this bit is set to 1 on completion of access).

OPS[3:0] (Operating Mode Status) Bits

These bits indicate the current operating mode.

For the operating modes, see section 45.3.1.1, Operating Modes.

chanhvinh.com

45.2.5 Current Descriptor Address Register q (CDARq) (q = 0 to 21)

The CDARq register indicates the current descriptor address.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CDA[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CDA[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.9 CDARq (q = 0 to 21) register contents

Bit Position	Bit Name	Function
b31 to b0	CDA[31:0]	Current Descriptor Address The address of the current descriptors for the transmission queues

CDA[31:0] (Current Descriptor Address) Bits

CDAR0 to CDAR3 indicate the addresses of the current descriptors for the corresponding transmission queues while CDAR4 to CDAR21 indicate the addresses of the current descriptors for the corresponding reception queues.

If the operating mode is changed to operation mode, the contents of the register for the queue to be used are set in the descriptor base address table register (DBAT).

Also, when the descriptor base address load request register (DLR) issues a load request, the contents of the corresponding register are set in the descriptor base address table register (DBAT).

Conditions for updating:

These bits are set to 0 when the operating mode is not operation mode.

This register is updated in response to processing of the descriptor for a queue.

chanhvinh.com

45.2.6 Error Status Register (ESR)

The ESR register indicates the error status in the AVB-DMAC.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	EIL	ET[3:0]				—	—	—	EQN[4:0]				
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.10 ESR register contents

Bit Position	Bit Name	Function
b31 to b13	—	Reserved These bits are read as 0. The write value should be 0.
b12	EIL	Error Information Lost 0: No error information lost. 1: Error information lost.
b11 to b8	ET[3:0]	Error Type B'0000: Descriptor reading error from the on-chip RAM B'0001: Descriptor writing error to the on-chip RAM B'0010: Illegal descriptor reading error B'0100: Data/time stamp reading error from the on-chip RAM B'0101: Data/time stamp writing error to the on-chip RAM B'1001: Transmission frame size error B'1010: Transmission FIFO overflow error
b7 to b5	—	Reserved These bits are read as 0. The write value should be 0.
b4 to b0	EQN[4:0]	Error Queue Number Error queue number.

EIL (Error Information Lost) Bit

This bit indicates that error information has been lost because a new error was detected when the previous error remained unprocessed.

Conditions for updating:

This bit is set to 0 when the operating mode is not operation mode.
This bit is also set to 0 when 0 is written to the queue error interrupt status bit in the error interrupt status register (EIS.QEF).

This bit is set to 1 when the condition for setting the queue error interrupt status bit in the error interrupt status register (EIS.QEF) to 1 is satisfied when the queue error interrupt status bit in the error interrupt status register (EIS.QEF) is 1.

ET[3:0] (Error Type) Bits

These bits indicate the detail of the error detected.
When the detected error is related to descriptor reading (ESR.ET = B'0000 or B'0010), the current descriptor address bits in the current descriptor address register q (CDARq.CDA) are not updated thus the same descriptor is processed again. To handle this, correct the descriptor.

chanhvinh.com

When the detected error is related to descriptor writing (ESR.ET = B'0001), the current descriptor address bits in the current descriptor address register q (CDARq.CDA) and the transmit start request bit in the transmit configuration control register (TCCR.TSRQt) are updated as in the normal case thus the descriptor chain processing is not affected.
The other errors are temporary and might be eliminated by transfer continuation. For details, refer to section 45.3.2.3, Checking Integrity.

This bit should be referred to only when the queue error interrupt status bit in the error interrupt status register (EIS.QEF) is 1.

Conditions for updating:

These bits are updated when the condition for setting the queue error interrupt status bit in the error interrupt status register (EIS.QEF) to 1 is satisfied when the queue error interrupt status bit in the error interrupt status register (EIS.QEF) is 0.

EQN (Error Queue Number) Bits

These bits indicate the queue number in which the error has been detected.
When ESR.EQN is 0 to 3, it means that the related queue is a transmission queue and the EQN value is the transmission queue number. When ESR.EQN is 4 or larger, it means that the related queue is a reception queue and the "ESR.EQN – 4" is the reception queue number r.
Refer to these bits only when the queue error interrupt status bit in the error interrupt status register (EIS.QEF) is 1.

Conditions for updating:

These bits are updated when the condition for setting the queue error interrupt status bit in the error interrupt status register (EIS.QEF) to 1 is satisfied when the EIS.QEF bit is 0.

chanhvinh.com

45.2.7 Receive Configuration Register (RCR)

The RCR register is used to make settings related to reception for the AVB-DMAC.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	RFCL[12:0]												
Initial value	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	ETS2	ETS0	ESF[1:0]		ENCF	EFFS
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.11 RCR register contents

Bit Position	Bit Name	Function
b31 to b29	—	Reserved These bits are read as 0. The write value should be 0.
b28 to b16	RFCL[12:0]	Receive FIFO Caution Level
b15 to b6	—	Reserved These bits are read as 0. The write value should be 0.
b5	ETS2	Time Stamp Enable (Stream) 0: Time stamping is disabled. 1: Time stamping is enabled.
b4	ETS0	Time Stamp Enable (Best Effort) 0: Time stamping is disabled. 1: Time stamping is enabled.
b3, b2	ESF[1:0]	Stream Filtering Select Settings for reception queues 2 to 17 B'00: Filtering is disabled. Frames are processed in queue 0 (best effort). B'01: The filter for separating AVB stream frames from non-AVB stream frames is enabled; non-matching frames from the stream are processed in queue 0 (best effort). B'10: The filter for separating AVB stream frames from non-AVB stream frames is enabled; non-matching frames are discarded. B'11: The filter for separating AVB stream frames from non-AVB stream frames is enabled; non-matching frames from a stream are processed in queue 0 (best effort).
b1	ENCF	Network Control Filtering Enable Setting for reception queue 1 (network control) 0: Network control is disabled. 1: Network control is enabled.
b0	EFFS	Error Frame Enable 0: Error frames are disabled. 1: Error frames are enabled.

chanhvinh.com

RFCL[12:0] (Receive FIFO Caution Level) Bits

These bits set the caution level for the reception FIFO and are used to maintain the priority order of the storage of received data and the fetching of data for transmission.

If the reception FIFO contains less data than this level, processing of both transmission and reception queues becomes pending.

If the reception FIFO contains more data than this level, only data in the reception queue are transferred, and processing of the transmission queue becomes pending.

Writing to this bit is only possible when the current operating mode is configuration mode.

CAUTION

- In the case of this LSI chip, set these bits to H'1800.

ETS2 (Time Stamp Enable (Stream)) Bit

Enables the inclusion of time-stamp information in reception queues 2 to 17.

The descriptor structure differs depending on whether time-stamp information is included or not. The reception queue in which time-stamp information is enabled uses the extended descriptor.

Writing to this bit is only possible when the current operating mode is configuration mode.

ETS0 (Time Stamp Enable (Best Effort)) Bit

Enables the inclusion of time-stamp information in reception queue 0.

The descriptor structure differs depending on whether time-stamp information is included or not. The reception queue 0 in which time-stamp information is enabled uses the extended descriptor.

Writing to this bit is only possible when the current operating mode is configuration mode.

ESF[1:0] (Stream Filtering Select) Bits

These bits select separation filtering for reception queues 2 to 17.

The queue-dependent separation filter can be used in combination with the identification of AVB stream frames.

When the value is B'00, filtering is disabled and frames from streams are processed in reception queue 0 (best effort).

When the value is B'01, the separation filter is enabled for both AVB stream frames and non-AVB stream frames; frames from non-matching streams are processed in reception queue 0 (best effort).

When the value is B'10, the separation filter is enabled for AVB stream frames; frames from non-matching streams are discarded.

When the value is B'11, the separation filter is enabled for AVB stream frames; frames from non-matching streams are processed in reception queue 0 (best effort).

For separation filtering, see 45.3.4.1 (1) Separation Filtering.

Writing to this bit is only possible when the current operating mode is configuration mode.

ENCF (Enable Network Control Filtering) Bit

Enables the AVB network control frame for reception queue 1.

chanhvinh.com

When reception queue 1 is disabled, a received frame is stored in reception queue 0 (best effort).

Writing to this bit is only possible when the current operating mode is configuration mode.

EFFS (Enable Error Frame) Bit

Enables or disables the reception of frames that have been classified as error frames by the E-MAC.

Received error frames are stored in reception queue 0 (best effort).

An indicator of error detection by the E-MAC during reception is stored in the descriptor (DESCR.MS).

Writing to this bit is only possible when the current operating mode is configuration mode.

chanhvinh.com

45.2.8 Receive Queue Configuration Register i (RQCi) (i = 0 to 4)

The RQC0 register is used to set up reception queues 0 to 3.

The RQC1 register is used to set up reception queues 4 to 7.

The RQC2 register is used to set up reception queues 8 to 11.

The RQC3 register is used to set up reception queues 12 to 15.

The RQC4 register is used to set up reception queues 16 to 17.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	UFCC3[1:0]		—	—	RSM3[1:0]		—	—	UFCC2[1:0]		—	—	RSM2[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	UFCC1[1:0]		—	—	RSM1[1:0]		—	—	UFCC0[1:0]		—	—	RSM0[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Table 45.12 RQCi register contents (1/2)

Bit Position	Bit Name	Function
b31, b30	—	Reserved These bits are read as 0. The write value should be 0.
b29, b28	UFCC3[1:0]	Unread Frame Counter Configuration (Receive Queue 3+i*4) These bits set the unread frame counter used in reception queue 3+4*i.
b27, b26	—	Reserved These bits are read as 0. The write value should be 0.
b25, b24	RSM3[1:0]	Receive Synchronous Mode (Receive Queue 3+i*4) B'00: Mode with write-back Other than B'00: Setting prohibited
b23, b22	—	Reserved These bits are read as 0. The write value should be 0.
b21, b20	UFCC2[1:0]	Unread Frame Counter Configuration (Receive Queue 2+i*4) These bits set the unread frame counter used in reception queue 2+4*i.
b19, b18	—	Reserved These bits are read as 0. The write value should be 0.
b17, b16	RSM2[1:0]	Receive Synchronous Mode (Receive Queue 2+i*4) B'00: Mode with write-back Other than B'00: Setting prohibited
b15, b14	—	Reserved These bits are read as 0. The write value should be 0.
b13, b12	UFCC1[1:0]	Unread Frame Counter Configuration (Receive Queue 1+i*4) These bits set the unread frame counter used in reception queue 1+4*i.
b11, b10	—	Reserved These bits are read as 0. The write value should be 0.
b9, b8	RSM1[1:0]	Receive Synchronous Mode (Receive Queue 1+i*4) B'00: Mode with write-back Other than B'00: Setting prohibited
b7, b6	—	Reserved These bits are read as 0. The write value should be 0.
b5, b4	UFCC0[1:0]	Unread Frame Counter Configuration (Receive Queue 0+i*4) These bits set the unread frame counter used in reception queue 0+4*i.

chanhvinh.com

Table 45.12 RQCi register contents (2/2)

Bit Position	Bit Name	Function
b3, b2	—	Reserved These bits are read as 0. The write value should be 0.
b1, b0	RSM0[1:0]	Receive Synchronous Mode (Receive Queue 0+i*4) B'00: Mode with write-back Other than B'00: Setting prohibited

UFCCr[1:0] (r = 0 to 17) Unread Frame Counter Configuration Bits

These bits set the unread frame counter for reception queue r.

With the AVB-DMAC, four patterns of settings are available for the unread frame counter. Use the unread frame counter stop level configuration register (UFCS) to set the stop level of the unread frame counter.

When these bits are set to B'00, the stop function is disabled.

Writing to the bits is only possible when the current operating mode is configuration mode.

RSMr[1:0] (r = 0 to 17) Receive Synchronous Mode Bits

These bits set receive synchronous mode.

Set B'00 in this bit.

Writing to the bits is only possible when the current operating mode is configuration mode.

chanhvinh.com

45.2.9 Receive Padding Configuration Register (RPC)

The RPC register is used to set padding for received frames.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DCNT[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	PCNT[2:0]			—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 45.13 RPC register contents

Bit Position	Bit Name	Function
b31 to b24	—	Reserved These bits are read as 0. The write value should be 0.
b23 to b16	DCNT[7:0]	Stored Data Counter These bits specify the amount of data to be stored with the descriptor. The setting is in words. I.e. 1 in the counter indicates 1 word (4 bytes).
b15 to b11	—	Reserved These bits are read as 0. The write value should be 0.
b10 to b8	PCNT[2:0]	Stored Padding Counter These bits indicate the amount of padding to be stored in data areas for descriptors. The setting is in words. I.e. 1 in the counter indicates 1 word (4 bytes).
b7 to b0	—	Reserved These bits are read as 0. The write value should be 0.

CAUTION

Padding can be used to extend frame lengths, but frame lengths should not exceed 4 Kbytes.

DCNT[7:0] Stored Data Counter Bits

These bits specify the amount of the frame data (1 to 255) to be stored following the padding. Counting by one indicates one word (4 bytes). For example, when these bits are set to 47, the amount of data is 47 words (= 188 bytes).

When these bits are 0, all received data have been stored following the initial padding.

Writing to the bits is only possible when the current operating mode is configuration mode.

For details on padding, see section 45.3.4.3 (c) Padding.

chanhvinh.com

PCNT[2:0] Stored Padding Counter Bits

These bits specify the amount of padding to be appended to the on-chip RAM. Counting by one indicates one word (4 bytes). For example, when these bits are set to 1, the amount of padding is one word (= 4 bytes).

Writing to the bits is only possible when the current operating mode is configuration mode.

For details on padding, see section 45.3.4.3 (c) Padding.

chanhvinh.com

45.2.10 Unread Frame Counter Stop Level Configuration Register (UFCS)

The UFCS register sets the stop levels for unread frames.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	SL3[5:0]						—	—	SL2[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	SL1[5:0]						—	—	SL0[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 45.14 UFCS register contents

Bit Position	Bit Name	Function
b31, b30	—	Reserved These bits are read as 0. The write value should be 0.
b29 to b24	SL3[5:0]	Stop Level 3 Unread frame count stop level 3
b23, b22	—	Reserved These bits are read as 0. The write value should be 0.
b21 to b16	SL2[5:0]	Stop Level 2 Unread frame count stop level 2
b15, b14	—	Reserved These bits are read as 0. The write value should be 0.
b13 to b8	SL1[5:0]	Stop Level 1 Unread frame count stop level 1
b7, b6	—	Reserved These bits are read as 0. The write value should be 0.
b5 to b0	SL0[5:0]	Stop Level 0 Unread frame count stop level 0 These bits are read as 0. The write value should be 0.

SL0 to SL3[5:0] Stop Level 0 to 3 Bits

These bits set the stop levels for unread frames.

One of the four stop levels from 0 to 3 can be set for each reception queue. When these bits are set to 0, the stop function is disabled. The level to be used is specified by the receive queue configuration register i (RQCi) (i = 0 to 4).

Writing to the bits is only possible when the current operating mode is configuration mode.

chanhvinh.com

45.2.11 Unread Frame Counter Register i (UFCVi) (i = 0 to 4)

The UFCV0 register indicates the number of unread frames in reception queues 0 to 3.

The UFCV1 register indicates the number of unread frames in reception queues 4 to 7.

The UFCV2 register indicates the number of unread frames in reception queues 8 to 11.

The UFCV3 register indicates the number of unread frames in reception queues 12 to 15.

The UFCV4 register indicates the number of unread frames in reception queues 16 and 17.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	CV3[5:0]						—	—	CV2[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CV1[5:0]						—	—	CV0[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.15 UFCS0 register contents

Bit Position	Bit Name	Function
b31, b30	—	Reserved These bits are read as 0. The write value should be 0.
b29 to b24	CV3[5:0]	Unread Frame Count 3+4*i Number of unread frames in reception queue 3+4*i
b23, b22	—	Reserved These bits are read as 0. The write value should be 0.
b21 to b16	CV2[5:0]	Unread Frame Count 2+4*i Number of unread frames in reception queue 2+4*i
b15, b14	—	Reserved These bits are read as 0. The write value should be 0.
b13 to b8	CV1[5:0]	Unread Frame Count 1+4*i Number of unread frames in reception queue 1+4*i
b7, b6	—	Reserved These bits are read as 0. The write value should be 0.
b5 to b0	CV0[5:0]	Unread Frame Count 0+4*i Number of unread frames in reception queue 0+4*i

CVr[5:0] Unread Frame Count r (r = 0 to 17) Bits

These bits indicate the number of unread frames in reception queue r.

The number of unread frames is decremented by the value that is written to the unread frame counter decrement register i (UFCDi).

For a description of how to use unread frames, refer to section 45.3.4.4, Unread Frame Counters.

chanhvinh.com

Conditions for updating:

The bits are set to 0 when the operating mode is not operation mode and when the descriptor base address load request register (DLR) issues a base address load request.

The value of these bits is incremented when a frame is stored in reception queue r normally and a write transaction to the corresponding descriptor is issued. (The maximum setting is H'3F. If the value exceeds H'3F, incrementation will not proceed.) Note that the value is incremented without waiting for response to the write transaction. Therefore confirm that the descriptor type (DESCR.DT) of the corresponding descriptor has been updated before processing the received data.

The value is decremented by the value written to the unread frame counter decrement register i

(UFCDi). Confirm that the descriptor types (DESCR.DT) of all the target descriptors for decrementing have been updated before decrementing.

chanhvinh.com

45.2.12 Unread Frame Counter Decrement Register i (UFCDi) (i = 0 to 4)

The UFCD0 register is used to decrement unread counters in reception queues 0 to 3.

The UFCD1 register is used to decrement unread counters in reception queues 4 to 7.

The UFCD2 register is used to decrement unread counters in reception queues 8 to 11.

The UFCD3 register is used to decrement unread counters in reception queues 12 to 15.

The UFCD4 register is used to decrement unread counters in reception queues 16 and 17.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	DV3[5:0]						—	—	DV2[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	DV1[5:0]						—	—	DV0[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.16 UFCDi register contents

Bit Position	Bit Name	Function
b31, b30	—	Reserved These bits are read as 0. The write value should be 0.
b29 to b24	DV3[5:0]	Unread Frame Decrement Value 3+4*i Unread frame decrement value for reception queue 3+4*i
b23, b22	—	Reserved These bits are read as 0. The write value should be 0.
b21 to b16	DV2[5:0]	Unread Frame Decrement Value 2+4*i Unread frame decrement value for reception queue 2+4*i
b15, b14	—	Reserved These bits are read as 0. The write value should be 0.
b13 to b8	DV1[5:0]	Unread Frame Decrement Value 1+4*i Unread frame decrement value for reception queue 1+4*i
b7, b6	—	Reserved These bits are read as 0. The write value should be 0.
b5 to b0	DV0[5:0]	Unread Frame Decrement Value 0+4*i Unread frame decrement value for reception queue 0+4*i

DVr[5:0] Unread Frame Decrement Value r (r = 0 to 17) Bits

These bits set the decrement value for unread frames in reception queue r. The value of an unread frame counter register i (UFCVi) (i = 0 to 4) is decremented by the value set in the corresponding bits of this register.

Write H'3F to these bits to reset the unread counters in reception queue r.

These bits are always read as 0.

chanhvinh.com

45.2.13 Separation Filter Offset Register (SFO)

The SFO register sets an offset into frames for use by the separation filter.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	FBP[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.17 SFO register contents

Bit Position	Bit Name	Function
b31 to b6	—	Reserved These bits are read as 0. The write value should be 0.
b5 to b0	FBP[5:0]	First Byte Position Position in Ethernet frames of the first byte of the bytes to be used by the separation filter

FBP[5:0] First Byte Position Bits

These bits set the position in Ethernet frames of the first byte of the bytes to be used by the separation filter.

When these bits are 0, the separation filter starts from the start of each Ethernet frame (first byte of the destination address). For bytes in Ethernet frames, see Figure 45.2, in section 45.2.1, AVB-DMAC Mode Register (CCC).

Writing to the bits is only possible when the current operating mode is configuration mode.

For separation filtering, see section 45.3.4.1 (1) Separation Filtering.

CAUTION

Received frames having fewer bytes than the setting of these bits + 8 bytes are judged to be non-matching by the separation filter. In this case, the data will either be sorted into a reception queue or discarded in accord with the setting of the separation filtering select bits in the receive configuration register (RCR.ESF).

chanhvinh.com

45.2.14 Separation Filter Pattern Register i (SFPi) (i = 0 to 31)

Pairs of SFPi registers set the pattern for the separation filters to be used by the corresponding reception queues 2 to 17.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	FPs[31+32*(i%2):16+32*(i%2)]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FPs[15+32*(i%2):0+32*(i%2)]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.18 SFPi register contents

Bit Position	Bit Name	Function
b31 to b0	FPs[31+32*(i%2):0+32*(i%2)]	Separation Filter Pattern These bits set the pattern of the separation filter. The 64-bit filter pattern is set for each queue.

FPs[63:0](s = i/2) Separation Filter Pattern Bits

These bits set the pattern for a separation filter to be used with reception queues 2 to 17 (for streams 0 to 15).

Each queue has a 64-bit setting; reception queue 2 (for stream 0) uses SFP0 and SFP1, reception queue 17 (for stream 15) uses SFP30 and SFP31, and so on.

SFPi.FPs[7:0] (where i is an even number) are used for the byte of Ethernet frame data specified by the separation filter offset register, while SFPi.FPs[63:56] (where i is the corresponding odd number) are used for the byte at the address specified by the separation filter offset register (SFO) + 7.

Data from received frames passes the separation filter when the following formula is satisfied.

(Data from received frames & SFMi.CFM) == SFPi.FPs

Eight bytes from the position specified by the start byte position bits (SFO.FBP) of data in the received frames are used by the separation filter.

Writing to the bits is only possible when the current operating mode is configuration mode.

For separation filtering, see section 45.3.4.1 (1) Separation Filtering.

chanhvinh.com

45.2.15 Separation Filter Mask Register i (SFMi) (i = 0 or 1)

A pair of SFMi registers sets the mask value for the separation filter used by the corresponding reception queue 2 to 17.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CFM[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CFM[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.19 SFMi register contents

Bit Position	Bit Name	Function
b31 to b0	CFM[31:0]	Separation Filter Mask These bits set the mask value for the separation filter.

CFM[63:0] Separation Filter Mask Bits

These bits set the mask value for the separation filter for use with the corresponding reception queue 2 to 17 (stream 0 to 15).

SFM0.CFM[7:0] are used for bytes of Ethernet frame data specified by the separation filter offset register, while SFM1.CFM[63:56] are used for the separation filter offset register (SFO) + 7.

Writing to the bits is only possible when the current operating mode is configuration mode.

For separation filtering, see section 45.3.4.1 (1) Separation Filtering.

chanhvinh.com

45.2.16 Transmit Configuration Register (TGC)

The TGC register is used to make settings related to transmission for the AVB-DMAC.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	TBD3[1:0]		—	—	TBD2[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	TBD1[1:0]		—	—	TBD0[1:0]		—	—	TQP[1:0]		TSM3	TSM2	TSM1	TSM0
Initial value	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.20 TGC register contents (1/2)

Bit Position	Bit Name	Function
b31 to b22	—	Reserved These bits are read as 0. The write value should be 0.
b21, b20	TBD3[1:0]	Transmit FIFO Size (Stream Class A) Number of frames to be fetched from transmission queue 3 (for stream class A) **CAUTION** Write 2 to these bits.
b19, b18	—	Reserved These bits are read as 0. The write value should be 0.
b17, b16	TBD2[1:0]	Transmit FIFO Size (Stream Class B) Number of frames to be fetched from transmission queue 2 (for stream class B) **CAUTION** Write 2 to these bits.
b15, b14	—	Reserved These bits are read as 0. The write value should be 0.
b13, b12	TBD1[1:0]	Transmit FIFO Size (Network Control) Number of frames to be fetched from transmission queue 1 (for network control) **CAUTION** Write 2 to these bits.
b11, b10	—	Reserved These bits are read as 0. The write value should be 0.
b9, b8	TBD0[1:0]	Transmit FIFO Size (Best Effort) Number of frames to be fetched from transmission queue 0 (for best effort) **CAUTION** Write 2 to these bits.
b7, b6	—	Reserved These bits are read as 0. The write value should be 0.
b5, b4	TQP[1:0]	Transmit Queue Priority 00: Non-ABV mode 01: AVB mode 1 10: Setting prohibited 11: AVB mode 2

chanhvinh.com

Table 45.20 TGC register contents (2/2)

Bit Position	Bit Name	Function
b3	TSM3	Transmit Synchronous Mode (Stream Class A) 0: With write-back 1: Setting prohibited
b2	TSM2	Transmit Synchronous Mode (Stream Class B) 0: With write-back 1: Setting prohibited
b1	TSM1	Transmit Synchronous Mode (Network Control) 0: With write-back 1: Setting prohibited
b0	TSM0	Transmit Synchronous Mode (Best Effort) 0: With write-back 1: Setting prohibited

TBD0 to TBD3[1:0] Transmit FIFO Size (Stream Class A/ Stream Class B/Network Control/ Best Effort) Bits

These bits set the sizes of the transmission FIFO buffers for use with each of the transmission queues.

Writing to these bits is only possible when the current operating mode is configuration mode.

Set these bits to 2.

TQP[1:0] Transmit Queue Priority Bits

These bits set the priority of the transmission queues.

B’00: Non-AVB mode: Q3→Q2→Q1→Q0

B’01: AVB mode 1: Q3 (CBS)→Q2 (CBS)→Q1→Q0

B’10: Setting prohibited

B’11: AVB mode 2: Q1→Q3 (CBS)→Q2 (CBS)→Q0

For the credit-based shaping (CBS) algorithm, see section 45.3.6, CBS (Credit-Based Shaping).

The CBS algorithm is not applied in non-AVB mode (i.e. when the value is B’00).

Writing to the bits is only possible when the current operating mode is configuration mode.

TSM0 to TSM3 Transmit Synchronous Bits

Set these bits to 0.

chanhvinh.com

45.2.17 Transmit Configuration Control Register (TCCR)

The TCCR register controls transmission by the AVB-DMAC and is used to make related settings.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TFR	TFEN	—	—	—	—	TSRQ3	TSRQ2	TSRQ1	TSRQ0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 45.21 TCCR register contents

Bit Position	Bit Name	Function
b31 to b10	—	Reserved These bits are read as 0. The write value should be 0.
b9	TFR	Time Stamp FIFO Release 0(W): No request to the time-stamp FIFO. 1(W): Releases the oldest entry in the time-stamp FIFO.
b8	TFEN	Time Stamp FIFO Enable 0: Recording of transmission time stamps in the time-stamp FIFO is disabled. 1: Recording of transmission time stamps in the time-stamp FIFO is enabled.
b7 to b4	—	Reserved These bits are read as 0. The write value should be 0.
b3	TSRQ3	Transmit Start Request (Queue 3 (Stream Class A)) 0(R): Transmission queue is empty or stopped. 1(W): A transmission start request is issued. 1(R): Fetching of data for transmission is pending.
b2	TSRQ2	Transmit Start Request (Queue 2 (Stream Class B)) 0(R): Transmission queue is empty or stopped. 1(W): A transmission start request is issued. 1(R): Fetching of data for transmission is pending.
b1	TSRQ1	Transmit Start Request (Queue 1 (Network Control)) 0(R): Transmission queue is empty or stopped. 1(W): A transmission start request is issued. 1(R): Fetching of data for transmission is pending.
b0	TSRQ0	Transmit Start Request (Queue 0 (Best Effort)) 0(R): Transmission queue is empty or stopped. 1(W): A transmission start request is issued. 1(R): Fetching of data for transmission is pending.

TFR Time Stamp FIFO Release Bit

This bit releases the oldest entry in the time-stamp FIFO.

When 1 is written to this bit, the TFAi register is updated by the oldest entry in the time-stamp FIFO.

For a description of how to use the time-stamp FIFO, see section 45.3.5.4, Time Stamping in Transmission.

This bit is always read as 0.

When the time stamp FIFO count bits (TSR.TFFL) are set to B'000, 1 should not be written to this bit.

chanhvinh.com

TFEN Time Stamp FIFO Enable Bit

This bit enables storage in the time-stamp FIFO.

When it is set, time-stamp information is stored for descriptors with DESCR.TSR set to 1 (for DESCR.TSR, see section 45.3.5.2 (2) Configuration of Transmission Frame Data Descriptors.

When 0 is set in this bit, no entries are made in the time-stamp FIFO.

For a description of how to use the time-stamp FIFO, see section 45.3.5.4, Time Stamping in Transmission.

TSRQt Transmit Start Request (Queue t) (t = 0 to 3) Bit

This bit issues a request to start transmission for transmission queue t.

When read, this bit being set to 1 indicates that transmission queue t has a frame that has not yet been fetched to the transmission FIFO.

Frame transmission by the E-MAC is processed independently from fetching to the transmission FIFO. The timing of transmission from a queue depends on the priority order of transmission.

For the scheduling of transmission queues, see section 45.3.5.1, Transmission Modes.

Writing to this bit is only possible when the current operating mode is configuration mode.

Only 1 can be written to the bit. Writing 0 to the bit has no effect.

Conditions for updating:

The bit is set to 0 when the operating mode is not operation mode, when a descriptor of type EEMPTY, FEMPTY, or LEMPTY (no usable data) is processed, when an EOS descriptor is processed, and when a descriptor with defective data is processed.

chanhvinh.com

45.2.18 Transmit Status Register (TSR)

The TSR register indicates the state of transmission by the AVB-DMAC.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TFFL[2:0]			—	—	—	—	CCS1[1:0]		CCS0[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.22 TSR register contents

Bit Position	Bit Name	Function
b31 to b11	—	Reserved These bits are read as 0. The write value should be 0.
b10 to b8	TFFL[2:0]	Time Stamp FIFO Count Number of time-stamp FIFOs
b7 to b4	—	Reserved These bits are read as 0. The write value should be 0.
b3, b2	CCS1[1:0]	CBS Counter Status 1 (Class A) B'00: The current credit value is within the limit. B'01: The current credit value is less than or equal to the lower limit. B'10: The current credit value is greater than or equal to the upper limit. B'11: (Reserved)
b1, b0	CCS0[1:0]	CBS Counter Status 0 (Class B) B'00: The current credit value is within the limit. B'01: The current credit value is less than or equal to the lower limit. B'10: The current credit value is greater than or equal to the upper limit. B'11: (Reserved)

TFFL[3:0] Time Stamp FIFO Count Bits

These bits indicate the number of time stamps in the time-stamp FIFO.

The values 0 and 2 indicate that the time-stamp FIFO is empty and full, respectively (values 3 to 7 are reserved).

Conditions for updating:

The bits are set to 0 when the operating mode is not operation mode and when the time stamp FIFO enable bit in the transmit configuration control register (TCCR.TFEN) = 0.

When the time stamp FIFO enable bit (TCCR.TFEN) is 1 and these bits are not 2, the value of these bits is incremented after a frame with DESCR.TSR set has been transmitted by the E-MAC (for DESCR.TSR, see section 45.3.5.2 (2) Configuration of Transmission Frame Data Descriptors.

The value of these bits is decremented if it is not 0 when 1 is written to the time stamp FIFO release bit in the transmit configuration control register (TCCR.TFR).

chanhvinh.com

CCS0 and CCS1[1:0] CBS Counter Status 0 and 1 Bits

These bits indicate the CBS (credit-based shaping) state of stream data transmission queues 0 and 1. If the calculated credit value is outside the range specified by CBS upper limit register c (CULc) and CBS lower limit register c (CLLc), it falls outside the range for CBS.

Conditions for updating:

The bits are set to B'00 when the operating mode is not operation mode.

The bits are set to B'00 when the credit value calculated by the CBS is within the range specified by the CBS upper limit register c (CULc) (c = 0, 1) and the CBS lower limit register c (CLLc) (c = 0, 1).

The bits are set to B'01 if the credit value calculated by the CBS is lower than the value in CBS lower limit register c (CLLc).

The bits are set to B'10 if the credit value calculated by the CBS is higher than the value in CBS upper limit register c (CULc).

chanhvinh.com

45.2.19 Time Stamp FIFO Access Register 0 (TFA0)

TFA0 indicates the time stamp value.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TSV[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSV[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.23 TFA0 register contents

Bit Position	Bit Name	Function
b31 to b0	TSV[31:0]	Time Stamp Value Time stamp value

TSV[79:0] Time Stamp Value Bits

These 80 bits consist of TFA0.TSV[31:0], TFA1.TSV[63:32], and TFA2.TSV[79:64], which together indicate the oldest time stamp value stored in the time-stamp FIFO.

Once the time-stamp FIFO is full, no further time-stamp values are stored.

Conditions for updating:

The bits are set to H'0000 0000 when the operating mode is not operation mode.

The register is updated whenever a value is stored in the time-stamp FIFO (when the time-stamp FIFO count bit in the transmit status register (TSR.TFFL) changes from 0 to 1).

The register is updated when the oldest entry is released (when the time stamp FIFO release bit in the transmit configuration control register (TCCR.TFR) is set to 1).

chanhvinh.com

45.2.20 Time Stamp FIFO Access Register 1 (TFA1)

The TFA1 register indicates the time-stamp value.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TSV[63:48]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSV[47:32]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.24 TFA1 register contents

Bit Position	Bit Name	Function
b31 to b0	TSV[63:32]	Time Stamp Value Time-stamp value

TSV[63:32] Time Stamp Value Bits

For details, see section 45.2.19, Time Stamp FIFO Access Register 0 (TFA0).

chanhvinh.com

45.2.21 Time Stamp FIFO Access Register 2 (TFA2)

The TFA2 register indicates the time-stamp tag and part of the time-stamp value.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	TST[9:0]									
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSV[79:64]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.25 TFA2 register contents

Bit Position	Bit Name	Function
b31 to b26	—	Reserved These bits are read as 0.
b25 to b16	TST[9:0]	Time Stamp Tag Time-stamp tag
b15 to b0	TSV[79:64]	Time Stamp Value Time-stamp value

TST[9:0] Time Stamp Tag Bits

These bits indicate the contents of the DESCR.TAG bit within the descriptor for frame transmission. These values are used to check the correlation between frames within the transmission queue and the time-stamp values (accessible through time stamp FIFO access register i (TFAi.TSV, i = 0 to 2)) which can be placed in the FIFO.

For the tagging of frames in transmission, see section 45.3.5.4, Time Stamping in Transmission.

When the time stamp FIFO count bits (TSR.TFFL) are 0, do not refer to these bits.

Conditions for updating:

The bits are set to H'000 when the operating mode is not operation mode.

Updated when a value is stored in the time-stamp FIFO (when the value of the time stamp FIFO count bit in the transmit status register (TSR.TFFL) changes from 0 to 1).

Updated when the oldest entry has been released (1 is set in the time stamp FIFO release bit in the transmit configuration control register (TCCR.TFR)).

TSV[73:64] Time Stamp Value Bits

For details, see section 45.2.19, Time Stamp FIFO Access Register 0 (TFA0).

chanhvinh.com

45.2.22 CBS Increment Value Register c (CIVRc) (c = 0 or 1)

The CIVR0 register sets the increment in the CBS algorithm for transmission queue 2 (for stream class B).

The CIVR1 register sets the increment in the CBS algorithm for transmission queue 3 (for stream class A).

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CIV[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CIV[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.26 CIVRc register contents

Bit Position	Bit Name	Function
b31 to b0	CIV[31:0]	CBS Increment Value Setting value: 1 to 65535 (H'00000001 to H'0000FFFF)

CIV[31:0] CBS Increment Value Bits

These bits set the increment for the CBS algorithm.

Set a value in the range from 1 to 65535 (H'0000 0001 to H'0000 FFFF).

The value to be written to these bits depends on the Ethernet bit rate and Bϕ (internal bus clock). For details, see section 45.3.6, CBS (Credit-Based Shaping).

chanhvinh.com

45.2.23 CBS Decrement Value Register c (CDVRc) (c = 0 or 1)

The CDVR0 register sets the decrement in the CBS algorithm for transmission queue 2 (for stream class B).

The CDVR1 register sets the decrement in the CBS algorithm for transmission queue 3 (for stream class A).

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CDV[31:16]															
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CDV[15:0]															
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.27 CDVR0 and CDVR1 register contents

Bit Position	Bit Name	Function
b31 to b0	CDV[31:0]	CBS Decrement Value Setting value: -1 to -65536 (H'FFFF FFFF to H'FFFF 0000)

CDV[31:0] CBS Decrement Value Bits

These bits set the decrement for the CBS algorithm.

Set a negative value from -1 to -65536 (H'FFFF FFFF to H'FFFF 0000).

The value to be written to these bits depends on the Ethernet bit rate and Bϕ (internal bus clock). For details, see section 45.3.6, CBS (Credit-Based Shaping).

chanhvinh.com

45.2.24 CBS Upper Limit Register c (CULc) (c = 0 or 1)

The CUL0 register sets the upper limit for credit values calculated by using the CSB algorithm for transmission queue 2 (for stream class B).

The CUL1 register sets the upper limit for credit values calculated by using the CSB algorithm for transmission queue 3 (for stream class A).

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ULV[31:16]															
Initial value	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ULV[15:0]															
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.28 CUL0 and CUL1 register contents

Bit Position	Bit Name	Function
b31?b0	ULV[31:0]	CBS Upper Limit Upper limit on CBS values

ULV[31:0] CBS Upper Limit Bits

These bits set the upper limit for credit values calculated by using the CBS algorithm.

The setting is a limiting value for error detection and does not normally affect operation of the algorithm.

Write a positive value to these bits.

For details, see section 45.3.6, CBS (Credit-Based Shaping).

chanhvinh.com

45.2.25 CBS Lower Limit Register c (CLLc) (c = 0 or 1)

The CUL0 register sets the lower limit for credit values calculated by using the CSB algorithm for transmission queue 2 (for stream class B).

The CUL1 register sets the lower limit for credit values calculated by using the CSB algorithm for transmission queue 3 (for stream class A).

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	LLV[31:16]															
Initial value	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LLV[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.29 CULc register contents

Bit Position	Bit Name	Function
b31 to b0	LLV[31:0]	CBS Lower Limit Lower limit on CBS values

LLV[31:0] CBS Lower Limit Bits

These bits set the lower limit for credit values calculated by using the CBS algorithm.

The setting is a limiting value for error detection and does not normally affect operation of the algorithm.

Write a negative value to these bits.

For details, see section 45.3.6, CBS (Credit-Based Shaping).

chanhvinh.com

45.2.26 Descriptor Interrupt Control Register (DIC)

The DIC register is used to control descriptor interrupts 1 to 15.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DPE15	DPE14	DPE13	DPE12	DPE11	DPE10	DPE9	DPE8	DPE7	DPE6	DPE5	DPE4	DPE3	DPE2	DPE1	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 45.30 DIC register contents (1/2)

Bit Position	Bit Name	Function
b31 to b16	—	Reserved These bits are read as 0. The write value should be 0.
b15	DPE15	Descriptor Interrupt Enable 15 0: Disabled 1: Enabled
b14	DPE14	Descriptor Interrupt Enable 14 0: Disabled 1: Enabled
b13	DPE13	Descriptor Interrupt Enable 13 0: Disabled 1: Enabled
b12	DPE12	Descriptor Interrupt Enable 12 0: Disabled 1: Enabled
b11	DPE11	Descriptor Interrupt Enable 11 0: Disabled 1: Enabled
b10	DPE10	Descriptor Interrupt Enable 10 0: Disabled 1: Enabled
b9	DPE9	Descriptor Interrupt Enable 9 0: Disabled 1: Enabled
b8	DPE8	Descriptor Interrupt Enable 8 0: Disabled 1: Enabled
b7	DPE7	Descriptor Interrupt Enable 7 0: Disabled 1: Enabled
b6	DPE6	Descriptor Interrupt Enable 6 0: Disabled 1: Enabled
b5	DPE5	Descriptor Interrupt Enable 5 0: Disabled 1: Enabled
b4	DPE4	Descriptor Interrupt Enable 4 0: Disabled 1: Enabled

chanhvinh.com

Table 45.30 DIC register contents (2/2)

Bit Position	Bit Name	Function
b3	DPE3	Descriptor Interrupt Enable 3 0: Disabled 1: Enabled
b2	DPE2	Descriptor Interrupt Enable 2 0: Disabled 1: Enabled
b1	DPE1	Descriptor Interrupt Enable 1 0: Disabled 1: Enabled
b0	—	Reserved This bit is read as 0. The write value should be 0.

RPE1 to RPE15 Descriptor Interrupt Enable Bits 1 to 15

When an interrupt source flag is set (a bit from among the DPF1 to DPF15 bits in the descriptor interrupt status register (DIS) = 1) while the interrupt is enabled, the interrupt is issued.

chanhvinh.com

45.2.27 Descriptor Interrupt Status Register (DIS)

The DIS register indicates the state of descriptor interrupts.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DPF15	DPF14	DPF13	DPF12	DPF11	DPF10	DPF9	DPF8	DPF7	DPF6	DPF5	DPF4	DPF3	DPF2	DPF1	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 45.31 DIS register contents (1/2)

Bit Position	Bit Name	Function
b31 to b16	—	Reserved These bits are read as 0. The write value should be 0.
b15	DPF15	Descriptor Interrupt Status15 0: The interrupt is not pending. 1: The interrupt is pending.
b14	DPF14	Descriptor Interrupt Status14 0: The interrupt is not pending. 1: The interrupt is pending.
b13	DPF13	Descriptor Interrupt Status13 0: The interrupt is not pending. 1: The interrupt is pending.
b12	DPF12	Descriptor Interrupt Status12 0: The interrupt is not pending. 1: The interrupt is pending.
b11	DPF11	Descriptor Interrupt Status11 0: The interrupt is not pending. 1: The interrupt is pending.
b10	DPF10	Descriptor Interrupt Status10 0: The interrupt is not pending. 1: The interrupt is pending.
b9	DPF9	Descriptor Interrupt Status9 0: The interrupt is not pending. 1: The interrupt is pending.
b8	DPF8	Descriptor Interrupt Status8 0: The interrupt is not pending. 1: The interrupt is pending.
b7	DPF7	Descriptor Interrupt Status7 0: The interrupt is not pending. 1: The interrupt is pending.
b6	DPF6	Descriptor Interrupt Status6 0: The interrupt is not pending. 1: The interrupt is pending.
b5	DPF5	Descriptor Interrupt Status5 0: The interrupt is not pending. 1: The interrupt is pending.
b4	DPF4	Descriptor Interrupt Status4 0: The interrupt is not pending. 1: The interrupt is pending.

chanhvinh.com

Table 45.31 DIS register contents (2/2)

Bit Position	Bit Name	Function
b3	DPF3	Descriptor Interrupt Status3 0: The interrupt is not pending. 1: The interrupt is pending.
b2	DPF2	Descriptor Interrupt Status2 0: The interrupt is not pending. 1: The interrupt is pending.
b1	DPF1	Descriptor Interrupt Status1 0: The interrupt is not pending. 1: The interrupt is pending.
b0	—	Reserved This bit is read as 0. The write value should be 0.

DPF1 to DPF15 Descriptor Interrupt Status Bits

The corresponding bit indicates completion of the processing of a descriptor with DESCR.DIE set to the corresponding number from 1 to 15 within the reception or transmission queue.

When DESCR.DIE is 0, the descriptor interrupt is not generated.

Only 0 can be written to these bits.

[Conditions for Changing]

A bit is set to 0 when the operating mode is not operation mode.

A bit is set to 1 when a descriptor with DESCR.DIE set to the corresponding number from 1 to 15 is processed and a write transaction to the corresponding descriptor is issued. Note that the bit is set to 1 without waiting for response to the write transaction. Therefore confirm that the descriptor type (DESCR.DT) of the corresponding descriptor has been updated before post-processing the processed descriptor.

chanhvinh.com

45.2.28 Error Interrupt Control Register (EIC)

The EIC register controls the AVB-DMAC-related error interrupts.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TFFE	CULE1	CULE0	CLLE1	CLLE0	SEE	QEE	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 45.32 EIC register contents

Bit Position	Bit Name	Function
b31 to b9	—	Reserved These bits are read as 0. The write value should be 0.
b8	TFFE	Time Stamp FIFO Full-Error Interrupt Enable 0: Disabled 1: Enabled
b7	CULE1	CBS Upper Limit Error Interrupt Enable (Class A) 0: Disabled 1: Enabled
b6	CULE0	CBS Upper Limit Error Interrupt Enable (Class B) 0: Disabled 1: Enabled
b5	CLLE1	CBS Lower Limit Error Interrupt Enable (Class A) 0: Disabled 1: Enabled
b4	CLLE0	CBS Lower Limit Error Interrupt Enable (Class B) 0: Disabled 1: Enabled
b3	SEE	Separation Filter Error Interrupt Enable 0: Disabled 1: Enabled
b2	QEE	Queue Error Interrupt Enable 0: Disabled 1: Enabled
b1, b0	—	Reserved These bits are read as 0. The write value should be 0.

TFFE Time Stamp FIFO Full-Error Interrupt Enable Bits

When the time stamp FIFO is full (TFFF in the error interrupt status register (EIS) = 1) and the interrupt is enabled, the interrupt is issued.

CULE1 CBS Upper Limit Error Interrupt Enable Bit (Class A)

When the Class A CBS reaches its upper limit (CULF1 in the error interrupt status register (EIS) = 1) and the interrupt is enabled, the interrupt is issued.

chanhvinh.com

CULE0 CBS Upper Limit Error Interrupt Enable Bit (Class B)

When the Class B CBS reaches its upper limit (CULF0 in the error interrupt status register (EIS) = 1) and the interrupt is enabled, the interrupt is issued.

CLLE1 CBS Lower Limit Error Interrupt Enable Bit (Class A)

When the Class A CBS reaches its lower limit (CLLF1 in the error interrupt status register (EIS) = 1) and the interrupt is enabled, the interrupt is issued.

CLLE0 CBS Lower Limit Error Interrupt Enable Bit (Class B)

When the Class B CBS reaches its lower limit (CLLF0 in the error interrupt status register (EIS) = 1) and the interrupt is enabled, the interrupt is issued.

SEE Separation Filter Error Interrupt Enable Bit

When the separation filter error is detected (SEF in the error interrupt status register (EIS) = 1) and the interrupt is enabled, the interrupt is issued.

QEE Queue Error Interrupt Enable Bit

When the queue error is detected (QEF in the error interrupt status register (EIS) = 1) and the interrupt is enabled, the interrupt is issued.

chanhvinh.com

45.2.29 Error Interrupt Status Register (EIS)

The EIS register indicates the states of AVB-DMAC-related error interrupts.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	QFS
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TFFF	CULF1	CULF0	CLLF1	CLLF0	SEF	QEF	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 45.33 EIS register contents

Bit Position	Bit Name	Function
b31 to b17	—	Reserved These bits are read as 0. The write value should be 0.
b16	QFS	Queue Full Error Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b15 to b9	—	Reserved These bits are read as 0. The write value should be 0.
b8	TFFF	Time Stamp FIFO Full Error Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b7	CULF1	CBS Upper Limit Error Interrupt Status (Class A) 0: The interrupt is not pending. 1: The interrupt is pending.
b6	CULF0	CBS Upper Limit Error Interrupt Status (Class B) 0: The interrupt is not pending. 1: The interrupt is pending.
b5	CLLF1	CBS Lower Limit Error Interrupt Status (Class A) 0: The interrupt is not pending. 1: The interrupt is pending.
b4	CLLF0	CBS Lower Limit Error Interrupt Status (Class B) 0: The interrupt is not pending. 1: The interrupt is pending.
b3	SEF	Separation Filter Error Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b2	QEF	Queue Error Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b1, b0	—	Reserved These bits are read as 0. The write value should be 0.

chanhvinh.com

QFS Queue Full Error Status Bit

With the interrupts enabled, this bit indicates that a queue is full (the receive queue r full interrupt status bit (QFFr) or the receive FIFO full interrupt status bit (RFFF) in receive interrupt status register 2 (RIS2) = 1).

[Conditions for Changing]

If the receive queue r full interrupt status bit (RIS2.QFFr) and the receive queue r full interrupt enable bit in the receive interrupt control register 2 (RIC2.QFEr) are updated, this bit is also updated.

If the receive FIFO full interrupt status bit (RIS2.RFFF) and the receive FIFO full interrupt enable bit (RIC2.RFFE) are updated, this bit is also updated.

TFFF Time Stamp FIFO Full-Error Interrupt Status Bit

This bit indicates that a new transmission time stamp has been discarded due to the time-stamp FIFO being full (i.e. has reached the overflow state).

Only 0 can be written to the bit.

[Conditions for Changing]

The bit is set to 0 when the operating mode is not operation mode.

The bit is set to 1 when a frame with DESCR.TSR set is transmitted while the time stamp FIFO enable bit in the transmit configuration control register (TCCR.TFEN) is set to 1 and the time stamp FIFO count bit in the transmit status register (TSR.TFFL) is set to 2.

CULF1 CBS Upper Limit Error Interrupt Status Bit (Class A)

This bit indicates that CBS counter 1 has exceeded the set upper limit (CUL1.ULV in the CBS upper limit register c (CULc)).

Only 0 can be written to the bit.

[Conditions for Changing]

This bit is set to 0 when the operating mode is not operation mode.

This bit is set to 1 when the value of the CBS counter status 1 (Class A) bits in the transmit status register (TSR.CCS1) change from 00 (indicating a value within the range between the limits) to 10 (indicating a value over the upper limit).

CULF0 CBS Upper Limit Error Interrupt Status Bit (Class B)

This bit indicates that CBS counter 0 has exceeded the set upper limit (CUL0.ULV in the CBS upper limit register c (CULc)).

Only 0 can be written to the bit.

[Conditions for Changing]

The bit is set to 0 when the operating mode is not operation mode.

The bit is set to 1 when the value of the CBS counter status 0 (Class B) bit in the transmit status register changes from 00 (indicating a value within the range between the limits) to 10 (indicating a value over the upper limit).

chanhvinh.com

CULF1 CBS Lower Limit Error Interrupt Status Bit (Class A)

This bit indicates that CBS counter 1 has fallen below the set lower limit (CLL1.LLV in CBS lower limit register c (CLLc)).

Only 0 can be written to the bit.

[Conditions for Changing]

The bit is set to 0 when the operating mode is not operation mode.

The bit is set to 1 when the value of the CBS counter status 1 (Class A) bit in the transmit status register (TSR.CCS1) changes from 00 (indicating a value within the range between the limits) to 01 (indicating a value less than the lower limit).

CLLF0 CBS Lower Limit Error Interrupt Status Bit (Class B)

This bit indicates that CBS counter 0 has fallen below the set lower limit (CLL0.LLV in the CBS lower limit register c (CLLc)).

Only 0 can be written to the bit.

[Conditions for Changing]

The bit is set to 0 when the operating mode is not operation mode.

The bit is set to 1 when the value of the CBS counter status 0 (Class B) bit in the transmit status register (TSR.CCS0) changes from 00 (indicating a value within the range between the limits) to 01 (indicating a value less than the lower limit).

SEF Separation Filter Error Interrupt Status Bit

This bit indicates that a received frame has been discarded due to non-matching of the defined separation filter.

Only 0 can be written to this bit.

[Conditions for Changing]

The bit is set to 0 when the operating mode is not operation mode.

If the stream filtering select bits in the receive configuration register (RCR.ESF) are B’10, this bit is set to 1 when the received frame is discarded because the received AVB stream data frame has not match any of the separation filters.

QEF Queue Error Interrupt Status Bit

This bit indicates that an error is detected when the transmission and reception queues are processed. For details, see section 45.3.2.3, Checking Integrity.

Only 0 can be written to this bit.

[Conditions for Changing]

The bit is set to 0 when the operating mode is not operation mode.

If an error in the transmission and reception queues is detected, this bit is set to 1.

chanhvinh.com

45.2.30 Receive Interrupt Control Register 0 (RIC0)

The RIC0 register controls the AVB-DMAC receive interrupts.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FRE17	FRE16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FRE15	FRE14	FRE13	FRE12	FRE11	FRE10	FRE9	FRE8	FRE7	FRE6	FRE5	FRE4	FRE3	FRE2	FRE1	FRE0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.34 RIC0 register contents (1/2)

Bit Position	Bit Name	Function
b31 to b18	—	Reserved These bits are read as 0. The write value should be 0.
b17	FRE17	Receive Frame Interrupt Enable 17 (Stream) 0: Disabled 1: Enabled
b16	FRE16	Receive Frame Interrupt Enable 16 (Stream) 0: Disabled 1: Enabled
b15	FRE15	Receive Frame Interrupt Enable 15 (Stream) 0: Disabled 1: Enabled
b14	FRE14	Receive Frame Interrupt Enable 14 (Stream) 0: Disabled 1: Enabled
b13	FRE13	Receive Frame Interrupt Enable 13 (Stream) 0: Disabled 1: Enabled
b12	FRE12	Receive Frame Interrupt Enable 12 (Stream) 0: Disabled 1: Enabled
b11	FRE11	Receive Frame Interrupt Enable 11 (Stream) 0: Disabled 1: Enabled
b10	FRE10	Receive Frame Interrupt Enable 10 (Stream) 0: Disabled 1: Enabled
b9	FRE9	Receive Frame Interrupt Enable 9 (Stream) 0: Disabled 1: Enabled
b8	FRE8	Receive Frame Interrupt Enable 8 (Stream) 0: Disabled 1: Enabled
b7	FRE7	Receive Frame Interrupt Enable 7 (Stream) 0: Disabled 1: Enabled
b6	FRE6	Receive Frame Interrupt Enable 6 (Stream) 0: Disabled 1: Enabled

chanhvinh.com

Table 45.34 RIC0 register contents (2/2)

Bit Position	Bit Name	Function
b5	FRE5	Receive Frame Interrupt Enable 5 (Stream) 0: Disabled 1: Enabled
b4	FRE4	Receive Frame Interrupt Enable 4 (Stream) 0: Disabled 1: Enabled
b3	FRE3	Receive Frame Interrupt Enable 3 (Stream) 0: Disabled 1: Enabled
b2	FRE2	Receive Frame Interrupt Enable 2 (Stream) 0: Disabled 1: Enabled
b1	FRE1	Receive Frame Interrupt Enable 1 (Network Control) 0: Disabled 1: Enabled
b0	FRE0	Receive Frame Interrupt Enable 0 (Best Effort) 0: Disabled 1: Enabled

FRE0 to FRE17 Receive Frame Interrupt Enable Bits 0 to 17

When an interrupt source flag is set (a bit from among the receive interrupt status bits in the receive interrupt status register (RIS0.FRF0 to 17) = 1) while the interrupt is enabled, the interrupt is issued.

chanhvinh.com

45.2.31 Receive Interrupt Status Register 0 (RIS0)

The RIS0 register indicates the states of the AVB-DMAC receive interrupts.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FRF17	FRF16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FRF15	FRF14	FRF13	FRF12	FRF11	FRF10	FRF9	FRF8	FRF7	FRF6	FRF5	FRF4	FRF3	FRF2	FRF1	FRF0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.35 RIS0 register contents (1/2)

Bit Position	Bit Name	Function
b31 to b18	—	Reserved These bits are read as 0. The write value should be 0.
b17	FRF17	Receive Frame Interrupt Status 17 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b16	FRF16	Receive Frame Interrupt Status 16 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b15	FRF15	Receive Frame Interrupt Status 15 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b14	FRF14	Receive Frame Interrupt Status 14 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b13	FRF13	Receive Frame Interrupt Status 13 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b12	FRF12	Receive Frame Interrupt Status 12 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b11	FRF11	Receive Frame Interrupt Status 11 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b10	FRF10	Receive Frame Interrupt Status 11 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b9	FRF9	Receive Frame Interrupt Status 9 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b8	FRF8	Receive Frame Interrupt Status 8 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b7	FRF7	Receive Frame Interrupt Status 7 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b6	FRF6	Receive Frame Interrupt Status 6 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.

chanhvinh.com

Table 45.35 RIS0 register contents (2/2)

Bit Position	Bit Name	Function
b5	FRF5	Receive Frame Interrupt Status 5 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b4	FRF4	Receive Frame Interrupt Status 4 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b3	FRF3	Receive Frame Interrupt Status 3 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b2	FRF2	Receive Frame Interrupt Status 2 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b1	FRF1	Receive Frame Interrupt Status 1 (Network Control) 0: The interrupt is not pending. 1: The interrupt is pending.
b0	FRF0	Receive Frame Interrupt Status 0 (Best Effort) 0: The interrupt is not pending. 1: The interrupt is pending.

FRF0 to FRF17 Receive Frame Interrupt Status Bits 0 to 17

Each bit indicates that a corresponding frame has been stored normally in reception queues 0 to 17 and that data are ready for CPU processing.

Only 0 can be written to the bit.

[Conditions for Changing]

A bit is set to 0 when the operating mode is not operation mode.

A bit is set to 0 when a value is written to the unread frame counter decrement register i (UFCDi) (i = 0 to 4), and this decrements the value of unread frame counter register i (UFCVi) (i = 0 to 4) to 0.

When a frame is stored in reception queue normally and a write transaction to the corresponding descriptor is issued, the corresponding bit is set to 1. Note that the bit is set to 1 without waiting for response to the write transaction. Therefore confirm that the descriptor type (DESCR.DT) of the corresponding descriptor has been updated before processing the received data.

chanhvinh.com

45.2.32 Receive Interrupt Control Register 1 (RIC1)

The RIC1 register controls the AVB-DMAC receive interrupts.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RFWE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.36 RIC1 register contents

Bit Position	Bit Name	Function
b31	RFWE	Receive FIFO Warning Interrupt Enable 0: Disabled 1: Enabled
b30 to b0	—	Reserved These bits are read as 0. The write value should be 0.

RFWE Receive FIFO Warning Interrupt Enable Bit

If the reception FIFO reaches the caution level (the value set in the receive FIFO caution level bits in the receive configuration register (RCR.RFCL) with the corresponding interrupt enabled, the interrupt is issued.

chanhvinh.com

45.2.33 Receive Interrupt Status Register 1 (RIS1)

The RIS1 register indicates the states of the AVB-DMAC receive interrupts.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RFWF	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.37 RIS1 register contents

Bit Position	Bit Name	Function
b31	RFWF	Receive FIFO Warning Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b30 to b0	—	Reserved These bits are read as 0. The write value should be 0.

RFWF Receive FIFO Warning Interrupt Status Bit

This bit indicates that the reception FIFO has reached the set caution level (the value set in the receive FIFO caution level bits in the receive configuration register (RCR.RFCL)).

Only 0 can be written to the bit.

[Conditions for Changing]

The bit is set to 0 when the operating mode is not operation mode.

The bit is set to 1 when the reception FIFO exceeds the set caution level (the value set in the receive FIFO caution level bits in the receive configuration register (RCR.RFCL)).

chanhvinh.com

45.2.34 Receive Interrupt Control Register 2 (RIC2)

The RIC2 register controls the AVB-DMAC receive interrupts.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RFFE	—	—	—	—	—	—	—	—	—	—	—	—	—	QFE17	QFE16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	QFE15	QFE14	QFE13	QFE12	QFE11	QFE10	QFE9	QFE8	QFE7	QFE6	QFE5	QFE4	QFE3	QFE2	QFE1	QFE0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.38 RIC2 register contents (1/2)

Bit Position	Bit Name	Function
b31	RFFE	Receive FIFO Full Interrupt Enable 0: Disabled 1: Enabled
b30 to b18	—	Reserved These bits are read as 0. The write value should be 0.
b17	QFE17	Receive Queue 17 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b16	QFE16	Receive Queue 16 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b15	QFE15	Receive Queue 15 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b14	QFE14	Receive Queue 14 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b13	QFE13	Receive Queue 13 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b12	QFE12	Receive Queue 12 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b11	QFE11	Receive Queue 11 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b10	QFE10	Receive Queue 10 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b9	QFE9	Receive Queue 9 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b8	QFE8	Receive Queue 8 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b7	QFE7	Receive Queue 7 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled

chanhvinh.com

Table 45.38 RIC2 register contents (2/2)

Bit Position	Bit Name	Function
b6	QFE6	Receive Queue 6 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b5	QFE5	Receive Queue 5 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b4	QFE4	Receive Queue 4 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b3	QFE3	Receive Queue 3 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b2	QFE2	Receive Queue 2 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b1	QFE1	Receive Queue 1 (Network Control) Full Interrupt Enable 0: Disabled 1: Enabled
b0	QFE0	Receive Queue 0 (Best Effort) Full Interrupt Enable 0: Disabled 1: Enabled

RFFE Receive FIFO Full Interrupt Enable Bit

When the reception FIFO is full and the interrupt is enabled, the interrupt is issued.

QFE0 to 17 Receive Queue 0 to 17 Full Interrupt Enable Bits

When a reception queue is full and the interrupt is enabled, the interrupt is issued.

chanhvinh.com

45.2.35 Receive Interrupt Status Register 2 (RIS2)

The RIS2 register indicates the states of the AVB-DMAC receive interrupts.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RFFF	—	—	—	—	—	—	—	—	—	—	—	—	—	QFF17	QFF16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	QFF15	QFF14	QFF13	QFF12	QFF11	QFF10	QFF9	QFF8	QFF7	QFF6	QFF5	QFF4	QFF3	QFF2	QFF1	QFF0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.39 RIS2 register contents (1/2)

Bit Position	Bit Name	Function
b31	RFFF	Receive FIFO Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending
b30 to b18	—	Reserved These bits are read as 0. The write value should be 0.
b17	QFF17	Receive Queue 17 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b16	QFF16	Receive Queue 16 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b15	QFF15	Receive Queue 15 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b14	QFF14	Receive Queue 14 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b13	QFF13	Receive Queue 13 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b12	QFF12	Receive Queue 12 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b11	QFF11	Receive Queue 11 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b10	QFF10	Receive Queue 10 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b9	QFF9	Receive Queue 9 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b8	QFF8	Receive Queue 8 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b7	QFF7	Receive Queue 7 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.

chanhvinh.com

Table 45.39 RIS2 register contents (2/2)

Bit Position	Bit Name	Function
b6	QFF6	Receive Queue 6 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b5	QFF5	Receive Queue 5 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b4	QFF4	Receive Queue 4 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b3	QFF3	Receive Queue 3 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b2	QFF2	Receive Queue 2 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b1	QFF1	Receive Queue 7 (Network Control) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b0	QFF0	Receive Queue 0 (Best Effort) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.

RFFF Receive FIFO Full Interrupt Status Bit

This bit indicates that a frame was received but storing it was not possible due to the reception FIFO being full.

When receiving a frame is not possible, the frame will be discarded.

Other information regarding discarded frames is not retained. Even if the frame is not discarded, this bit may also be set to 1 if the E-MAC determines that the frame is an error frame

Only 0 can be written to the bit.

[Conditions for Changing]

The bit is set to 0 when the operating mode is not operation mode.

The bit is set to 1 when the reception FIFO cannot hold received data.

QFF0 to 17 Receive 0 to 17 Full Interrupt Status Bits

These bits indicate that reception queue r did not have space for storing a received frame.

A reception queue is treated as full when it has no descriptors (descriptor type (DESCR.DT) = FEMPTY, FEMPTY_IS, FEMPTY_IC, or FEMPTY_ND) available or reaches the set level for stopping.

CAUTION

If no FEMPTY descriptors or no empty space for descriptors remains in the queue during storing of a divided frame (see section 45.3.4.3 (b) Divided Frames), an error frame is stored in the queue. Such error frames are treated as descriptor sequence errors.

chanhvinh.com

[Conditions for Changing]

A bit is set to 0 when the operating mode is not operation mode.

A bit is set to 1 when reception queue r has no space available for storage.

A bit is set to 1 when the unread frame counter (unread frame counter register i (UFCVi) (i = 0 to 4)) reaches the set level for stopping.

A bit is set to 1 when the unread frame counter reaches the level for stopping before the subsequent received frame is discarded.

chanhvinh.com

45.2.36 Transmit Interrupt Control Register (TIC)

The TIC register controls the AVB-DMAC transmit interrupts.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TFWE	TFUE	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Table 45.40 TIC register contents

Bit Position	Bit Name	Function
b31 to b10	—	Reserved These bits are read as 0. The write value should be 0.
b9	TFWE	Time Stamp FIFO Warning Interrupt Enable 0: Disabled 1: Enabled
b8	TFUE	Time Stamp FIFO Update Interrupt Enable 0: Disabled 1: Enabled
b7 to b0	—	Reserved These bits are read as 0. The write value should be 0.

TFWE Time Stamp FIFO Warning Interrupt Enable Bit

When the time-stamp FIFO reaches the warning level while the interrupt is enabled, the interrupt is issued.

TFUE Time Stamp FIFO Update Interrupt Enable Bit

When the time-stamp FIFO is updated while the interrupt is enabled, the interrupt is issued.

chanhvinh.com

45.2.37 Transmit Interrupt Status Register (TIS)

The TIS register indicates the states of the AVB-DMAC transmit interrupts.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TFWF	TFUF	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Table 45.41 TIS register contents

Bit Position	Bit Name	Function
b31 to b10	—	Reserved These bits are read as 0. The write value should be 0.
b9	TFWF	Time Stamp FIFO Warning Interrupt Status 0: The interrupt is not pending. 1: The time-stamp FIFO has reached the warning level.
b8	TFUF	Time Stamp FIFO Update Interrupt Status 0: The interrupt is not pending. 1: The time-stamp FIFO has been updated.
b7 to b0	—	Reserved These bits are read as 0. The write value should be 0.

TFWF Time Stamp FIFO Warning Interrupt Status Bit

This bit indicates that the transmission time-stamp FIFO has reached the warning level.

Only 0 can be written to the bit.

[Conditions for Changing]

The bit is set to 0 when the operating mode is not operation mode and when the time stamp FIFO enable bit in the transmit configuration control register (TCCR.TFEN) is 0.

The bit is set to 1 after a frame including DESCR.TSR set has been transmitted and one entry has already been stored in the time-stamp FIFO.

chanhvinh.com

TFUF Time Stamp FIFO Update Interrupt Status Bit

This bit indicates that the transmission time-stamp FIFO has been updated.

Only 0 can be written to the bit.

[Conditions for Changing]

The bit is set to 0 when the operating mode is not operation mode, when the time stamp FIFO enable bit in the transmit configuration control register (TCCR.TFEN) is 0, and when 1 is written to the time stamp FIFO release bit in the transmit configuration control register (TCCR.TFR).

The bit is set to 1 when the time stamp FIFO enable bit (TCCR.TFEN) is 1 after a frame including DESCR.TSR set has been transmitted.

chanhvinh.com

45.2.38 Interrupt Summary Status Register (ISS)

The ISS register gives a summary of the states of AVB-DMAC-related interrupts.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DPS15	DPS14	DPS13	DPS12	DPS11	DPS10	DPS9	DPS8	DPS7	DPS6	DPS5	DPS4	DPS3	DPS2	DPS1	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CGIS	RFWS	—	—	TFWS	TFUS	MS	ES	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.42 ISS register contents (1/2)

Bit Position	Bit Name	Function
b31	DPS15	Descriptor Interrupt 15 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b30	DPS14	Descriptor Interrupt 14 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b29	DPS13	Descriptor Interrupt 13 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b28	DPS12	Descriptor Interrupt 12 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b27	DPS11	Descriptor Interrupt 11 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b26	DPS10	Descriptor Interrupt 10 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b25	DPS9	Descriptor Interrupt 9 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b24	DPS8	Descriptor Interrupt 8 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b23	DPS7	Descriptor Interrupt 7 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b22	DPS6	Descriptor Interrupt 6 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b21	DPS5	Descriptor Interrupt 5 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b20	DPS4	Descriptor Interrupt 4 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b19	DPS3	Descriptor Interrupt 3 Summary 0: The interrupt is not pending. 1: The interrupt is pending.

chanhvinh.com

Table 45.42 ISS register contents (2/2)

Bit Position	Bit Name	Function
b18	DPS2	Descriptor Interrupt 2 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b17	DPS1	Descriptor Interrupt 1 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b16 to b14	—	Reserved These bits are read as 0. The write value should be 0.
b13	CGIS	gPTP Interrupt Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b12	RFWS	Receive FIFO Warning Interrupt Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b11 to b10	—	Reserved These bits are read as 0. The write value should be 0.
b9	TFWS	Time Stamp FIFO Warning Interrupt Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b8	TFUS	Time Stamp FIFO Update Interrupt 0: The interrupt is not pending. 1: The interrupt is pending.
b7	MS	E-MAC Interrupt Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b6	ES	Error Interrupt Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b5 to b0	—	Reserved These bits are read as 0. The write value should be 0.

DPS1 to DPS15 Descriptor Interrupt 1 to 15 Summary Bits

These bits are set to 1 when the given descriptor interrupt enable bit (DIC.DPE1 to DPE15) and descriptor interrupt status flag (DIS.DPF1 to DPF15) are both 1.

CGIS gPTP Interrupt Summary Bit

This bit is set to 1 when either interrupt-related bit in the two gPTP-related interrupt registers (GIC and GIS) is 1.

RFWS Receive FIFO Warning Interrupt Summary Bit

This bit is set to 1 when the receive FIFO warning interrupt enable bit (RIC1.RFWE) and receive FIFO warning interrupt status flag (RIS1.RFWF) are both 1.

TFWS Time Stamp FIFO Warning Interrupt Summary Bit

This bit is set to 1 when the time stamp FIFO warning interrupt enable bit (TIC.TFWE) and time stamp FIFO warning interrupt status flag (TIS.TFWF) are both 1.

TFUS Time Stamp FIFO Update Interrupt Summary Bit

This bit is set to 1 when the time stamp FIFO update interrupt enable bit (TIC.TFUE) and time stamp FIFO update interrupt status flag (TIS.TFUF) are both 1.

chanhvinh.com

MS E-MAC Interrupt Summary Bit

This bit is set to 1 when an E-MAC interrupt is issued.

ES Error Interrupt Summary Bit

This bit is set to 1 when any of the valid flags in the error interrupt status register (EIS) is 1 or the queue full error interrupt status bit (EIS.QFS) in the error interrupt status register (EIS) is 1.

chanhvinh.com

45.2.39 gPTP Configuration Control Register (GCCR)

The GCCR register is used to set and control the gPTP (generalized precision time protocol).

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TCSS[1:0]		—	—	LMTT	LPTC	LTI	LTO	TCR[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.43 GCCR register contents

Bit Position	Bit Name	Function
b31 to b10	—	Reserved These bits are read as 0. The write value should be 0.
b9, b8	TCSS[1:0]	Timer Capture Source Select 00: gPTP timer value 01: Adjusted gPTP timer value 10: AVTP presentation time 11: Setting prohibited
b7, b6	—	Reserved These bits are read as 0. The write value should be 0.
b5	LMTT	Maximum Transit Time Configuration Request 0: Setting completed 1(W): Issue a configuration request. 1(R): Completion of settings is pending.
b4	LPTC	Presentation Time Compare Value Configuration Request 0: Setting completed 1(W): Issue a configuration request. 1(R): Completion of settings is pending.
b3	LTI	Timer Increment Value Configuration Request 0: Setting completed 1(W): Issue a configuration request. 1(R): Completion of settings is pending.
b2	LTO	Timer Offset Value Configuration Request 0: Setting completed 1(W): Issue a configuration request. 1(R): Completion of settings is pending.
b1, b0	TCR[1:0]	Timer Control Request 00: Timer control is not requested. 01: gPTP/AVPT presentation time reset 10: Setting prohibited 11: Captures the value set in the TCSS bit.

chanhvinh.com

TCSS[1:0] Timer Capture Source Select Bits

These bits select the source used for updating the captured timer register (gPTP timer capture register (GCTi.CTV)).

These bits should still be controlled when timer control is not being requested (GCCR.TCR = B'00).

LMTT Maximum Transit Time Configuration Request Bit

This bit issues requests for configuring the gPTP maximum transit time configuration register (GMTT).

Only 1 can be written to the bit.

[Conditions for Changing]

The bit is set to 1 when the operating mode is not operation mode.

The bit is set to 0 when the value of the gPTP maximum transit time configuration register (GMTT) is reflected in the gPTP timer.

LPTC Presentation Time Compare Value Configuration Request Bit

This bit issues requests for configuring the gPTP presentation time comparison register (GPTC).

Only 1 can be written to the bit.

[Conditions for Changing]

The bit is set to 1 when the operating mode is not operation mode.

The bit is set to 0 when the value of the gPTP presentation time comparison register (GPTC) is reflected in the gPTP timer.

LTI Timer Increment Value Configuration Request Bit

This bit issues requests for configuring the gPTP timer increment configuration register (GTI).

Only 1 can be written to the bit.

[Conditions for Changing]

The bit is set to 1 when the operating mode is not operation mode.

The bit is set to 0 when the value of the gPTP timer increment configuration register (GTI) is reflected in the gPTP timer.

LTO Timer Offset Value Configuration Request Bit

This bit issues requests for configuring gPTP timer offset configuration register i (GTOi).

Only 1 can be written to the bit.

[Conditions for Changing]

The bit is set to 1 when the operating mode is not operation mode.

The bit is set to 0 when the value of gPTP timer offset configuration register i (GTOi) is reflected in the gPTP timer.

chanhvinh.com

TCR[1:0] Timer Control Request Bits

These bits issue requests for controlling the gPTP timer.

Writing to the bits is only possible when the current operating mode is operation mode and the bits are B’00.

Do not write to the bit when the gPTP timer clock select bit in the AVB-DMAC mode register is B’00.

[Conditions for Changing]

The bits are set to B’00 when the operating mode is not operation mode and on completion of the requested processing.

chanhvinh.com

45.2.40 gPTP Maximum Transit Time Configuration Register (GMTT)

The GMTT register sets the maximum time for transitions of the gPTP timer.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MTTV[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MTTV[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.44 GMTT register contents

Bit Position	Bit Name	Function
b31 to b0	MTTV[31:0]	Maximum Transit Time The maximum transition time for addition to the presentation time

MTTV[31:0] Maximum Transit Time Bits

These bits set the maximum transition time for use in calculating AVTP presentation times.

Write the desired setting to the bits, then issue the configuration request by setting the maximum transit time configuration request bit in the gPTP configuration control register (GCCR.LMTT) to 1.

CAUTION

Do not write a value to these bits when the operating mode is operation mode and the maximum transit time configuration request bit (GCCR.LMTT) is 1.

chanhvinh.com

45.2.41 gPTP Presentation Time Comparison Register (GPTC)

The GPTC register sets a value for comparison with presentation times in the gPTP timer.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTCV[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PTCV[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.45 GPTC register contents

Bit Position	Bit Name	Function
b31 to b0	PTCV[31:0]	Presentation Time Comparison Value Value for comparison with the gPTP presentation times

PTCV[31:0] Presentation Time Comparison Value Bits

These bits set a value for comparison with AVTP timer values to which a maximum transit time is not appended.

Write the desired setting to the bits, then issue the configuration request by setting the presentation time comparison value configuration request bit in the gPTP configuration control register (GCCR.LPTC) to 1.

CAUTION

Do not write a value to these bits when the operating mode is operation mode and the presentation time comparison value configuration request bit (GCCR.LPTC) is 1.

Do not set these bits to a value in the range from x-1 to x+1 (x is the setting value of the gPTP timer increment value bits (GTI.TIV)).

chanhvinh.com

45.2.42 gPTP Timer Increment Configuration Register (GTI)

The GTI register sets the increment for the gPTP timer.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	TIV[27:16]											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TIV[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.46 GTI register contents

Bit Position	Bit Name	Function
b31 to b28	—	Reserved These bits are read as 0. The write value should be 0.
b27 to b0	TIV[27:0]	gPTP Timer Increment Value Increment for the gPTP timer

TIV[27:0] Bits (gPTP Timer Increment Value)

When the gPTP clock select bits in the AVB-DMAV mode register (CCC.CSEL) are selecting a clock signal, these bits set the value by which the timer is incremented each time a cycle of that clock signal elapses.

Write the desired setting to the bits, then issue the configuration request by setting the timer increment value configuration request bit in the gPTP configuration control register (GCCR.LTI) to 1.

CAUTION

Do not write a value to these bits when the operating mode is operation mode and the timer increment value configuration request bit (GCCR.LTI) is 1.

Do not write 0 to the bits.

chanhvinh.com

45.2.43 gPTP Timer Offset Configuration Register i (GTOi) (i = 0 to 2)

The GTOi register sets an offset value for the gPTP timer.

The offset value is added to the combination of bits 0 to 31 in GTO0, 32 to 63 in GTO1, and 64 to 79 in GTO2, which together make up the gPTP timer.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TOV[31+32*i:16+32*i]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TOV[15+32*i:32*i]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.47 GTOi register contents

Bit Position	Bit Name	Function
b31 to b0	TOV[31+32*i:32*i]	Timer Offset Value Offset value for the gPTP timer

TOV[79:0] Timer Offset Value Bits

This is an 80-bit value consisting of the settings in GTO0.TOV[31:0], GTO1.TOV[63:32], and GTO2.TOV[79:64], and is used to set an offset for adding to the value of the gPTP timer.

Write the desired setting to the bits, then issue the configuration request by setting the timer offset value configuration request bit in the gPTP configuration control register (GCCR.LTO) to 1.

CAUTION

Do not write a value to these bits when the operating mode is operation mode and the timer offset value configuration request bit (GCCR.LTO) is 1.

Write H'0000 to GTO2.TOV[95:80].

Set a value in the range from 0 to 10^9-1 (H'0000 0000 to H'3B9A C9FF) in GTO0.TOV[31:0].

chanhvinh.com

45.2.44 gPTP Interrupt Control Register (GIC)

The GCI register is used to control gPTP-related interrupts.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	PTME	PTOE	PTCE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 45.48 GIC register contents

Bit Position	Bit Name	Function
b31 to b3	—	Reserved These bits are read as 0. The write value should be 0.
b2	PTME	Presentation Time Match Interrupt Enable 0: Disabled 1: Enabled
b1	PTOE	Presentation Time Overrun Interrupt Enable 0: Disabled 1: Enabled
b0	PTCE	Presentation Time Capture Interrupt Enable 0: Disabled 1: Enabled

PTME Presentation Time Match Interrupt Enable Bit

When this bit is 1, setting of the presentation time match interrupt flag in the gPTP interrupt status register (GIS.PTMF) to 1 leads to generation of that interrupt.

PTOE Presentation Time Overrun Interrupt Enable Bit

When this bit is 1, setting of the presentation time overrun interrupt flag in the gPTP interrupt status register (GIS.PTOF) to 1 leads to generation of that interrupt.

PTCE Presentation Time Capture Interrupt Enable Bit

When this bit is 1, setting of the presentation time capture interrupt flag in the gPTP interrupt status register (GIS.PTCF) to 1 leads to generation of that interrupt.

chanhvinh.com

45.2.45 gPTP Interrupt Status Register (GIS)

The GIC register indicates the state of the gPTP-related interrupt.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	PTMF	PTOF	PTCF
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 45.49 GIS register contents

Bit Position	Bit Name	Function
b31 to b3	—	Reserved These bits are read as 0. The write value should be 0.
b2	PTMF	Presentation Time Match Interrupt Flag 0: The interrupt is not pending. 1: The interrupt is pending.
b1	PTOF	Presentation Time Overrun Interrupt Flag 0: The interrupt is not pending. 1: The interrupt is pending.
b0	PTCF	Presentation Time Capture Interrupt Flag 0: The interrupt is not pending. 1: The interrupt is pending.

PTMF Presentation Time Match Interrupt Flag Bit

This bit indicates that the value of the AVTP timer exceeds the value of the gPTP presentation time comparison register (GPTC).

Only 0 can be written to the bit.

[Conditions for Changing]

The bit is set to 0 when the operating mode is not operation mode.

The bit is set to 1 when the AVTP timer value is greater than or equal to the value of the gPTP presentation time comparison register (GPTC).

PTOF Presentation Time Overrun Interrupt Flag Bit

This bit indicates that a capture event is generated before completion of the previous capture process.

Only 0 can be written to the bit.

[Conditions for Changing]

The bit is set to 0 when the operating mode is not operation mode.

The bit is set to 0 when the presentation time capture interrupt flag bit (GIS.PTCF) is set to 0. The bit is set to 1 when a capture event is generated while the value of the presentation time capture interrupt flag bit (GIS.PTCF) is 1.

chanhvinh.com

PTCF Presentation Time Capture Interrupt Flag Bit

This bit indicates that a capture event is generated.

Only 0 can be written to the bit.

[Conditions for Changing]

The bit is set to 0 when the operating mode is not operation mode.

The bit is set to 1 when a capture event is generated.

chanhvinh.com

45.2.46 gPTP Presentation Time Capture Register (GCPT)

The GCPT register indicates the captured presentation time.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CPTV[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CPTV[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.50 GCPT register contents

Bit Position	Bit Name	Function
b31 to b0	CPTV[31:0]	Presentation Time Capture Presentation time value

CPTV[31:0] Presentation Time Capture Bits

These bits indicate the AVTP presentation time captured in response to a capture event. Read the time value from these bits and clear the presentation time capture interrupt flag bit (GIS.PTCF) before triggering a next capture.

Whether capturing a correct value was possible can be confirmed by following the procedure below.

1. Read these bits.
2. Read the presentation time overrun interrupt flag bit (GIS.PTOF).
3. Read these bits again.
4. Check whether the presentation time overrun interrupt flag bit (GIS.PTOF) is set to 1 and whether the value read from these bits has changed. If either of the two conditions is met, a correct value has not been captured.

   [Conditions for Changing]

   The bits are set to H’00000000 when the operating mode is not operation mode.

   When a capture event occurs, the bits are updated with the AVTP presentation time value at the time of the event.

chanhvinh.com

45.2.47 gPTP Timer Capture Register i (GCTi) (i = 0 to 2)

The GCTi registers form an 80-bit register that captures the gPTP timer value.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CTV[31+32*i:16+32*i]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CTV[15+32*i:32*i]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.51 GCTi register contents

Bit Position	Bit Name	Function
b31 to b0	CTV[31+32*i:32*i]	gPTP Timer Capture Value Captured timer value

CTV[79:0] gPTP Timer Capture Value Bits

These 80 bits consist of GCT0.CTV[31:0], GCT1.CTV[63:32] and GCT2.CTV[79:64], which together indicate captured timer values.

When B’00 (value of the gPTP timer) or B'01 (corrected gPTP timer value) is selected by the timer capture source select bits in the gPTP configuration control register (GCCR.TCSS), the corresponding 80-bit values are stored in GCT0.CTV[31:0], GCT1.CTV[63:32], and GCT2.CTV[79:64].

When B’10 (AVTP presentation time) is selected by the timer capture source select bit (GCCR.TCSS), the corresponding 32-bit values are stored in GCT0.CTV[31:0].

Actual writing of the timer value specified by the timer capture source select bits (GCCR.TCSS) proceeds when B’11 (timer capture request) is written to the timer control request bits in the gPTP configuration control register (GCCR.TCR).

Do not read the value while the value of the timer control request bits (GCCR.TCR) is B’11, because this indicates that storage is still in progress.

When operating mode is not operation mode, the bits are set to H’00000000.

chanhvinh.com

45.2.48 gPTP Capture Event Control Register (GCEC)

The GCEC register sets the capture event.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	DRC[10:0]										
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SCS[1:0]		—	—	—	DEN	—	—	—	CES
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R	R	R	R/W

Table 45.52 GCEC register contents

Bit Position	Bit Name	Function
b31 to b27	—	Reserved These bits are read as 0. The write value should be 0.
b26 to b16	DRC[10:0]	Division Ratio Setting
b15 to b10	—	Reserved These bits are read as 0. The write value should be 0.
b9, b8	SCS[1:0]	Serial Sound Interface Channel Select 00: Serial sound interface channel 0 01: Serial sound interface channel 1 10: Serial sound interface channel 2 11: Serial sound interface channel 3
b7 to b5	—	Reserved These bits are read as 0. The write value should be 0.
b4	DEN	Divider Operation Enable 0: Divider operation is disabled. 1: Divider operation is enabled.
b3 to b1	—	Reserved These bits are read as 0. The write value should be 0.
b0	CES	Capture Event Select 0: External capture event (AVB_CAPTURE pin) 1: Output from the divider

CAUTION

Set the bits following the procedure described below. For pin control, see section 41., Ports.

<1> Change the capture event from an external capture event (AVB_CAPTURE pin) to output from the divider.

1. Perform pin control to disable the AVB_CAPTURE pin function.
2. Make the setting for the serial sound interface.
3. Set GCEC.CES to 1.
4. Set a desired value in GCEC.SCS and GCES.DRC.
5. Set GCEC.DEN to 1.

chanhvinh.com

<2> Change the capture event from output from the divider to the external capture event (AVB_CAPTURE pin).

1. Set GCEC.DEN to 0.
2. Wait for 2 cycles of output from the divider specified by GCEC.SCS and GCES.DRC.
3. Set GCEC.CES to 0.
4. Perform pin control to enable the AVB_CAPTURE pin function.

<3> Switch SCS or DRC while output from the divider is selected as the capture event.

1. Set GCEC.DEN to 0.
2. Wait for 2 cycles of output from the divider specified by GCEC.SCS and GCES.DRC.
3. Set a desired value in GCEC.SCS and GCES.DRC.
4. Set GCEC.DEN to 1.

DRC[10:0] Division Ratio Setting Bits

These bits set the division ratio which is used in dividing the SSIWS signal supplied from the serial sound interface by the divider.

The division ratio is “the value set in these bits + 1”.

SCS[1:0] Serial Sound Interface Channel Select Bits

These bits select the channel of the serial sound interface which supplies the SSIWS signal to the divider.

CAUTION

The frequency of the SSIWS signal of the selected channel must be equal to or smaller than a quarter of the clock frequency set by the gPTP clock select bits (CCC.CSEL).

DEN Divider Operation Enable Bit

This bit sets the operating state of the divider.

Change the settings of GCEC.DRC, GCEC.SCS, GCEC.CES, and serial sound interface after setting this bit to 0.

Do not set this bit to 1 while GCEC.CES is set to 0.

CES Capture Event Select Bit

This bit sets the supply source of the capture event.

The capture event is detected on the rising edge of the signal selected by this bit.

CAUTION

When the external capture event (AVB_CAPTURE pin) is selected, the AVB_CAPTURE pin should be held high for at least two cycles of the clock specified by the gPTP clock select bits (CCC.CSEL) to generate an external capture event.

When output from the divider is selected, perform pin control to disable the AVB_CAPTURE

chanhvinh.com

pin function. For pin control, see section 41., Ports.

chanhvinh.com

45.2.49 E-MAC Mode Register (ECMR)

ECMR is used to specify the operating mode of the E-MAC. The settings in this register are normally made in the initialization process following a reset.

The operating mode settings must not be changed while transmission or reception is enabled (i.e. while the RE or TE bit in this register is 1).

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	TRCCM	—	—	RCSC	—	DPAD	RZPF	ZPF	PFR	RXF	TXF
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	RE	TE	—	—	—	DM	PRM
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R/W

Table 45.53 ECMR register contents (1/2)

Bit Position	Bit Name	Function
b31 to b27	—	Reserved These bits are read as 0. The write value should be 0.
b26	TRCCM	Counter Clear Mode 0: Writing to a counter register leads to the register being cleared to 0. 1: Reading from a counter register leads to the register being cleared to 0.
b25, b24	—	Reserved These bits are read as 0. The write value should be 0.
b23	RCSC	Checksum Calculation 0: Checksums are not automatically calculated. 1: Checksums are automatically calculated.
b22	—	Reserved This bit is read as 0. The write value should be 0.
b21	DPAD	Data Padding 0: Padding to make up 60 bytes is inserted in data for transmission when fewer than 60 bytes are to be transmitted. 1: Padding is not inserted in data for transmission when fewer than 60 bytes are to be transmitted and the data are transmitted without being changed.
b20	RZPF	PAUSE Frame Reception with Time = 0 0: Reception of PAUSE frames with the TIME parameter value 0 is disabled. 1: Reception of PAUSE frames with the TIME parameter value 0 is enabled.
b19	ZPF	PAUSE Frame Usage with TIME = 0 Enable 0: Control in response to and for the sending of PAUSE frames with the TIME parameter value 0 is disabled. 1: Control in response to and for the sending of PAUSE frames with the TIME parameter value is 0 is enabled.
b18	PFR	PAUSE Frame Receive Mode 0: PAUSE frames are not transferred to the AVB-DMAC. 1: PAUSE frames are transferred to the AVB-DMAC.
b17	RXF	Operating Mode for Flow Control in Reception 0: Detection of PAUSE frames is disabled. 1: Flow control for the receiving port is enabled.

chanhvinh.com

Table 45.53 ECMR register contents (2/2)

Bit Position	Bit Name	Function
b16	TXF	Operating Mode for Flow Control in Transmission 0: Flow control for the transmitting port is disabled (PAUSE frames are not automatically transmitted). 1: Flow control for the transmitting port is enabled (PAUSE frames are automatically transmitted as required).
b15 to b7	—	Reserved These bits are read as 0. The write value should be 0.
b6	RE	Reception Enable 0: Reception is disabled. 1: Reception is enabled.
b5	TE	Transmission Enable 0: Transmission is disabled. 1: Transmission is enabled.
b4 to b2	—	Reserved These bits are read as 0. The write value should be 0.
b1	DM	Full-duplex Transfer Enable 0: Disables full-duplex transfer. 1: Enables full-duplex transfer.
b0	PRM	Promiscuous Mode 0: Normal operation 1: Promiscuous mode operation

TRCCM Counter Clear Mode Bit

This bit sets the method for clearing the counter register. Refer to the descriptions of the counter registers.

RCSC Checksum Calculation Bit

Setting this bit to 1 enables automatic calculation of checksums for data in received frames.

Only the data field of an Ethernet frame is in the scope of checksum calculation. Specifically, the checksum is calculated from the data field, which follows the length/type field and is followed by the CRC field. Calculation only involves 16-bit addition; it does not involve bit inversion.

DPAD Data Padding Control Bit

This bit specifies padding or non-padding of data when less than 60 bytes are to be transmitted.

When this bit is set to 1, data are transmitted without padding; when it is set to 0, data are padded to make up 60-byte units for transmission.

chanhvinh.com

RZPF (PAUSE Frame Reception with Time = 0) Bit

When the RZPF bit is set to 0, received PAUSE frames with the Timer value 0 are discarded.

When the RZPF bit is set to 1, release from the transmission wait state follows reception of a PAUSE frame with the Timer value 0.

ZPF (PAUSE Frame Usage with TIME = 0 Enable) Bit

When the ZPF bit is set to 0, the next frame to be transmitted is not transmitted until the time specified by the Timer value has elapsed.

Received PAUSE frames with the Timer value 0 are discarded.

When the ZPF bit is set to 1, if the amount of data in the reception FIFO becomes less than the setting of the receive FIFO critical level bits (RCR.RFCL) before the time specified by the Timer value elapses, a PAUSE frame with a Timer value of 0 is automatically transmitted. If the interface is in the transmission wait state, it is released from that state on receiving a PAUSE frame with a Timer value of 0.

PFR PAUSE Frame Receive Mode Bit

This bit specifies whether PAUSE frames are transferred to the AVB-DMAC.

RXF (Operating Mode for Flow Control in Reception) Bit

When the RXF bit is set to 1 and a PAUSE frame is received, a next frame to be transmitted is not transmitted until the time indicated by the Timer value in the PAUSE frame has elapsed. However, the transmission of a current frame is continued. The number of received PAUSE frames is also counted. For details, see section 45.2.57, PAUSE Frame Receive Counter (PFRCR).

Setting this bit to 0 disables PAUSE frame detection.

TXF (Operating Mode for Flow Control in Transmission) Bit

The TXF bit enables or disables flow control in transmission by the port.

Setting this bit to 0 disables PAUSE frame detection.

RE Reception Enable Bit

If this bit is switched from reception being enabled (RE = 1) to reception being disabled (RE = 0) while a frame is being received, reception will continue until reception of that frame is completed.

TE Transmission Enable Bit

If this bit is switched from transmission being enabled (TE = 1) to transmission being disabled (TE = 0) while a frame is being transmitted, transmission will continue until transmission of that frame is completed.

DM Full-Duplex Transfer Enable Bit

This bit enables or disables full-duplex transfer.

PRM Promiscuous Mode Bit

Setting the PRM bit enables all Ethernet frames to be received. All Ethernet frames means all receivable frames, irrespective of differences or enabled/disabled status (destination address, broadcast address, multicast bit, etc.).

chanhvinh.com

45.2.50 Receive Frame Length Register (RFLR)

The RFLR register specifies the maximum length (in bytes) of frames that can be received by this LSI. Settings in this register must not be changed while reception is enabled (while the RE bit in the E-MAC mode register (EMCR) is 1).

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFL[17:16]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RFL[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.54 RFLR register contents

Bit Position	Bit Name	Function
b31 to b18	—	Reserved These bits are read as 0. The write value should be 0.
b17 to b0	RFL[17:0]	Receive Frame Length H'00000 to H'005EE: 1,518 bytes H'005EF: 1,519 bytes H'005F0: 1,520 bytes : : H'007FF: 2,047 bytes H'00800: 2,048 bytes : : H'01000: 4,096 bytes : : H'10000: 65,535 bytes : : H'20000 to H'3FFFF: 131,072 bytes

RFL[17:0] Receive Frame Length Bits

Frame data described here refers to all fields from the destination address up to the CRC data. Frame contents from the destination address up to the data are actually transferred to memory. CRC data are not included in the transfer. When more data than the specified number of bytes are received, the portion of data that exceeds the specified value is discarded.

chanhvinh.com

45.2.51 E-MAC Status Register (ECSR)

The ECSR register indicates the state of the E-MAC. The CPU can be notified of the state. For bits associated with interrupts, the interrupt can be enabled or disabled by the corresponding bit in the E-MAC Interrupt Permission Register (ECSIPR) described in section 45.2.52, E-MAC Interrupt Permission Register (ECSIPR).

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	PFROI	—	—	—	ICD
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Table 45.55 ECSR register contents

Bit Position	Bit Name	Function
b31 to b5	—	Reserved These bits are read as 0. The write value should be 0.
b4	PFROI	PAUSE Frame Retransmit Retry Over 0: PAUSE frame retransmit retry count does not exceed the upper limit. 1: PAUSE frame retransmit retry count exceeds the upper limit.
b3 to b1	—	Reserved These bits are read as 0. The write value should be 0.
b0	ICD	Illegal Carrier Detection 0: PHY-LSI has not detected an illegal carrier on the line. 1: PHY-LSI has detected an illegal carrier on the line.

PFROI PAUSE Frame Retransmit Retry Over Bit

This bit indicates that retransmission count has exceeded the retransmit upper limit set in the PAUSE frame retransmission count register (TPAUSER) during PAUSE frame retransmission when the flow control is enabled.

Writing 1 to this bit clears it to 0.

ICD Illegal Carrier Detection Bit

This bit indicates that the PHY-LSI has detected an illegal carrier on the line. If a change in the signal input from the PHY-LSI occurs in a period shorter than the software recognition period, the correct information may not be obtained. Refer to the timing specification for the PHY-LSI used.

Writing 1 to this bit clears it to 0.

chanhvinh.com

45.2.52 E-MAC Interrupt Permission Register (ECSIPR)

The ECSIPR register enables or disables the states indicated by the ECSR register as interrupt sources. Each effective bit disables or enables interrupts corresponding to the bits in ECSR.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	PFROIP	—	—	—	—	ICDIP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R/W

Table 45.56 ECSIPR register contents

Bit Position	Bit Name	Function
b31 to b5	—	Reserved These bits are read as 0. The write value should be 0.
b4	PFROIP	PAUSE Frame Retransmit Interrupt Enable 0: Interrupts on setting of the PFROI bit is disabled. 1: Interrupts on setting of the PFROI bit is enabled.
b3 to b1	—	Reserved These bits are read as 0. The write value should be 0.
b0	ICDIP	Illegal Carrier Detect Interrupt Enable 0: Interrupts on setting of the ICD bit is disabled. 1: Interrupt on setting of the ICD bit is enabled.

PFROIP PAUSE Frame Retransmit Interrupt Enable Bit

Setting this bit to 1 selects interrupt generation on setting of the PAUSE frame retransmit retry over bit (ECSR.PFROI) in the E-MAC status register to 1.

ICDIP Illegal Carrier Detect Interrupt Enable Bit

Setting this bit to 1 selects interrupt generation on setting of the illegal carrier detection bit (ECSR.ICD) in the E-MAC status register to 1.

chanhvinh.com

45.2.53 PHY Interface Register (PIR)

The PIR register provides a means of access to the PHY-LSI internal registers via the MII.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	MDI	MDO	MMD	MDC
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	—	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 45.57 PIR register contents

Bit Position	Bit Name	Function
b31 to b4	—	Reserved These bits are read as 0. The write value should be 0.
b3	MDI	MII Management Data-In Indicates the level of the ET_MDIO pin.
b2	MDO	MII Management Data-Out Holds data for output from the ET_MDIO pin.
b1	MMD	MII Management Mode 0: Read direction is specified. 1: Write direction is specified.
b0	MDC	MII Management Data Clock The value set in this bit is output from the ET_MDC pin, which supplies the management data clock for the MII.

MDI MII Management Data-In Bit

This bit indicates the level of the ET_MDIO pin.

MDO MII Management Data-Out Bit

This bit holds data for output from the ET_MDIO pin.

The ET_MDIO pin outputs a value set in this bit when the MMD bit is set to 1 (to specify writing as the direction). Data are not output while the MMD bit is set to 0 (to specify reading as the direction).

MMD MII Management Mode Bit

This bit specifies the direction for data through MDIO (reading or writing).

MDC MII Management Data Clock Bit

Values set in this bit are output on the ET_MDC pin to supply the MII with the management data clock. For the method of access to the MII registers, see section 45.3.12, Connection to PHY-LSI.

chanhvinh.com

45.2.54 Automatic PAUSE Frame Register (APR)

The APR register is used to set the value for the TIME parameter of automatically transmitted PAUSE frames. When a PAUSE frame is automatically transmitted, the value set in this register is used as its TIME parameter.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AP[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.58 APR register contents

Bit Position	Bit Name	Function
b31 to b16	—	Reserved These bits are read as 0. The write value should be 0.
b15 to b0	AP[15:0]	Automatic PAUSE These bits set the TIME parameter value of an automatic PAUSE frame.*1 H'0000: — H'0001: 512 × 1 bit-period H'0002: 512 × 2 bit-period : : H'FFFF: 512 × 65535 bit-period Note 1. A bit-period changes relative to the transfer speed as follows. 100Mbps: 1 bit-period = 10 ns 10Mbps: 1 bit-period = 100 ns

AP[15:0] Automatic PAUSE Bit

These bits set the value of the TIME parameter in automatically transmitted PAUSE frames. The unit for the setting is 512-bit periods.

Set these bits to a value other than H'0000 when the flow control in transmission (PAUSE frame transmission) is enabled (ECMR.TXF = 1).

chanhvinh.com

45.2.55 Manual PAUSE Frame Register (MPR)

The MPR register is used to set the value for the TIME parameter of manually generated PAUSE frames. When a PAUSE frame is manually transmitted, the value set in this register is used as its TIME parameter.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MP[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.59 MPR register contents

Bit Position	Bit Name	Function
b31 to b16	—	Reserved These bits are read as 0. The write value should be 0.
b15 to b0	MP[15:0]	Manual PAUSE These bits set the TIME parameter value of a manual PAUSE frame.*1 H'0000: — H'0001: 512 × 1 bit-period H'0002: 512 × 2 bit-period : : H'FFFF: 512 × 65535 bit-period Note 1. A bit-period changes relative to the transfer speed as follows. 100Mbps: 1 bit-period = 10 ns 10Mbps: 1 bit-period = 100 ns

MP[15:0] Manual PAUSE Bits

These bits set the value of the TIME parameter in manually generated PAUSE frames.

The unit for the setting is 512 bit periods.

chanhvinh.com

45.2.56 PAUSE Frame Transmit Counter (PFTCR)

The PFTCR register is a counter that indicates the number of times PAUSE frames have been transmitted.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PFTXC[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.60 PFTCR register contents

Bit Position	Bit Name	Function
b31 to b16	—	Reserved These bits are read as 0.
b15 to b0	PFTXC[15:0]	PAUSE Frame Transmit Counter Counter for counting the number of transmitted PAUSE frames

PFTXC[15:0] PAUSE Frame Transmit Counter Bits

These bits indicate the total number of PAUSE frames that have been transmitted (both manually and automatically).

The bits are cleared to 0 when they are read.

If counting up and clearing of the counter coincide, clearing the counter takes priority.

chanhvinh.com

45.2.57 PAUSE Frame Receive Counter (PFRCR)

The RFRCR register is a counter that indicates the number of times PAUSE frames have been received.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PFRXC[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.61 RFRCR register contents

Bit Position	Bit Name	Function
b31 to b16	—	Reserved These bits are read as 0.
b15 to b0	PFRXC[15:0]	PAUSE Frame Receive Counter Counter for counting the number of received PAUSE frames

PFRXC[15:0] PAUSE Frame Receive Counter Bits

These bits indicate the number of PAUSE frames that have been received when flow control in reception is enabled (the RXF bit in ECMR = 1).

The bits are cleared to 0 when they are read.

If counting up and clearing the counter coincide, clearing the counter takes priority.

chanhvinh.com

45.2.58 Automatic PAUSE Frame Retransmission Count Register (TPAUSER)

The TPAUSER register sets the upper limit of automatic PAUSE frame retransmission count.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TPAUSE[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.62 TPAUSER register contents

Bit Position	Bit Name	Function
b31 to b16	—	Reserved These bits are read as 0.
b15 to b0	TPAUSE[15:0]	Automatic PAUSE Frame Retransmission Count Upper Limit H'0000: No limitation H'0001: Once : : H'FFFF: 65535 times

TPAUSE[15:0] Automatic PAUSE Frame Retransmission Count Upper Limit Bits

The bits set the upper limit of automatic PAUSE frame retransmission count.

The setting in the bits must not be changed while transmission is enabled (EMCR.TE = 1).

chanhvinh.com

45.2.59 MAC Address High Register (MAHR)

The MAHR register specifies the 32 higher-order bits of the 48-bit MAC address. The settings in this register are normally made in the initialization process after a reset.

The settings in this register must not be changed while transmission or reception is enabled.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MA[47:32]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MA[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.63 MAHR register contents

Bit Position	Bit Name	Function
b31 to b0	MA[47:16]	MAC Address Bits 47 to 16 These bits are used to set the 32 higher-order bits of the MAC address.

MA[47:16] MAC Address Bits 47 to 16

These bits are used to set the 32 higher-order bits of the MAC address.

For example, if the MAC address is 01-23-45-67-89-AB (hexadecimal), set H'0123 4567 in the MAHR register.

chanhvinh.com

45.2.60 MAC Address Low Register (MALR)

The MALR register specifies the 16 lower-order bits of the 48-bit MAC address. The settings in this register are normally made in the initialization process after a reset.

The settings in this register must not be changed while transmission or reception is enabled.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MA[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.64 MALR register contents

Bit Position	Bit Name	Function
b31 to b16	—	Reserved These bits are read as 0. The write value should be 0.
b15 to b0	MA[15:0]	MAC Address Bits 15 to 0 These bits are used to set the 16 lower-order bits of the MAC address.

MA[15:0] MAC Address Bits 15 to 0

These bits are used to set the 16 lower-order bits of the MAC address.

For example, if the MAC address is 01-23-45-67-89-AB (hexadecimal), set H'89AB in the MALR register.

chanhvinh.com

45.2.61 CRC Error Frame Receive Counter Register (CEFCR)

The CEFCR register is a counter that indicates the number of times frames with CRC errors were received. Counting up stops when the value in this register reaches H'FFFF FFFF.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CEFC[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CEFC[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.65 CEFCR register contents

Bit Position	Bit Name	Function
b31 to b0	CEFC[31:0]	CRC Error Frame Counter These bits indicate the number of CRC error frames received.

CEFC[31:0] CRC Error Frame Counter Bits

These bits indicate the number of received frames having CRC errors.

The bits are cleared to 0 when they are read while the counter clear mode bit (TRCCM) in the E-MAC mode register is set to 1.

When TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

chanhvinh.com

45.2.62 Frame Receive Error Counter Register (FRECR)

The FRECR register is a counter that indicates the number of frames for which receive errors were generated by input on the ET_RXER pin from the PHY-LSI. Counting up stops when the value in this register reaches H'FFFF FFFF.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	FREC[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FREC[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.66 FRECR register contents

Bit Position	Bit Name	Function
b31 to 0	FREC[31:0]	Frame Receive Error Counter These bits indicate the number of errors during frame reception.

FREC[31:0] Frame Receive Error Counter Bits

These bits indicate the number of errors during frame reception.

The bits are cleared to 0 when they are read while the counter clear mode bit (TRCCM) in the E-MAC mode register is set to 1.

When TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

chanhvinh.com

45.2.63 Too-Short Frame Receive Counter Register (TSFRCR)

The TSFRCR register is a counter that indicates the number of received frames that were fewer than 64 bytes in length. Counting stops when the value in this register reaches H'FFFF FFFF.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TSFRC[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSFRC[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.67 TSFRCR register contents

Bit Position	Bit Name	Function
b31 to b0	TSFRC[31:0]	Too-Short Frame Receive Counter These bits indicate the number of frames received with a length of less than 64 bytes.

TSFRC[31:0] Too-Short Frame Receive Counter Bits

These bits indicate the number of received frames that were fewer than 64 bytes in length.

The bits are cleared to 0 when they are read while the counter clear mode bit (TRCCM) in the E-MAC mode register is set to 1.

When TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

chanhvinh.com

45.2.64 Too-Long Frame Receive Counter Register (TLFRCR)

The TLFRCR register is a counter that indicates the number of received frames that were longer than the value specified in the receive frame length register (RFLR). Counting up stops when the value in the TLFRCR register reaches H'FFFF FFFF.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TLFC[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TLFC[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.68 TLFRCR register contents

Bit Position	Bit Name	Function
b31 to b0	TLFC[31:0]	Too-Long Frame Receive Counter These bits indicate the number of frames received with a length exceeding the value in RFLR.

TLFC[31:0] Too-Long Frame Receive Counter Bits

These bits indicate the number of received frames that were longer than the value in RFLR.

The bits are cleared to 0 when they are read while the counter clear mode bit (TRCCM) in the E-MAC mode register is set to 1.

When TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

chanhvinh.com

45.2.65 Residual-Bit Frame Receive Counter Register (RFCR)

The RFCR register is a counter that indicates the number of received frames containing “residual bits” (trailing bits not making up an 8-bit unit). Counting up stops when the value in this register reaches H’FFFF FFFF.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RFC[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RFC[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.69 RFCR register contents

Bit Position	Bit Name	Function
b31 to b0	RFC[31:0]	Residual-Bit Frame Receive Counter These bits indicate the number of received frames containing residual bits.

RFC[31:0] Residual-Bit Frame Receive Counter Bits

These bits indicate the number of received frames containing residual bits.

The bits are cleared to 0 when they are read while the counter clear mode bit (TRCCM) in the E-MAC mode register is set to 1.

When TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

chanhvinh.com

45.2.66 Multicast Address Frame Receive Counter Register (MAFCR)

The MAFCR register is a counter that indicates the number of received frames for which a multicast address was specified. Counting up stops when the value in this register reaches H'FFFF FFFF.

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MAFC[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MAFC[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.70 MAFCR register contents

Bit Position	Bit Name	Function
b31 to b0	MAFC[31:0]	Multicast Address Frame Counter These bits indicate the number of multicast frames that have been received.

MAFC[31:0] Multicast Address Frame Counter Bits

These bits indicate the number of multicast frames that have been received.

The bits are cleared to 0 when they are read while the counter clear mode bit (TRCCM) in the E-MAC mode register is set to 1.

When TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

chanhvinh.com

45.3 Operation

The EthernetAVB consists of the following functional units:

- DMA transfer controller (AVB-DMAC): Handles DMA transfer between the data storage areas for reception and transmission in the on-chip RAM and the reception and transmission FIFO buffers
- MAC controller (E-MAC): Handles transfer between the reception and transmission FIFO buffers and the MII

Using its direct memory access (DMA) function, the AVB-DMAC handles DMA transfer of frame data between the destinations for storing Ethernet frame data for transmission and reception in the on-chip RAM and the FIFO buffers for reception and transmission. Data cannot be directly read from or written to the FIFO buffers.

To handle DMA transfer, the AVB-DMAC requires information that includes the addresses for storage of data for transmission and received data. These data are referred to as descriptors. The AVB-DMAC reads data for transmission from the storage area for data to be transmitted according to the information in descriptors and writes received data to the storage area for received data accompanied by information in descriptors. The descriptors are placed in the on-chip RAM. Arranging multiple descriptors in descriptor lists allows the continuous reception or transmission of multiple Ethernet frames.

The E-MAC supports an MII, which provides an interface format for the externally connected PHY-LSI. The E-MAC constructs Ethernet frames from data written to the transmission FIFO and transmits these frames to the MII. It also performs CRC checking of Ethernet frames received from the MII and writes the frames to the reception FIFO.

chanhvinh.com

45.3.1 AVB-DMAC Operating Modes

Figure 45.5 illustrates the operating modes of the AVB-DMAC.

Transitions of AVB-DMAC operating mode are under the control of the items listed below.

- CPU operating mode (power-on reset)
- Configuration of the operating mode configuration bits (CCC.OPC) in the AVB-DMAC mode register

The current operating mode can be determined by reading the operating mode status bits in the AVB-DMAC status register (CSR.OPS).

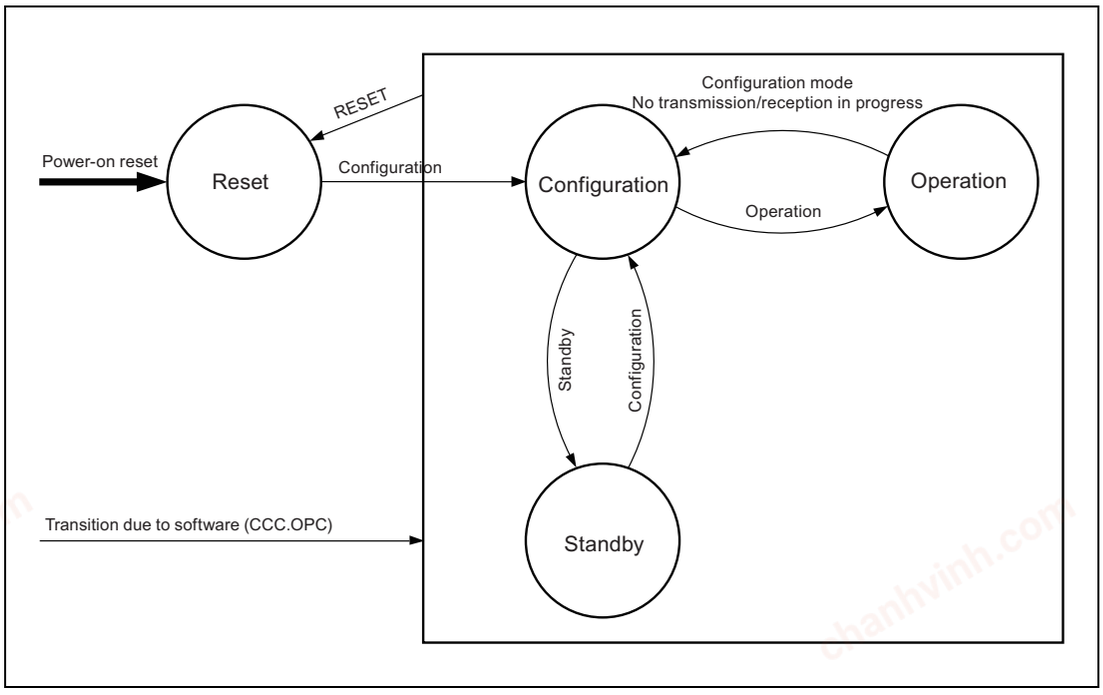


Figure 45.5 Operating Mode of AVB-DMAC

chanhvinh.com

45.3.1.1 Operating Modes

(1) Reset mode

After a power-on reset, the AVB-DMAC enters reset mode.
In reset mode, only the AVB-DMAC operating mode control function is controllable and other functions are all stopped. This mode is designed for reduced power when the Ethernet function is not necessary.

(2) Configuration mode

In configuration mode, various settings for the AVB-DMAC can be made.
The operation of most functions is stopped and all status registers are initialized to their reset values.
The E-MAC functions in this mode.

(3) Operation mode

In operation mode, all functions of the AVB-DMAC can operate.
Ethernet communications can only proceed in this mode.

(4) Standby mode

In standby mode, only the operating mode control function and E-MAC can be used. Other functions cannot be used.

chanhvinh.com

45.3.1.2 How to Set the Operating Mode

Set the operating mode configuration bits in the AVB-DMAC status register (CSR.OPC) to select the operating mode. Furthermore, the current operating mode can be checked by reading the operating mode status bits in the AVB-DMAC status register (CSR.OPS).

Transitions other than from operation mode to configuration mode are made after the value is written to the operating mode configuration bits (CCC.OPC) (Figure 45.6).

For transitions from operation mode to configuration mode, follow the procedure in Figure 45.7 because any transmission and reception in progress will be executed before the transition to configuration mode.

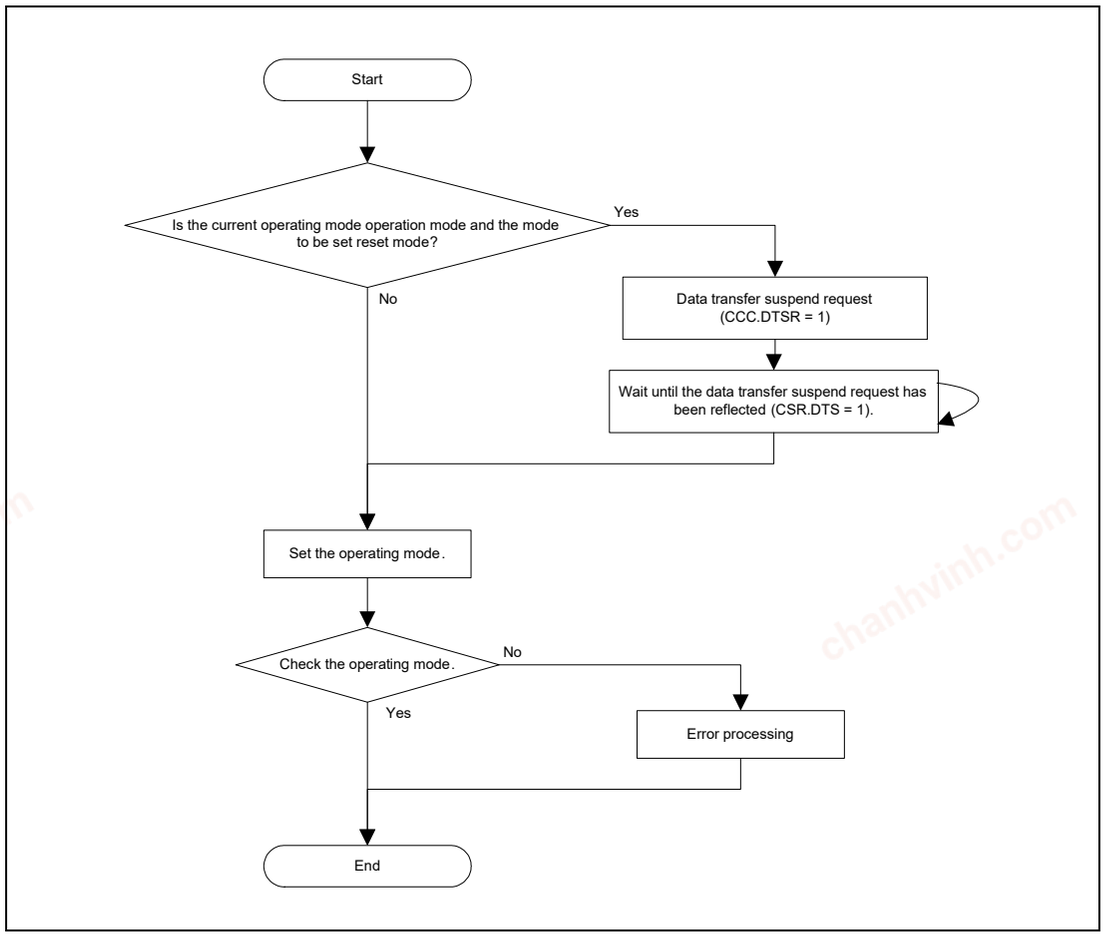


Figure 45.6 Flow for Transitions of Operating Mode (Other than from Operation Mode to Configuration Mode)

chanhvinh.com

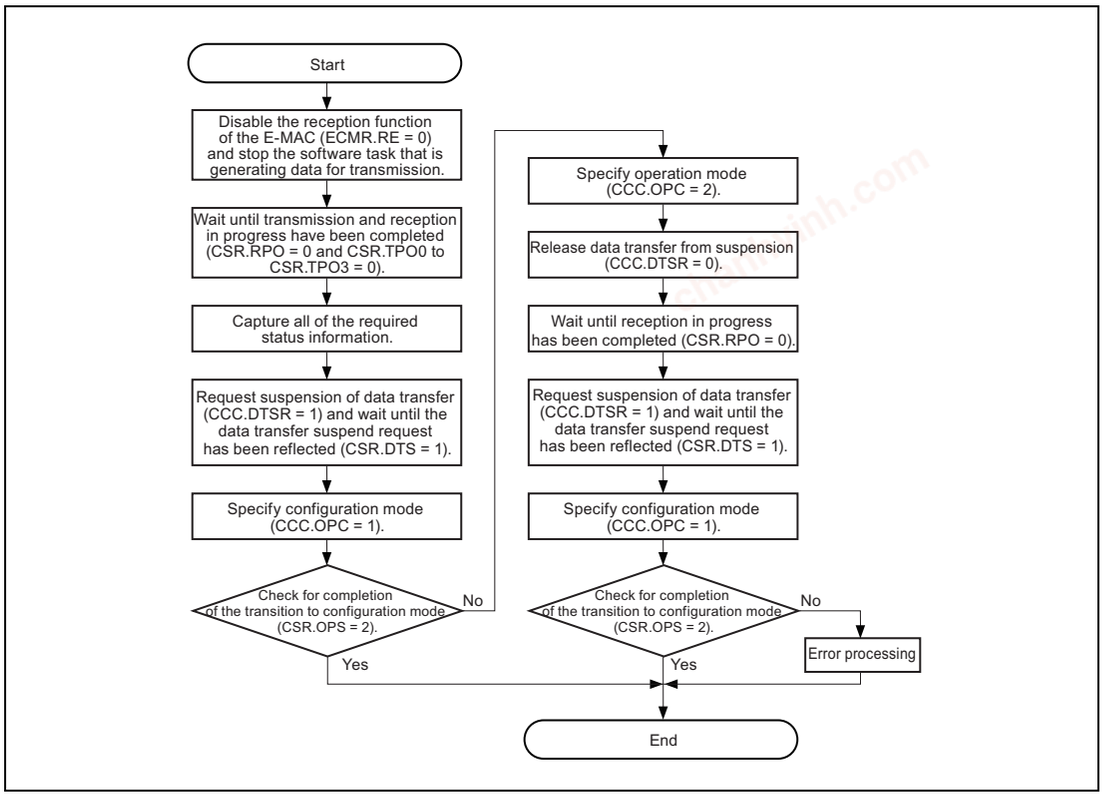


Figure 45.7 Flow for Transitions of Operating Mode (from Operation Mode to Configuration Mode)

In the transition from operation mode to configuration mode, the AVB-DMAC executes the following operations before the transition is completed. Read the operating mode status bits in the AVB-DMAC status register (CSR.OPS) to check that the transition to configuration mode has been completed.

- If the transfer of a frame between the reception FIFO and on-chip RAM is in progress, this is completed
  (other received frames remaining in the FIFO and any frames that are subsequently received by the E-MAC are discarded).
- If the transfer of a frame is in progress between the transmission FIFO and on-chip RAM, this is completed
  (frames for transmission remaining in the on-chip RAM will not be transmitted).
- All frames for transmission in the transmission FIFO are transferred to the E-MAC.

Notes:

When the operating mode shifts to configuration mode, all status registers are cleared.

We recommend following the procedure below in the case of this transition.

1. Disable reception.
2. Since reception actually stopping after being disabled requires time, wait for an interval equivalent to that for reception of a maximum length packet.
3. Stop the software task that is generating data for transmission.
4. Wait until the receive process status bit (CSR.RPO) and the transmit process status bits (CSR.TPO0 to 3) in the AVB-DMAC status register are set to 0.
5. Capture all of the required status information.

chanhvinh.com

6. Set the operating mode configuration bits in the AVB-DMAC mode register (CCC.OPC) to initiate the transition to configuration mode.

45.3.1.3 Operating Mode Transitions Due to Hardware

The following hardware factors can also initiate transitions of the AVB-DMAC operating mode.

(1) Power-on reset

Resetting of the LSI chip leads to resetting of the entire EthernetAVB module. The operating mode shifts to reset mode.

chanhvinh.com

45.3.2 Common Control for Transmission and Reception

45.3.2.1 Initialization Procedure

Figure 45.8 shows the overall initialization procedure in outline.

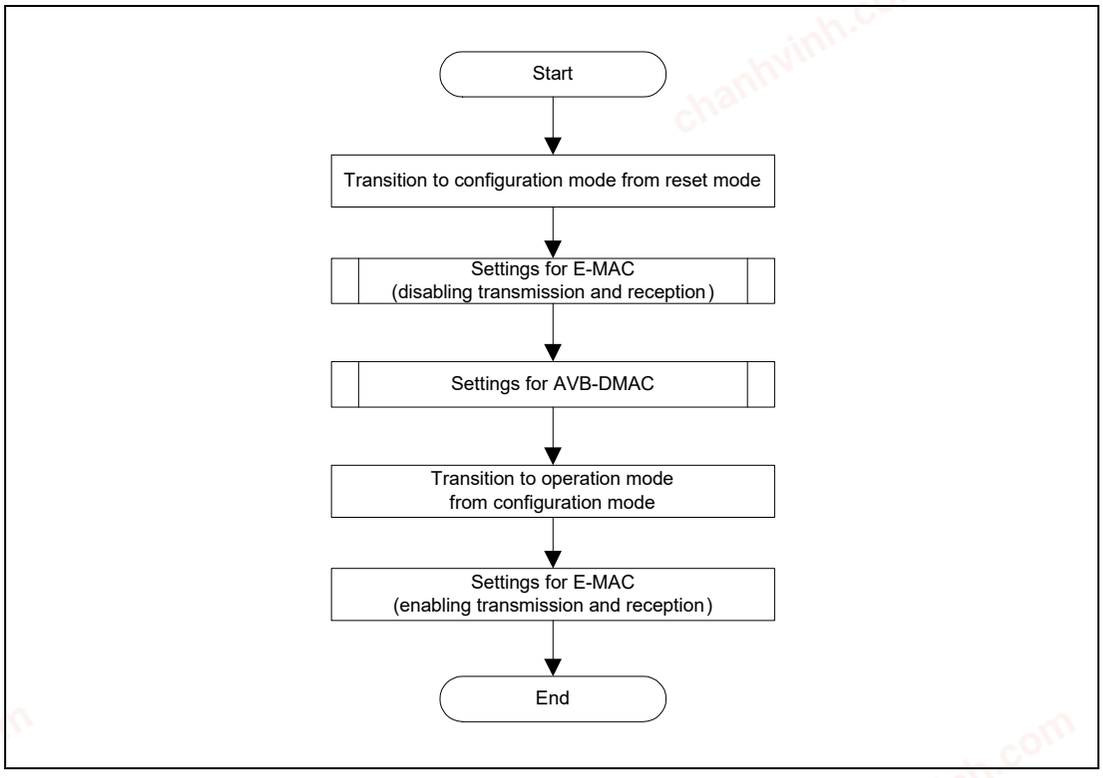


Figure 45.8 Outline of the Initialization Procedure

(1) Initializing the Receiver Section

Before starting reception, follow the procedure below.

Set the operating mode to operation mode or standby mode, and do not enable reception until the settings for the AVB-DMAC are completed.

- Set the operating mode to configuration mode.
- Set AVB filtering for network control frames and AVB stream frames to suit the specifications of the product the chip will be used in.
- Create a descriptor chain for each queue to be used.
- Set the base address for the descriptor table in the descriptor base address table register (DBAT).
- Specify the maximum frame length with the receive frame length upper limit register (RFLR).
- Specify whether padding is to be used with the receive padding configuration register (RPC).
- Set the unread frame counter for each queue with unread frame counter registers 0 to 4.

chanhvinh.com

(2) Initializing the Transmitter Section

Figure 45.9 illustrates initialization of the transmitter section.

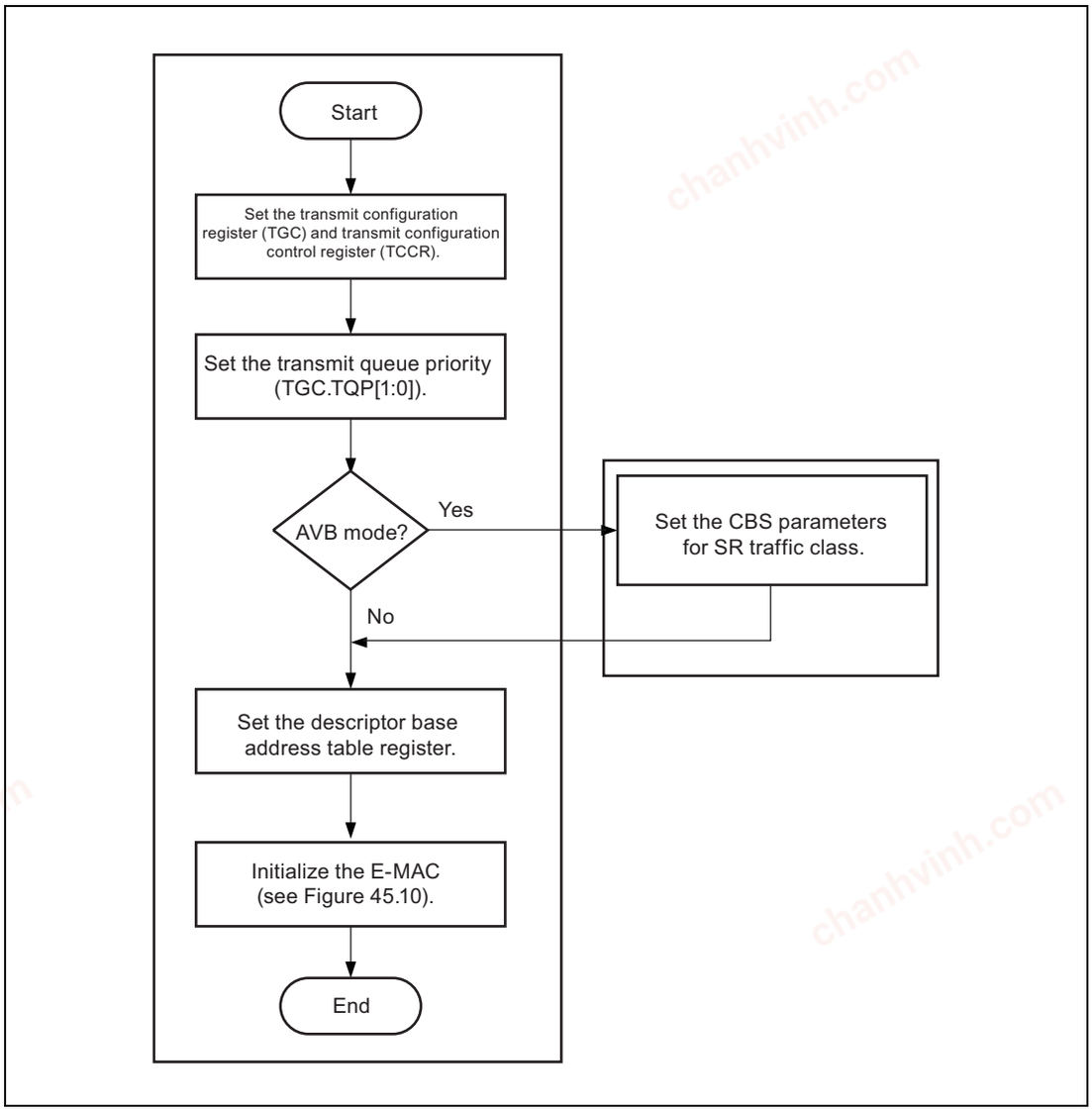


Figure 45.9 Procedure for Initializing the Transmitter Section

chanhvinh.com

(3) Initializing the E-MAC Section

Figure 45.10 illustrates initialization of the E-MAC section.

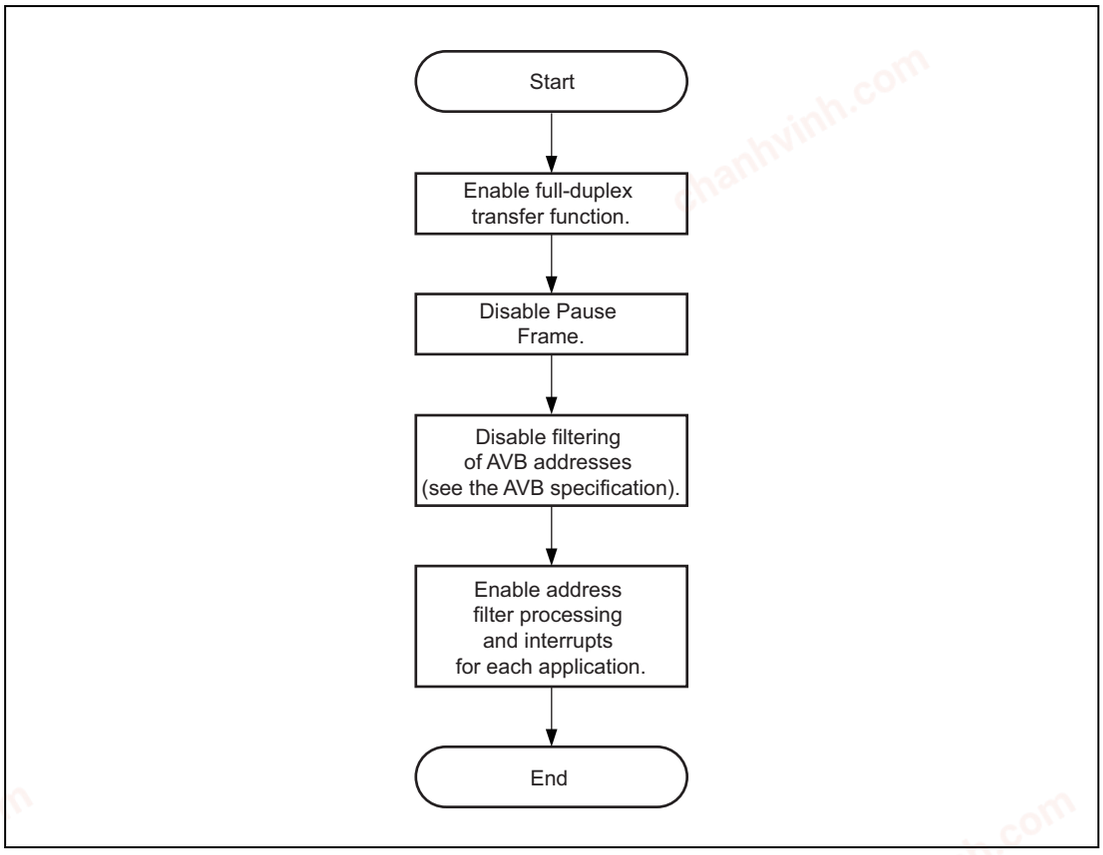


Figure 45.10 Procedure for Initializing the E-MAC Section

chanhvinh.com

(4) Initialization of the AVB-DMAC Unit

Figure 45.11 illustrates initialization of the AVB-DMAC unit.

For a description of how to set up the descriptors and the CBS, see section 45.3.3, Descriptors, and section 45.3.6, CBS (Credit-Based Shaping).

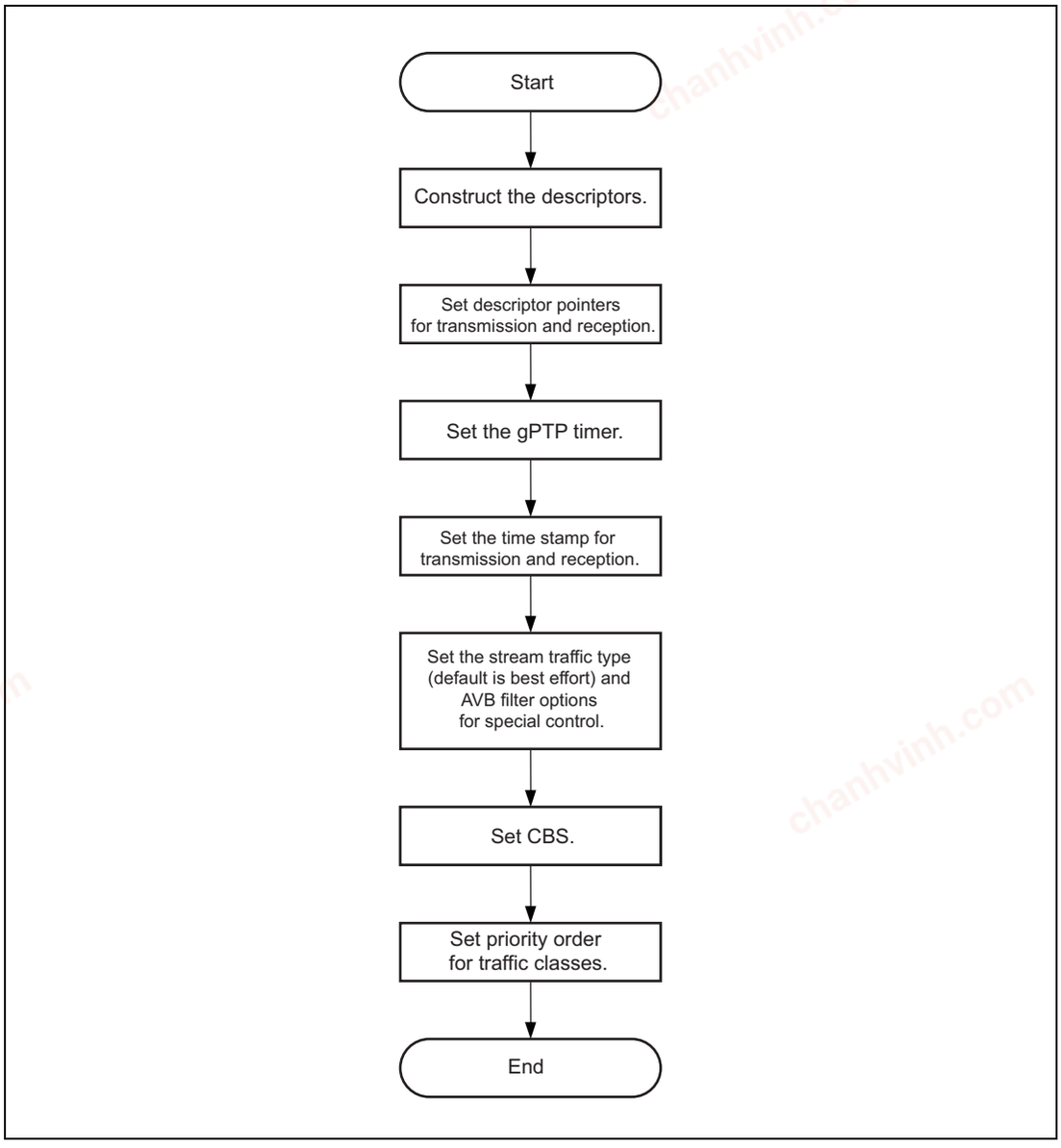


Figure 45.11 Procedure for Initializing the AVB-DMAC Unit

chanhvinh.com

45.3.2.2 Scheduling Reception and Transmission

The EthernetAVB normally has independent buses for transmission and reception. Furthermore, the four processes of fetching, storing, transmission and reception are basically independent of one another. Fetching and storing, however, share the same bus master so cannot be executed simultaneously. Access to the bus master is controlled by the scheduler.

Figure 45.12 is a schematic view of EthernetAVB operations in transmission and reception.

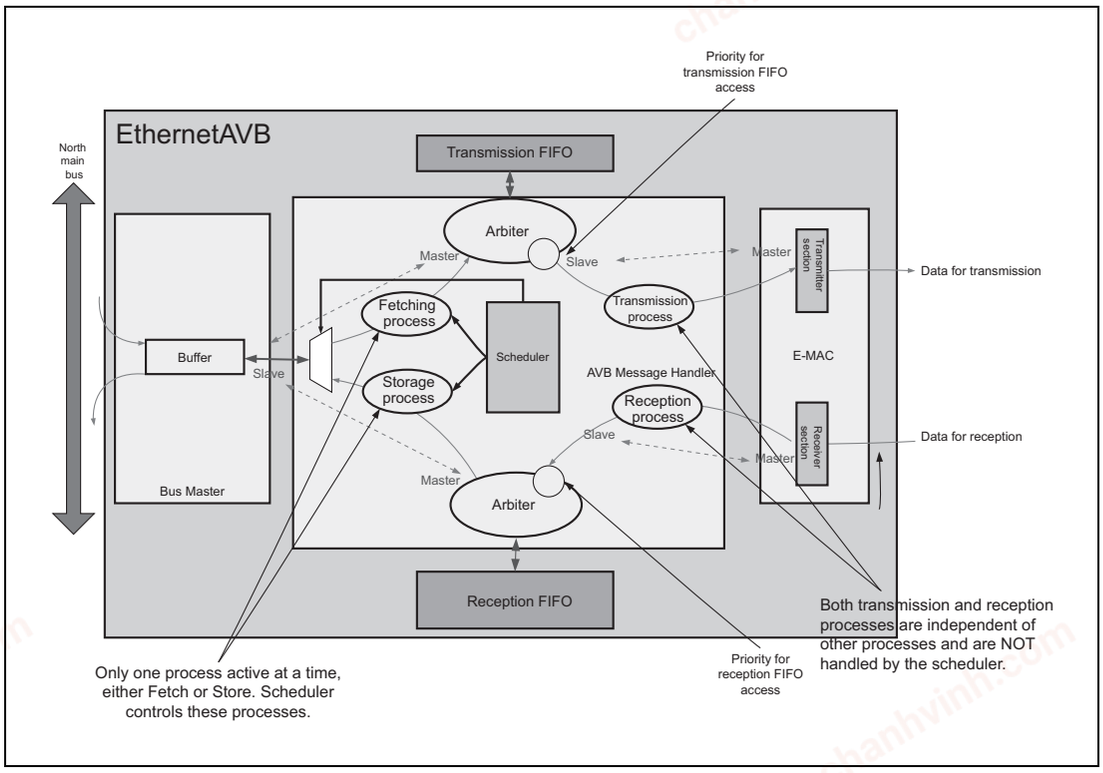


Figure 45.12 Schematic View of EthernetAVB Operations in Transmission and Reception

Storing and fetching are alternately performed. When the number of frames held by the reception FIFO reaches the warning level, storing takes precedence over fetching.

chanhvinh.com

(1) Relationship between Transmission Queue Numbers and Traffic Classes

In fetching, the relationships between the transmission queues and traffic classes are fixed, so the priority specified by the transmit queue priority bits in the transmit configuration register (TGC.TQP) has no effect.

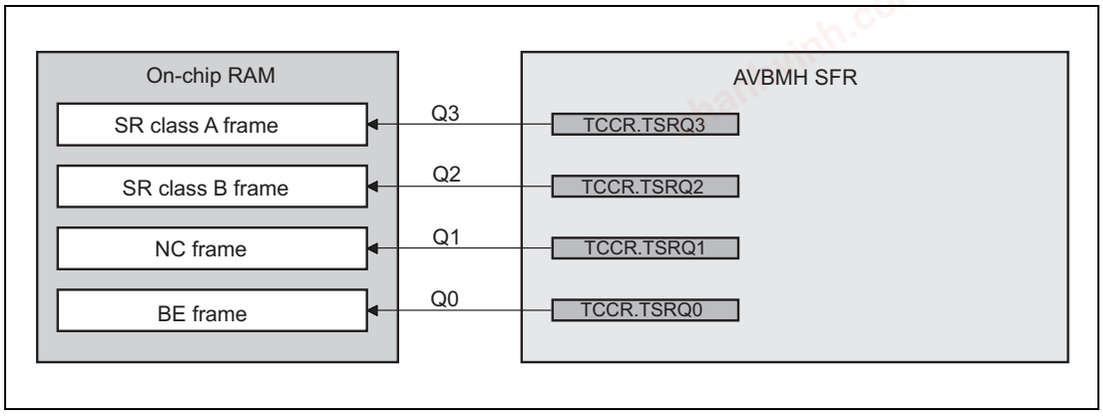


Figure 45.13 Class Associations of Queues for the Scheduler

In fetching, the credit values for stream classes A and B are not taken into account. Behavior depends on the setting of the transfer FIFO size configuration bits in the transfer configuration registers (TGC.TBDt).

When the transmit queue priority bits in the transmit configuration register (TGC.TQP) are B’00 or B'01, the priority order is Q3 → Q 2 → Q1 → Q0.

When the transmit queue priority bits in the transmit configuration register (TGC.TQP) are B’11, the priority order is Q1 → Q3 → Q2 → Q0.

chanhvinh.com

45.3.2.3 Checking Integrity

The EthernetAVB is capable of detecting and identifying errors produced in the processing of Ethernet frames and in the transfer of frame data for transmission and reception.

(1) Items for Monitoring in Both Reception and Transmission

(a) Errors in access to the on-chip RAM for reading of descriptors

- The error type bits (ESR.ET) are set to B'0000, and the queue number being processed is set to the error queue number bits (ESR.EQN).
- The same descriptor may be processed again because the current descriptor address (CDARq.CDA) was not changed.
- If this problem occurs in a divided frame, the sequence may be broken.

In reception

- The received frame will be lost.
- The same problem will occur for the next frame of data received for the same queue.

In transmission

- The transmit start request bit in the transmit configuration control register (TCCR.TSRQt) is set to 0.
- The frame will be lost from the transmission FIFO.

(b) Illegal configuration of a descriptor

- The error type bits (ESR.ET) are set to B'0010, and the queue number being processed is set to the error queue number bits (ESR.EQN).
- The same descriptor may be processed again because the current descriptor address (CDARq.CDA) was not changed.
- If this problem occurs in a divided frame, the sequence may be broken.

In reception

- The received frame will be lost.
- The same problem will occur for the next frame of data received for the same queue.

In transmission

- The transmit start request bit in the transmit configuration control register (TCCR.TSRQt) is set to 0.
- The frame will be lost from the transmission FIFO.

(c) Errors in access to the on-chip RAM for writing of descriptors

- The error type bits (ESR.ET) are set to B'0001, and the queue number being processed is set to the error queue number bits (ESR.EQN).
- As in the case where no error occurs, the current descriptor address (CDARq.CDA) and the transmit start request bit in the transmit configuration control register (TCCR.TSRQt) are updated.

chanhvinh.com

- As DESCR.DT was not updated, hardware and software synchronization may have been lost.

(2) Items for Monitoring in Reception

(a) Errors in access to the on-chip RAM for writing of data or time stamps

- The error type bits (ESR.ET) are set to B'0101, and the queue number being processed is set to the error queue number bits (ESR.EQN).
- As in the case where no error occurs, the current descriptor address (CDARq.CDA) is updated.
- DESCR.EI is set to 1 to indicate error detection.
- This problem occurring in a divided frame may break the descriptor sequence, making the queue unusable.

(b) Overflow of the Reception FIFO

- The reception FIFO full interrupt status bit (RIS2.RFFF) is set to 1.
- The received frames are all invalidated. The frames stored in the reception FIFO must be discarded according to the following procedure.

1. Stop reception and enter configuration mode according to the flow shown in **Figure 45.7, Flow for Transitions of Operating Mode (from Operation Mode to Configuration Mode)**.
2. Discard all the unprocessed received data stored in the on-chip RAM.

chanhvinh.com

(3) Items for Monitoring in Transmission

(a) Errors in Access for Reading Data from the on-chip RAM

- The error type bits (ESR.ET) are set to B'0100, and the queue number being processed is set to the error queue number bits (ESR.EQN).
- Data that have already been fetched are discarded from the transmission FIFO.
- When an error of this type occurs during processing of an FSINGLE or FEND descriptor:
  As in the case where no error occurs, the current descriptor address (CDARq.CDA) and the transmit start bit in the transmit configuration control register (TCCR.TSRQt) are updated. Fetching resumes after the error frame.
- When an error of this type occurs during processing of an FSTART or FMID descriptor:
  - The current descriptor address (CDARq.CDA) is not updated.
  - The transmit start bit in the transmit configuration control register (TCCR.TSRQt) is set to 0.

(b) Overflow of the Transmission FIFO

- The error type bits (ESR.ET) are set to B'1010, and the queue number being processed is set to the error queue number bits (ESR.EQN).
- As in the case where no error occurs, the current descriptor address (CDARq.CDA) and the transmit start bit in the transmit configuration control register (TCCR.TSRQt) are updated. Fetching resumes after the error frame.
- The frame will be discarded from the FIFO.

(c) Frame size error during transmission

- The error type bits (ESR.ET) are set to B'1001, and the queue number being processed is set to the error queue number bits (ESR.EQN).
- As in the case where no error occurs, the current descriptor address (CDARq.CDA) and the transmit start bit in the transmit configuration control register (TCCR.TSRQt) are updated. Fetching resumes after the error frame.

A transmit frame size error is detected when the size setting in one or more (in the case of a divided frame) descriptors for frame transmission is more than 1996 bytes. Such frames are cut out and transmitted.

chanhvinh.com

45.3.3 Descriptors

45.3.3.1 Data Representation in On-chip RAM

The AVB-DMAC transfers data for transmission and received data to and from the application software via the on-chip RAM.

Control structures referred to as descriptors and the frame data referred to as descriptor data area are allocated to the on-chip RAM. Dividing into a control area and data area allows the flexible allocation of frame data to the on-chip RAM.

Figure 45.14 shows an example of the memory maps for descriptors and the descriptor data area in the on-chip RAM.

A descriptor consists of its type (DESCR.DT), which controls the descriptor functions, a descriptor pointer (DESCR.DPTR) indicating the start address for storage of the frame data in the descriptor area, and the data size field (DESCR.DS), indicating the amount of frame data. Post-processing interrupt generation can be set up for each descriptor. Enabling and disabling of the interrupt is controlled by the descriptor interrupt enable bits (DESCR.DIE).

The descriptor may also hold information related to content. This information does not affect general descriptor functions. It provides information other than the frame data proper, such as on the state of reception.

For details, see section 45.3.4.2, Setting Up Reception Descriptors, and section 45.3.5.2, Setting Up Transmission Descriptors.

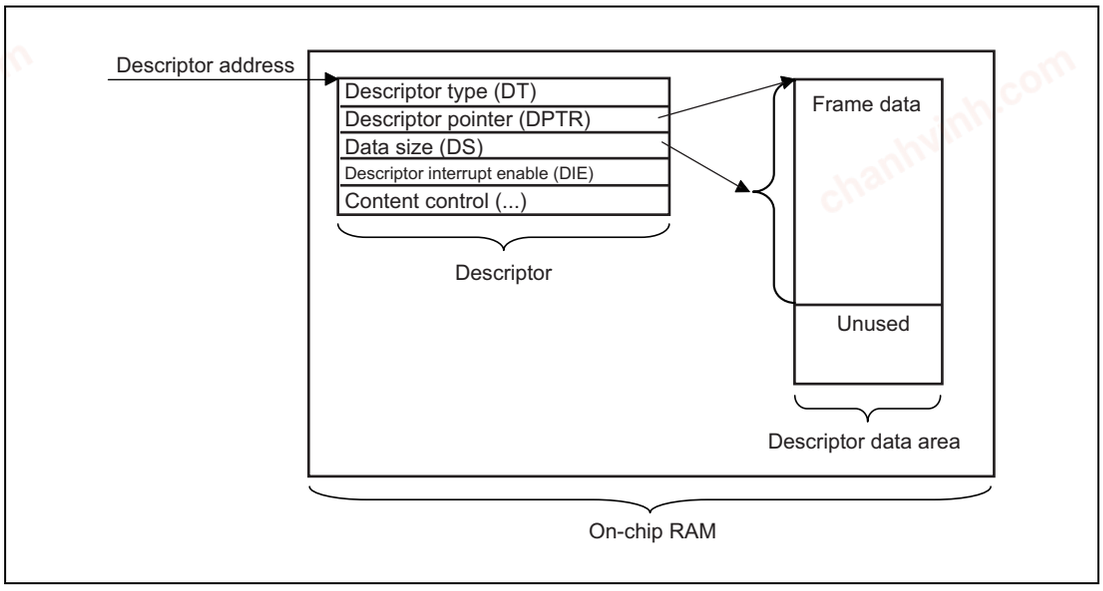


Figure 45.14 Example of On-chip RAM Memory Map

The descriptor must be aligned with a 32-bit boundary in the on-chip RAM.

Descriptors are generally configured of 64 bits, but are configured of 160 bits when reception and storage of gPTP time stamps is enabled.

The frame data must also be aligned with a 32-bit boundary in the on-chip RAM.

The amount of data in the frame is defined by the data size bits (DESCR.DS). In reception, these bits indicate the upper limit on the size of frames to be received. If the data size is not aligned with a 32-bit boundary, the bytes to the next 32-bit boundary in the data area will be an unused area.

chanhvinh.com

45.3.3.2 Using Descriptor Chains in Queues

Transmission and reception descriptors in the on-chip RAM are grouped into queues. Each queue handles frames so that they are transmitted in order of priority and received separately. A queue is capable of controlling one or more frames. Accordingly, multiple descriptors can be assigned to one queue. A combination of multiple descriptors is referred to as a descriptor chain.

For a descriptor chain, the three general descriptor types listed below are defined. For details on these descriptor types, see section 45.3.3.6, Descriptor Type.

- Descriptors that define frame data
- Descriptors that control the descriptor chain itself (e.g. LINK, EOS).
- Descriptors that arbitrate access by hardware or software

Figure 45.15 shows the two basic topologies for descriptor chains. In the simplified examples in the figure, all descriptors allocated to the chain are stored in the array.

- For a linear descriptor chain, the last descriptor in the array is a control descriptor indicating the end of the descriptors (e.g. EEMPTY).
- For a cyclic descriptor chain, the last descriptor in the array is a control descriptor that returns to the first descriptor in the array (e.g. LINK).

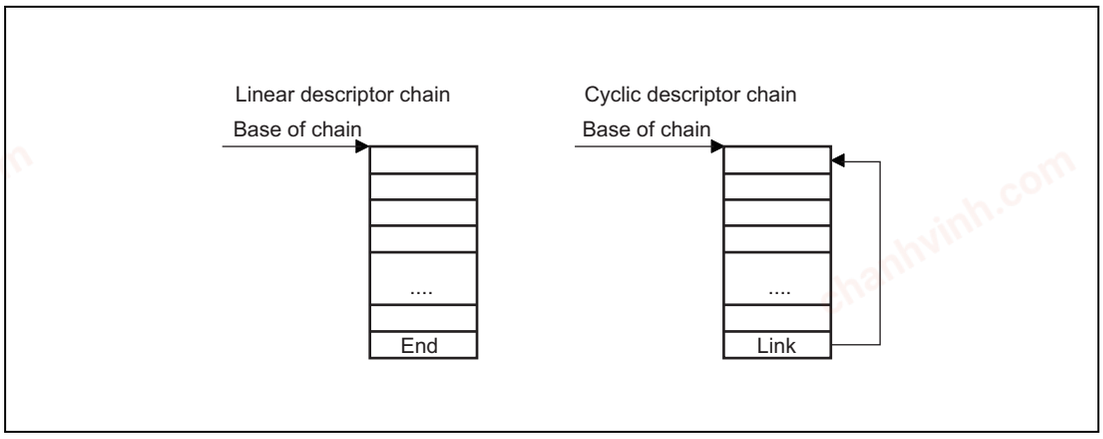


Figure 45.15 Outline of the Basic Descriptor Chains

The relationship between queues and descriptor chains is defined by the base addresses of chains. A queue is connected to one descriptor chain over one round of processing. There is also a method of switching to a different chain while in operation mode.

There are no restrictions on the number of link descriptors and their locations within the chain. The last descriptor of a designed chain determines the topology.

Which chain structure is to be used or which topology is suitable depends on the application. A description of how to design descriptor chains to suit various applications is given in section 45.3.4.2, Setting Up Reception Descriptors. Procedure for Setting Reception Descriptors, and section 45.3.5.2, Setting Up Transmission Descriptors.

chanhvinh.com

45.3.3.3 Descriptor Base Address Table

The base address table in the on-chip RAM contains the address of the first descriptor of all chains to be handled by the respective queues.

Entries 0 to 3 are used to access transmission queues 0 to 3. Subsequent entries are used to access reception queues. Entry 4 thus corresponds to reception queue 0.

The configuration of entries in the base address table is the same as the configuration of link descriptors. We recommend using the descriptor type (DESCR.DT) LINKFIX. Processing of this link descriptor does not change it, so it does not require updating. The first descriptor of a chain performs hardware and software synchronization. If the application requires hardware and software synchronization for the base addresses, use the descriptor type (DESCR.DT) LINK.

The CPU is only capable of using LINKFIX and LINK as descriptor types (DESCR.DT) of descriptors in the base address table.

Set the location of the base address table in the on-chip RAM in the descriptor base address table register (DBAT).

Figure 45.16 shows an example of a base address table for controlling four transmission and three reception queues. The boxes to the right of the table represent descriptor chains with the desired topologies.

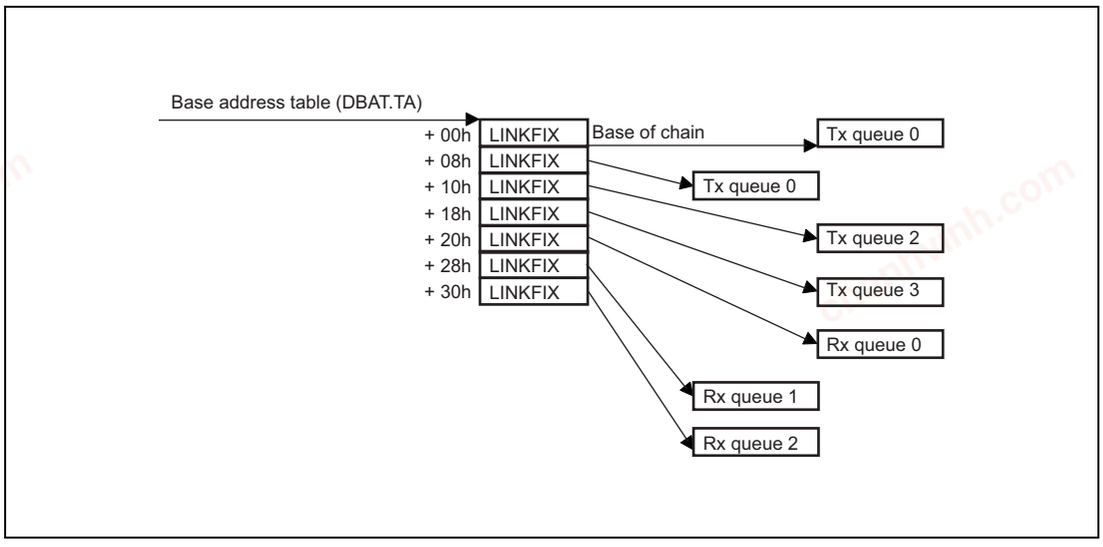


Figure 45.16 Example of a Base Address Table for Reception and Transmission Queues

CAUTION

The size of the descriptors in the base address table is always eight bytes even if the queue itself includes extended descriptors.

chanhvinh.com

45.3.3.4 Descriptor Chain Processing

When a descriptor is currently being processed or the queue for a descriptor is active, the current descriptor is a descriptor to be processed. The current descriptor address for use by a queue q can be checked in the current descriptor address register q (CDARq).

Current descriptors are:

- set in the descriptor base table address bits (DBAT.TA) +8*q for all q queues when the operating mode shifts to operation mode.
- set in the descriptor base table address bits (DBAT.TA) +8*q when a base address load request is issued for a queue q by setting the corresponding bit (DLR.LBAq) in the descriptor base address load request register (DLR).
- set in DESCR.DPTR for a link descriptor (LINK, LINKFIX) to be processed.
- incremented by the size of the descriptors (8 bytes for normal descriptors and 20 bytes for extended descriptors) after a descriptor has been processed. In this case, the AVB-DMAC updates the descriptor type and informs the CPU that the descriptor has been processed.

In other cases (for illegal descriptor processing and so on), the current descriptor address register q (CDARq) is not updated.

chanhvinh.com

45.3.3.5 Descriptor Interrupts

A descriptor is able to issue a descriptor interrupt on completion of its processing. The setting of the descriptor interrupt enable bits (DESCR.DIE) in each descriptor can disable the descriptor interrupts or select the specific descriptor interrupt to be generated.

The descriptor interrupt is a common resource that is shared between reception and transmission queues.

Figure 45.17 illustrates the way in which the AVB-DMAC generates descriptor interrupts (or sets bits in the descriptor interrupt status register (DIS.DPFi)). Processing of a descriptor with the value i in the descriptor interrupt enable bits (DESCR.DIE) leads to the corresponding bit in the descriptor interrupt status register (DIS.DPFi) being set.

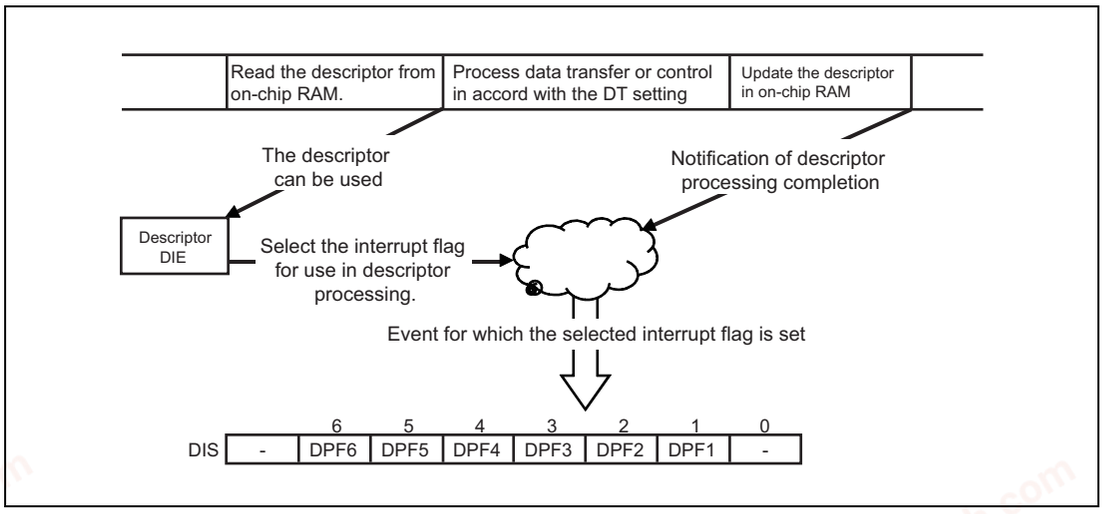


Figure 45.17 Method of Descriptor Interrupt Generation

chanhvinh.com

45.3.3.6 Descriptor Type

The descriptor types (indicated by the DESCR.DT bits) supported by the AVB-DMAC fall into the following three categories.

- Definitions of frame data
- Control of descriptor chains
- Hardware and software arbitration

Table 45.71 is a summary of the descriptor types available for the AVB-DMAC. Entries under “Name” are the names of the descriptor types and the values under “DT” are the corresponding values to be set in the descriptor type field (DESCR.DT). A given descriptor may be handled differently according to whether it is in a transmission or reception queue, so the transmission and reception columns list the scopes of control and processing of the descriptor types.

The abbreviations defined below are used in the transmission and reception columns.

Definition of SW:

- The descriptor is processed by software.
- Software has access to and may modify the descriptor and descriptor data area.
- This descriptor cannot be changed by hardware (AVB-DMAC).

Definition of HW:

- The descriptor is processed by hardware (AVB-DMAC).
- Software must modify neither the descriptor nor the descriptor data area.
- Hardware (AVB-DMAC) processes this descriptor and subsequently changes the descriptor type.

Invalid:

This descriptor type is not used in transfer in the given direction (transmission or reception).

Do not write this value to the descriptor type (DESCR.DT) field for transfer in the given direction.

Hardware does not process these descriptor types in the cases listed as invalid. The current descriptor address (CDARq.CDA) will not be changed when processing of a queue for the given direction arrives at a descriptor with this type setting.

chanhvinh.com

Table 45.71 Summary of Descriptor Types

Name	DT	Description	Reception	Transmission
Frame data				
FSTART	5	Frame Start The descriptor points to valid data for a frame. The frame starts with the given data and continues with that indicated by the next descriptor.	SW	HW
FMID	4	Frame Middle The descriptor points to valid data for a frame. The frame started with a previous descriptor and continues to the data indicated by the next descriptor.	SW	HW
FEND	6	Frame End The descriptor points to valid data for a frame. The frame continues from the previous descriptor and ends with the data indicated by in this descriptor.	SW	HW
FSINGLE	7	Frame Single The descriptor points to valid data for a complete frame.	SW	HW
Chain control				
LINK	8	Link Defines the descriptor in the next chain.	HW	HW
LINKFIX	9	Fixed Link Defines the descriptor in the next chain. Unlike the case of Link, it is not rewritten by hardware after descriptor processing.	SW	SW
EOS	10	End Of Set Controls division of the chain.	HW	HW
HW/SW arbitration				
FEMPTY	12	Frame Empty A descriptor related to frame data but not containing valid data for a frame	HW	SW
FEMPTY_IS	13	Frame Empty Incremental Start A descriptor related to frame data but not containing valid data for a frame DESCR.DPTR sets the base address of an "incremental data area" in the on-chip RAM.	HW	Invalid
FEMPTY_IC	14	Frame Empty Incremental Continue A descriptor related to frame data but not containing valid data for a frame Data indicated by the pointer are for storage in an incremental data area in the on-chip RAM.	HW	Invalid
FEMPTY_ND	15	Frame Empty No Data Storage A descriptor related to frame data but not containing valid data for a frame The descriptor is processed in the same way as FEMPTY but data are not stored in the on-chip RAM.	HW	Invalid
LEMPTY	2	Link Empty A link descriptor for processing by the AVB-DMAC	SW	SW
EEMPTY	3	EOS Empty An EOS descriptor for processing by the AVB-DMAC	SW	SW
DT0	0	Reserved	Invalid	Invalid
DT1	1	Reserved	Invalid	Invalid
DT11	11	Reserved	Invalid	Invalid

chanhvinh.com

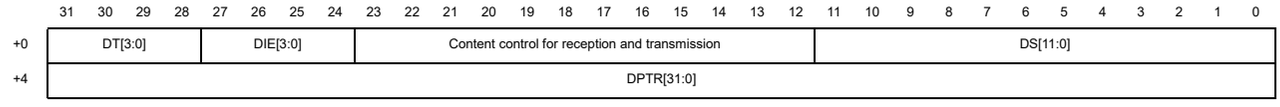
(1) Layout of General Descriptors in the On-chip RAM

The AVB-DMAC updates processed descriptors in the on-chip RAM. The field to be changed in a descriptor being updated depends upon whether the direction is transmission or reception and the queue mode. Other fields will not be changed. There are no restrictions on the values set in unused descriptor fields (indicated by "—" in the figure).

(2) Frame Data Descriptors

The allocation of bits in the frame data descriptors (FSTART, FMID, FEND, and FSINGLE) is shown below.

- Normal descriptor (usable in both reception and transmission)



- Extended descriptor (usable only in reception)

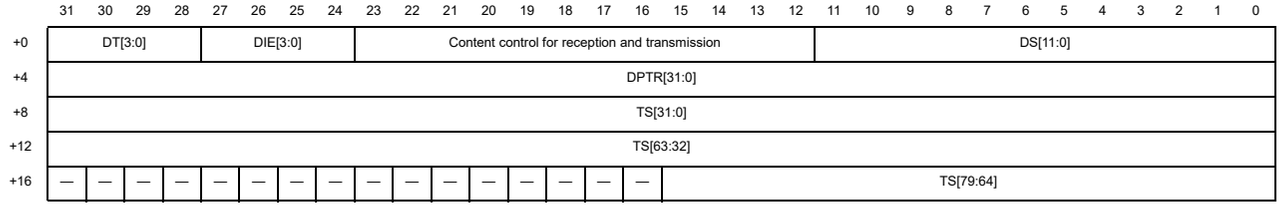


Table 45.72 Contents of Frame Data Descriptors (DESCR)

Bit Name	Function
DT[3:0]	Descriptor Type 5: FSTART 4: FMID 6: FEND 7: FSINGLE For details, see section 45.3.4.2, Setting Up Reception Descriptors, and section 45.3.5.2, Setting Up Transmission Descriptors.
DIE[3:0]	Descriptor Interrupt Enable B'0000: Descriptor interrupt is disabled. B'0001 to B'1111: The corresponding descriptor interrupt is generated (DIS.DPFi).
—	Content Control For details, see section 45.3.4.2, Setting Up Reception Descriptors, and section 45.3.5.2, Setting Up Transmission Descriptors.
DS[11:0]	Data Size Size of the data area/frame data for the descriptor (in bytes) The specifiable data lengths are as follows (when a frame is divided into parts, the total size of the frame should be within the ranges below). Transmission: 1 ≤ DS ≤ 1996 Reception: 1 ≤ DS ≤ 2000
DPTR[31:0]	Descriptor Pointer Pointer to the data area for the descriptor Register an address on a 32-bit boundary.
TS[79:0]	Time Stamp Time stamp of the received frame (only available in extended descriptors)

chanhvinh.com

CAUTION

Register an address aligned with a 32-bit boundary as the descriptor pointer (DESCR.DPTR).

H'0000 is written to the reserved bits in an extended descriptor (bits 31 to 16 in DESCR) when the time stamp is stored.

(3) Hardware/Software Arbitration Descriptors (Only for Reception)

The allocation of bits in the descriptors for hardware/software arbitration (FEMPTY, FEMPTY_IS, FEMPTY_IC, and FEMPTY_ND) is shown below.

- Normal descriptor

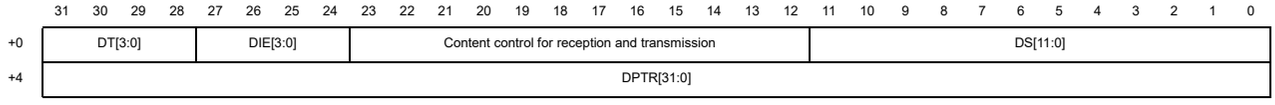


Table 45.73 Contents of Hardware/Software Arbitration Descriptors (DESCR)

Bit Name	Function
DT[3:0]	Descriptor Type 12: FEMPTY 13: FEMPTY_IS 14: FEMPTY_IC 15: FEMPTY_ND For details, see **Table 45.71, Summary of Descriptor Types**.
DIE[3:0]	Descriptor Interrupt Enable B'0000: Descriptor interrupt is disabled. B'0001 to B'1111: The corresponding descriptor interrupt is generated (DIS.DPFi).
—	Content Control For details, see section 45.3.4.2, Setting Up Reception Descriptors, and section 45.3.5.2, Setting Up Transmission Descriptors.
DS[11:0]	Data Size Size of the data area/frame data for the descriptor (in bytes)
DPTR[31:0]	Descriptor Pointer Pointer to the data area for the descriptor Register an address on a 32-bit boundary.

CAUTIONS

Register an address aligned with a 32-bit boundary as the descriptor pointer (DESCR.DPTR).

When an extended descriptor is used, a 12-byte unused area is added.

In an FEMPTY descriptor, the descriptor type (DT), descriptor interrupt enable (DIE), data size (DS), and descriptor pointer (DPTR) fields are used.

In an FEMPTY_IS descriptor, the descriptor type (DT), descriptor interrupt enable (DIE), and descriptor pointer (DPTR) fields are used.

In an FEMPTY_IC descriptor, the descriptor type (DT) and descriptor interrupt enable (DIE) are used.

In an FEMPTY_ND descriptor, the descriptor type (DT), descriptor interrupt enable (DIE), and data size (DS) are used.

chanhvinh.com

(4) Link Descriptors

The allocation of bits in the link descriptors (LINK and LINKFIX) is shown below.

- Normal descriptor

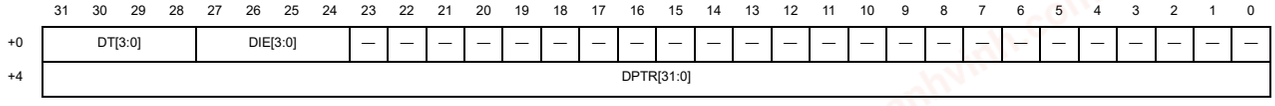


Table 45.74 Contents of Link Descriptors (DESCR)

Bit Name	Function
DT[3:0]	Descriptor Type 8: LINK 9: LINKFIX For details, see **Table 45.71, Summary of Descriptor Types**.
DIE[3:0]	Descriptor Interrupt Enable B'0000: Descriptor interrupt is disabled. B'0001 to B'1111: The corresponding descriptor interrupt is generated (DIS.DPFi).
—	Content Control For details, see section 45.3.4.2, Setting Up Reception Descriptors, and section 45.3.5.2, Setting Up Transmission Descriptors.
DPTR[31:0]	Descriptor Pointer Pointer to the data area for the next descriptor Register an address on a 32-bit boundary.

CAUTION

Register an address aligned with a 32-bit boundary as the descriptor pointer (DESCR.DPTR). When an extended descriptor is used, a 12-byte unused area is added.

chanhvinh.com

(5) Other Descriptors

The allocation of bits in the other descriptors (EOS, FEMPTY (only for transmission), LEMPTY, and EEMPTY) is shown below.

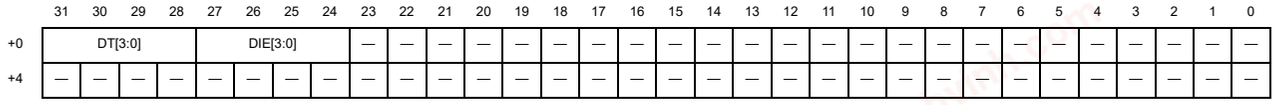


Table 45.75 Contents of Other Descriptors (DESCR)

Bit Name	Function
DT[3:0]	Descriptor Type 10: EOS 12: FEMPTY (only for transmission) 2: LEMPTY 3: EEMPTY For details, see **Table 45.71, Summary of Descriptor Types**.
DIE[3:0]	Descriptor Interrupt Enable B'0000: Descriptor interrupt is disabled. B'0001 to B'1111: The corresponding descriptor interrupt is generated (DIS.DPFi).

CAUTION

When an extended descriptor is used, a 12-byte unused area is added.

chanhvinh.com

(6) How to Use Frame Data Descriptors

The descriptor data area size field (DESCR.DS) can specify up to 2048 bytes of Ethernet frame data per data area. Settings higher than 2048 (bytes) cannot be made.

In general, Ethernet frames are not of uniform length. The AVB-DMAC is capable of dividing frame data into multiple descriptors in order to minimize the memory capacity for frame data. This function allows processing of frames that are longer than the limit for descriptor data areas. Division can also be applied to frames on the basis of their data structures.

To handle both frames divided up into multiple data areas and descriptors for complete single frames, four types (DESCR.DT) FSTART, FEND, FMID, and FSINGLE are defined.

Figure 45.18 shows the mapping of frame data by frame data descriptors. The descriptor data areas are allocated to the on-chip RAM. For frames that require division into four or more data areas, additional FMID descriptors can be added as required.

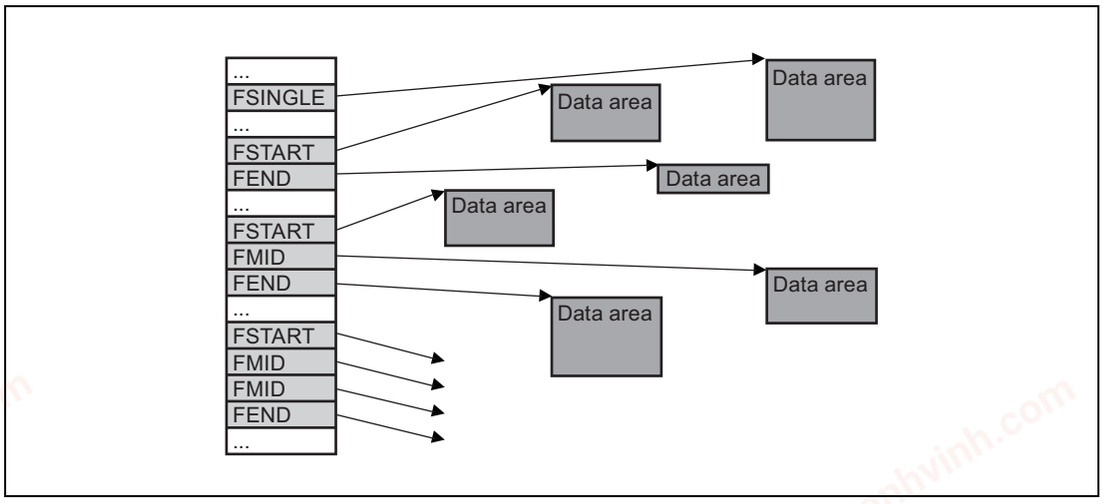


Figure 45.18 Mapping of Frame Data

For reception, set the descriptor data areas to the maximum size (i.e. give DESCR.DS its maximum value). The AVB-DMAC will store received frame data in the given area. If a received frame has more data than the maximum size, the AVB-DMAC will divide the data up.

For transmission, set the frame data size to the actual data size. The AVB-DMAC modifies the descriptor type (DESCR.DT) to FEMPTY after processing the relevant descriptor. The data size (DESCR.DS) and descriptor pointer (DESCR.PTR) fields retain their settings.

A descriptor data area including unused space produces an empty space between data areas. In reception, an "incremental data area" can be used to prevent empty spaces. For incremental data areas, see section 45.3.4.3 (2) Incremental Data Areas.

As well as reducing the memory capacity taken up by the descriptor area in the on-chip RAM, division into frames can be used to identify different sections of data (e.g. for separating a header and data).

chanhvinh.com

(7) How to Use Chain Control Descriptors

(a) Link Descriptors

The link descriptors can be used to set up cyclic descriptor chains (for details, see section 45.3.3.2, Using Descriptor Chains in Queues).

After a LINK descriptor is processed, its descriptor type (DESCR.DT) is changed to LEMPTY. The descriptor pointer (DESCR.PTR) retains its setting.

After processing of a LINKFIX descriptor, the descriptor type (DESCR.DT) is not updated. Software can change the descriptor type (DESCR.DT), descriptor interrupt enable (DESCR.DIE), and descriptor pointer (DESCR.DPTR). Take care, however, to check the current descriptor address register (CDARq.CDA) before changing the descriptor pointer (DESCR.DPTR).

(b) EOS Descriptor

Use the EOS descriptor to divide a descriptor chain into various segments. The queue can continue even after an EOS descriptor.

In transmission, the response to an EOS descriptor is clearing of the transmit start request bit in the transmit configuration control register (TCCR.TSRQq) to 0.

In reception, the response is generation of a receive queue full interrupt (RIS2.QFFr), although if the frame currently being received is being divided for storage (received data being stored in FMID- or FEND-type frames), the data are not completely stored.

(8) How to Use Hardware and Software Arbitration Descriptors

In hardware processing of descriptors, the empty descriptor types (FEMPTY, LEMPTY, and EEMPTY) are defined to distinguish various descriptors. For software, they can be used to initiate checking for empty spaces, etc.

(a) FEMPTY, FEMPTY_IS, FEMPTY_IC, and FEMPTY_ND

These descriptor types (DESCR.DT) are used for descriptors that do not contain effective data. Of these, only FEMPTY is used in transmission.

These descriptor pointers (DESCR.DPTR) indicate the descriptor data area.

(b) LEMPTY

This descriptor type (DESCR.DT) is assigned to LINK descriptors after they have been processed.

The descriptor pointer (DESCR.DPTR) of an LEMPTY descriptor still points to the linked descriptor.

(c) EEMPTY

This descriptor type (DESCR.DT) is assigned to EOS descriptors after they have been processed.

The descriptor pointer (DESCR.DPTR) of an EEMPTY descriptor is not used, and thus points to nothing.

chanhvinh.com

(9) Relationship between Descriptor Access by Hardware and Software

In the EthernetAVB, the allocation of descriptor types (DESCR.DT) to the on-chip RAM can be used. This makes it possible to minimize access to the EthernetAVB registers via the CPU, leading to higher performance.

- Each descriptor type in the set is exclusively for processing by hardware or software, depending on the direction of transfer (see Table 45.71).
- Software must not change a descriptor assigned to hardware processing (the hardware does not change descriptors assigned to software processing).

In the case of software processing, the software must process the information in the descriptor and the corresponding frame data before changing the descriptor type. If a descriptor type for hardware is set in DESCR.DT, the software should not change any part of the descriptor or of the corresponding frame data.

45.3.3.7 Optimization of Bus Performance when Using Descriptors

The following items are recommended as ways to ensure the optimal use of data structures in the on-chip RAM.

They are not requirements, but using a different approach may increase the load on the system bus within the LSI chip.

- Register descriptors with 64-bit alignment (this does not apply to extended descriptors).
- While in operation mode, use LINKFIX instead of LINK whenever a descriptor need not be changed. Hardware modifies the descriptor type (DESCR.DT) fields of LINK descriptors.
- Restrict frame data to a maximum size of 128 bytes.
- Minimize parallelism of processing to the descriptor chains.
  This helps in arranging the different segments exclusively for access by software or hardware by dividing the chains so as to be allocated to different on-chip RAM pages.
- Minimize the number of divided frames. This can reduce the overhead of descriptor handling.

chanhvinh.com

45.3.4 Control in Reception

The point of the AVB-DMAC is to transfer data between the E-MAC and on-chip RAM without intervention by the CPU.

Create descriptors that define the amounts of frame data to be stored and the locations. After the E-MAC receives a frame, it stores the received frame data and the conditions of reception as the MAC state. If the descriptor is extended, the time stamp is also stored. For a description of how to set up descriptors for use in reception, see section 45.3.4.2, Setting Up Reception Descriptors.

The AVB-DMAC filters received frames to separate them into various classifications (separation filtering). More specifically, this is done to separate received frames into the various reception queues and to set the priorities of different classes of received frames. For more on separation filtering, see section 45.3.4.1 (1) Separation Filtering.

Figure 45.19 shows the reception data bus and the selection of queues for use in reception.

Each frame received from the E-MAC is stored in the reception FIFO; in parallel with this, the frame is analyzed to identify its type and the target queue number. After the E-MAC completes reception, the target queue number is generated and stored in the reception FIFO. A reception flag is applied to each reception queue in the on-chip RAM, and the unread frame counter (UFC) is also applied to each reception queue.

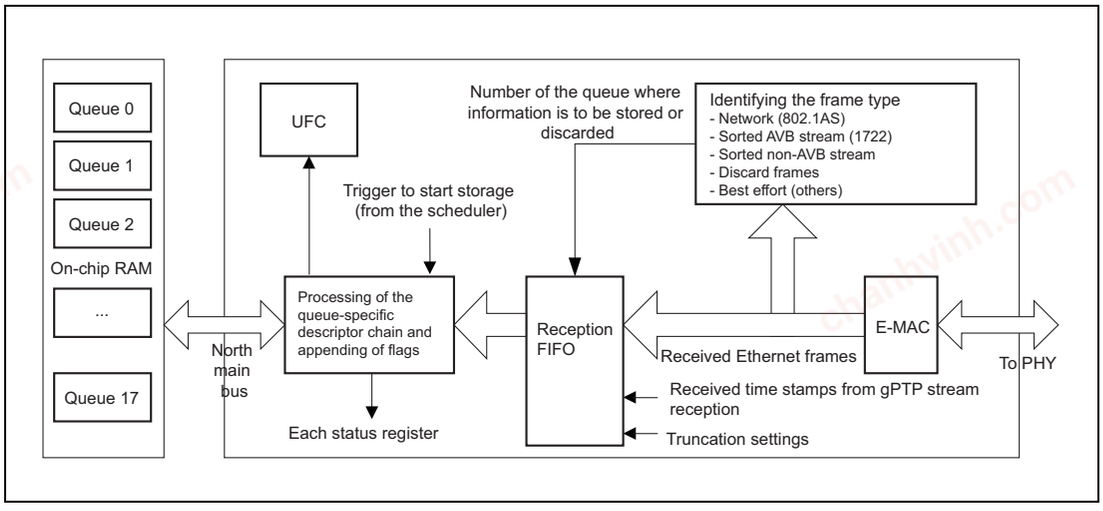


Figure 45.19 Mechanism of General Reception Queue Selection

chanhvinh.com

45.3.4.1 Reception Queues

The AVB-DMAC applies its separation filtering mechanism to select the reception queue for storing a received frame. The AVB-DMAC stores all received frames in the on-chip RAM.

There are two conditions for the AVB-DMAC to discard a received frame.

- Detection of an error during reception by the E-MAC
  - Whether error frames are discarded or stored in reception queue 0 (best effort) can be set by the error frame enable bit in the receive configuration register (RCR.EFFS). If error frames are to be stored (RCR.EFFE = 1), they are always stored in queue 0 (best effort). In this case, parameters of each queue (e.g. truncation) may not match. If the storage of time stamps for reception queue 0 (best effort) is enabled (the time stamp enable bit in the receive configuration register RCR.ETS0 = 1), time stamps are stored even for error frames.
- Non-matching determination conditions of the separation filter
  - Whether non-matching frames are discarded or stored in reception queue 0 (best effort) can be set by the stream filtering select bit in the receive configuration register (RCR.ESF).

The flowchart in Figure 45.20 shows how the AVB-DMAC selects the reception queue in accord with the frame type, including judgment by the separation filter. Selection of the queue starts when the E-MAC completes frame reception. The result is storage of the frame in the proper queue or the frame being discarded.

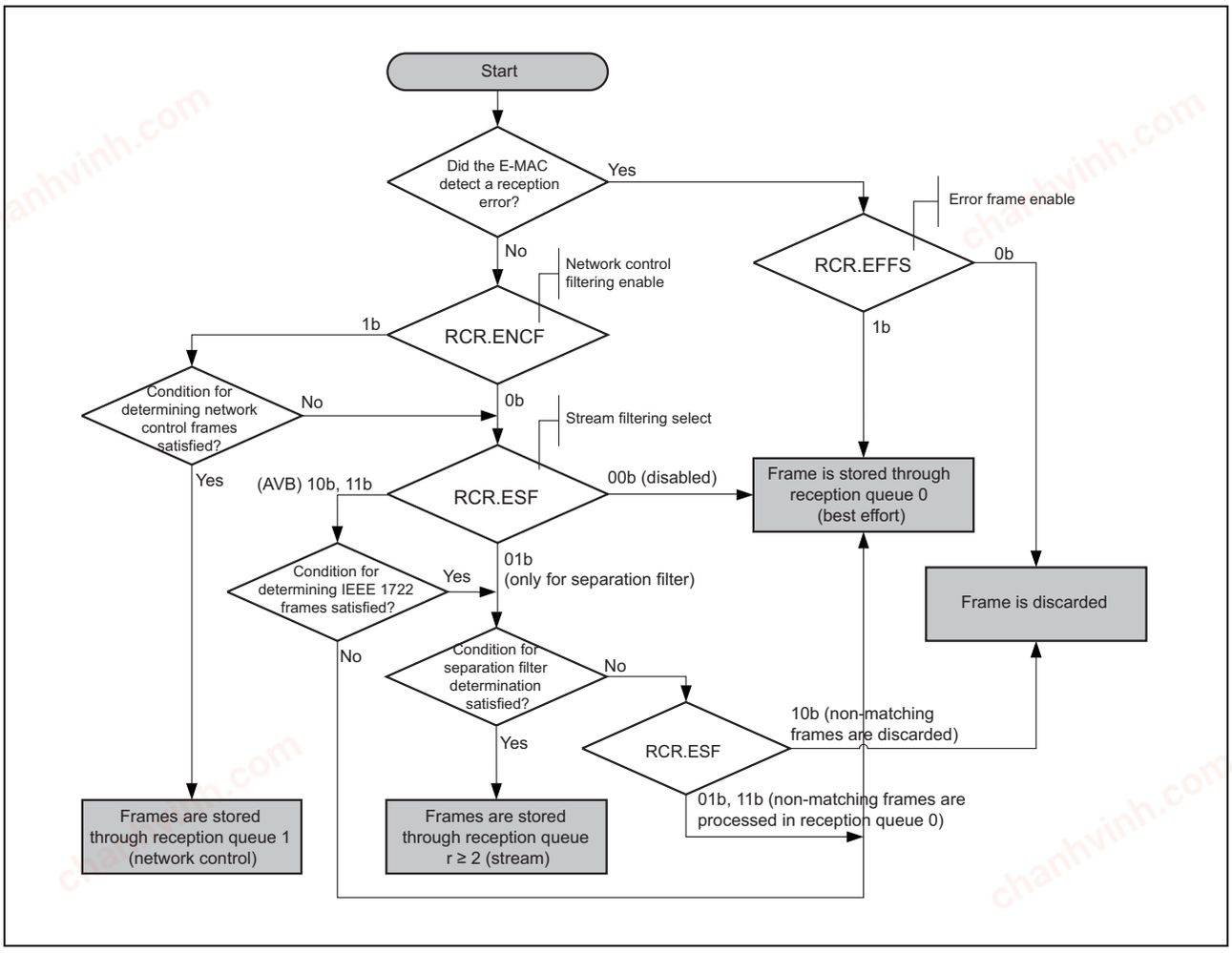


Figure 45.20 Flow of Reception Queue Selection

chanhvinh.com

Notes on the description of entries in the flowchart

- “Condition for determining network control frames”
  The Ethernet destination address (DA) is 01:80:C2:00:00:0E, and
  The Ethernet type (ET) is 88:F7.
- “Condition for determining IEEE 1722 frames”
  The Ethernet destination address (DA) is in the range from 91:E0:F0:00:00:00 to 91:E0:F0:00:FE:FF,
  The VLAN tagged TPID (tag protocol identifier) field (VL) is 81:00, and
  The Ethernet type (ET) is 22:F0.
- “Condition for separation filter determination”
  See section 45.3.4.1 (1) Separation Filtering.

Figure 45.21 shows the allocation of data in Ethernet frames used for determination of Network Control frame and IEEE1722 frame. The preambles of Ethernet frames are not taken into account.

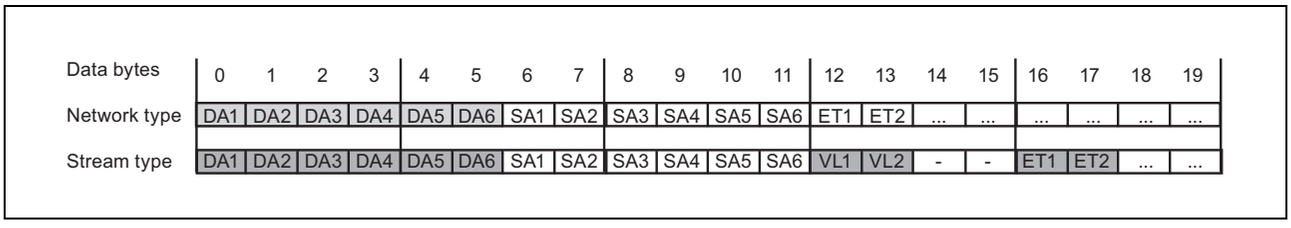


Figure 45.21 Data Allocation of Ethernet Frames Used in Frame Determination

(1) Separation Filtering

Separation filtering involves the checking of 64 bits (eight successive bytes) in received Ethernet frames. The setting for the first byte (i.e. the setting of the separation filter offset configuration register (SFO.FBP)), selects the part of frames to be used in separation filtering. There is also a common filter mask (set in the separation filter mask configuration register (SFMi.CFM)) that can filter less than the number of bytes and mask particular bits.

Examples

To use only the first byte from the top in separation filter, set separation filter mask configuration register 0 (SFM0.CFM) to H’0000 00FF and separation filter mask configuration register 1 (SFM1.CFM) to H’0000 0000.

To use seven bytes from the top in separation filter, set separation filter mask configuration register 0 (SFM0.CFM) to H’FFFF FFFF and separation filter mask configuration register 1 (SFM1.CFM) to H’00FF FFFF.

CAUTION

If bits are set to 0 in the separation filter mask, in order to match with the pattern, the bits at the corresponding positions of the pattern must also be set to 0. Only when "the received frame data & the separation filter mask register (SFMi.CFM) = the separation filter pattern register (SFPi.FPs)", it is judged that the bits match with the pattern.

chanhvinh.com

Figure 45.22 shows separation filtering. The selected data from a received frame (Rx_Frame[63:0]) are masked by the common filter mask. This value is compared with all filter patterns. The AVB-DMAC selects the queue having the smallest queue number from among the matched filter patterns.

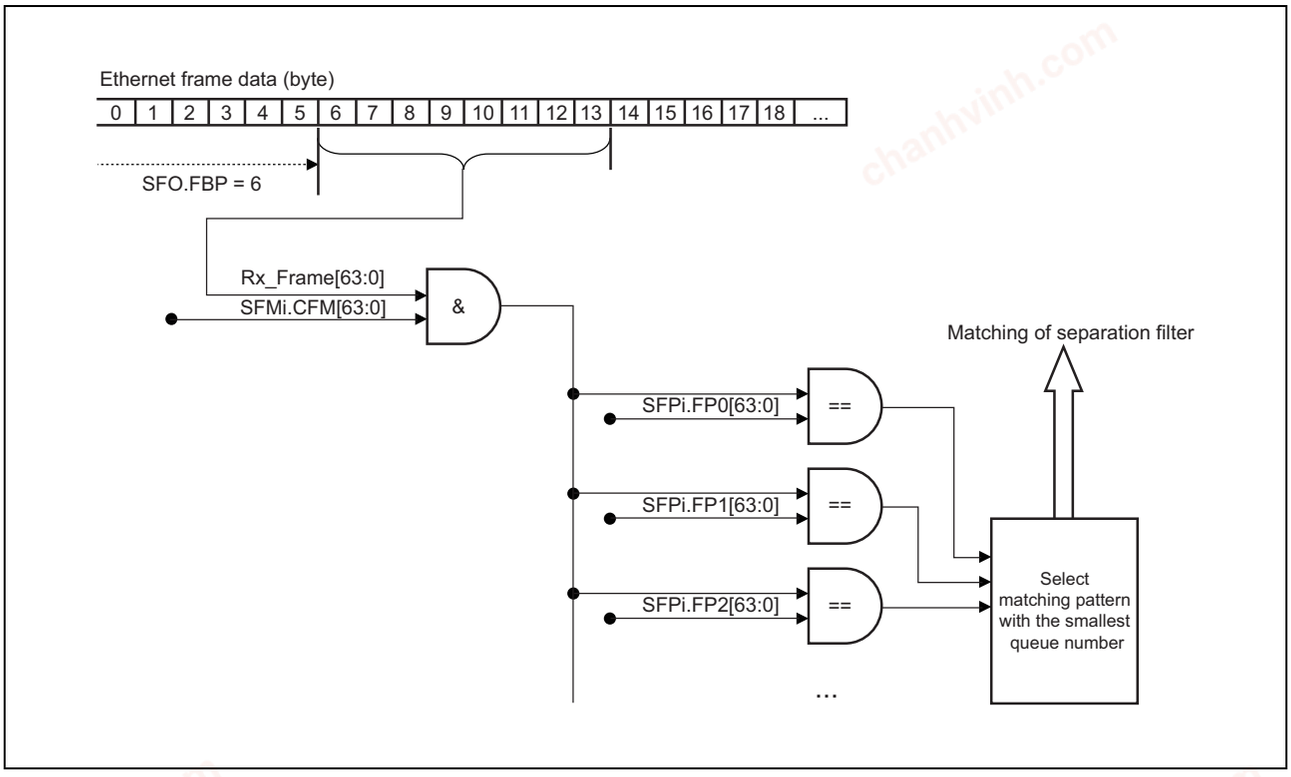


Figure 45.22 Overview of Separation Filtering (when SFO.FBP = 6)

(a) Restrictions on Number of Filters Used

The AVB-DMAC always checks all the separation filter patterns. When it is not necessary to use all the reception queues, providing a special pattern can prevent data from being stored in the unused reception queue.

Example 1:
Setting a value other than H'FFFF_FFFF_FFFF_FFFF to the separation filter mask register (SFMi.CFM) and setting H'FFFF_FFFF_FFFF_FFFF to the separation filter pattern register (SFPi.FPs) can prevent pattern matching in the unused reception queue.

Example 2:
Setting H'FFFF_FFFF_FFFF_FFFF to the separation filter mask register (SFMi.CFM) and setting the value identical to the pattern in reception queue 2 to the separation filter pattern register (SFPi.FPs) selects, upon a pattern matching, reception queue 2, which has a smaller queue number, thus preventing the unused reception queue from being used.

chanhvinh.com

(2) Separating Streams

The AVB-DMAC applies separation filtering to sort frames received in streams. An AVB network has a concept of “Talker” and “Listener”. A Talker is an end station that generates one or more streams. A Listener is an end station that has the role of being a sink for at least one stream. The various A/V streams are identified by 8-byte stream IDs.

The number of end stations within an ABV network and their roles differ with the application.

The stream ID is a specific pattern of the AVB network for identifying one stream. Figure 45.23 shows the bit allocation of bits in IEEE1722 Ethernet frames and stream ID fields.

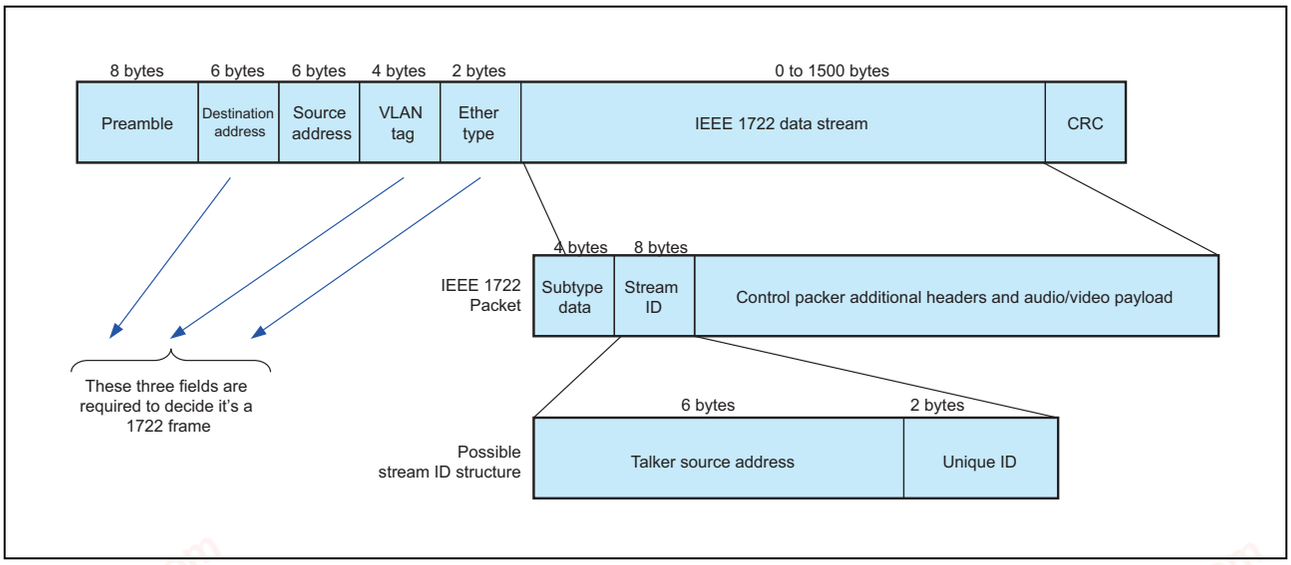


Figure 45.23 IEEE 1722 Frame Layout and Stream ID

The IEEE 1722 standard stipulates that the stream ID field starts from the 22nd byte (not counting the preamble). Accordingly, set the separation filter offset (SFO.FBP) to 22 in separations on IEEE 1722 streams. Set the separation filter mask (SFMi) and separation filter pattern (SFPi) in accord with the specification of the application in which the chip is being used.

Example: In the example of a stream ID shown in Figure 45.23, the representative application divides the field into the talker source address and the unique stream ID. The unique ID is used to differentiate between multiple streams from the same talker. Based on this, there are two settings for separation filter masking:

- To divide various streams into individual queues, set SFM0.CFM to H’FFFF FFFF and SFM1.CFM to H’FFFF FFFF.
- To divide streams for each talker into individual queues, set SFM0.CFM to H’FFFF FFFF and SFM1.CFM to H’0000 FFFF. This excludes the unique ID from the filter condition.

chanhvinh.com

45.3.4.2 Setting Up Reception Descriptors

For reception, the descriptor as described in section 45.3.3, Descriptors is used.

This section describes specific operations that are especially required in handling reception queues.

(1) Reception Descriptor Type

The type of a descriptor is defined by the descriptor type (DESCR.DT) field.

Table 45.76 shows the descriptor types used in reception. In the table, entries in the write-back column indicate how the AVB-DMAC changes the DESCR.DT field upon completion of descriptor processing.

Table 45.76 Descriptor Types in Reception

Descriptor Type (DESCR.DT)	Operation	Write-back
Frame Start (FSTART)	There is no storage space in a reception queue: The RIS2.QFFr bit indicates that queue r is full and the received frame is not stored. Descriptor processing proceeds again in response to further reception.	Not changed
Frame Middle (FMID)	Same as FSTART	Not changed
Frame End (FEND)	Same as FSTART	Not changed
Frame Single (FSINGLE)	Same as FSTART	Not changed
Link (LINK)	Processing proceeds to the descriptor specified by DESCR.DPTR.	LEMPTY
Fixed Link (LINKFIX)	Same as LINK	Not changed
End Of Set (EOS)	In the case of a divided frame (received frame being stored in FMID or FEND), storing of frames is stopped and the frame is lost. RIS2.QFFr indicates that the frame has been lost. If this happens at the start of a frame (received frame being stored in FSTART or FSINGLE), storing of frames starts from the next descriptor. In either case, processing shifts to the next descriptor in the chain.	EEMPTY
Frame Empty (FEMPTY)	Used to store received data. Up to DESCR.DS bytes are stored in the descriptor data area. For details, see section 45.3.4.3 (1) Storing Frame Data in the Descriptor Data Area.	FSTART, FMID, FEND, or FSINGLE
Frame Empty Incremental Start (FEMPTY_IS)	Used to store received data. The remaining frame data are all stored in the descriptor's data area. DESCR.DPTR indicates the base address of the incremental data area. For details, see section 45.3.4.3 (2) Incremental Data Areas.	FEND or FSINGLE
Frame Empty Incremental Continue (FEMPTY_IC)	Used to store received data. The remaining bytes of frame data are all stored in the descriptor's data area. DESCR.DPTR is undefined, but the address after processing of previous incremental data becomes the base address. For details, see section 45.3.4.3 (2) Incremental Data Areas.	FEND or FSINGLE
Frame Empty No Data storage (FEMPTY_ND)	Used to store received data. DESCR.DS bytes of spaces are secured in the reception FIFO but not stored. After processing, DESCR.DS is written back as 0. For details, see section 45.3.4.3 (2) Incremental Data Areas	FSTART, FMID, FEND or FSINGLE
Link Empty (LEMPTY)	Same as FSTART	Not changed
EOS Empty (EEMPTY)	Same as FSTART	Not changed

If a reception FIFO read error occurs when FEMPTY_ND is used, it is not reflected in DESCR.EI and queue error interrupt status bit (EIS.QEF).

Therefore, to detect a reception FIFO error also for unnecessary data, FEMPTY should be used.

chanhvinh.com

(2) Configuration of Reception Frame Data Descriptors

Figure 45.24 shows the configuration of descriptors for use with reception queues. The reception-specific fields are the same whether the descriptor is normal or extended. The reception-specific fields (DESCR.MSC, DESCR.PS, DESCR.EI, and DESCR.TR) are described in Table 45.77.

For the other fields and the descriptor types, see section 45.3.3.6, Descriptor Type.

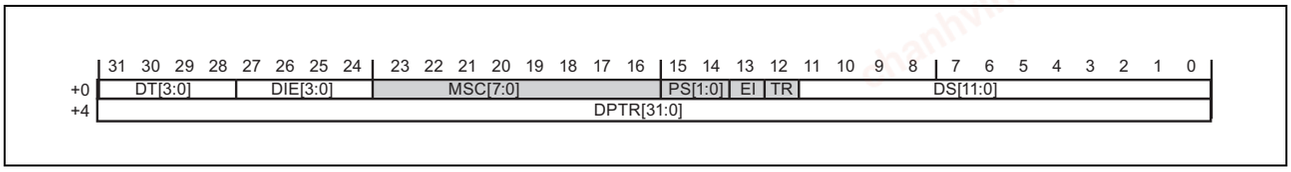


Figure 45.24 Configuration of Descriptor for a Received Frame

Table 45.77 Configuration of a Received Descriptor

Bit Name	Function
MSC	MAC Status Code These bits indicate errors in reception detected by the E-MAC. These bits are set to the same value between the frames divided by the descriptor. Details of the bits are as follows. MSC[7]: Received frame has a multicast address. MSC[6]: Reserved MSC[5]: Reserved MSC[4]: Received frame has residual bits. MSC[3]: Received frame is too long. MSC[2]: Received frame is too short MSC[1]: Error in frame reception MSC[0]: Received frame has a CRC error.
PS	Padding Selection These bits specify whether frame data are to be padded when stored in the incremental data area. Insertion of padding data is in accord with the settings in the receive padding configuration register (RPC). B’00: Padding is not to be inserted. B’01: Padding data may be inserted. This depends on the RPC settings. B'10: Setting prohibited. B'11: Setting prohibited.
EI	Error Indication This bit indicates the detection of an error in frame data while a frame was being stored. The bit is set to 1 for a descriptor in which an error has been detected. If the descriptor is for a divided frame, the frame data is discarded. 0: No error 1: Error is detected
TR	Truncation Indication This bit indicates whether frame data received from the E-MAC have been truncated before being stored in the on-chip RAM. These bits are set to the same value between the frames divided by the descriptor. 0: Data have not been truncated. 1: Data have been truncated.

CAUTION

The RCR.EFFS bit specifies whether or not frames with errors detected by the E-MAC are to be stored in the on-chip RAM. When the storing of error frames is disabled, error codes are not written to DESCR.MSC.

chanhvinh.com

45.3.4.3 Reception Processing

After initialization, the AVB-DMAC is able to select the proper reception queue and store received frames in the data area in the on-chip RAM as indicated by the descriptor. The AVB-DMAC continues to store received data in the on-chip RAM as long as space is available for descriptors and data areas.

Received frames are classified and stored in the reception FIFO in accord with the algorithm described in section 45.3.4.1 (1) Separation Filtering. A frame will already have been sorted by separation filtering, truncated, or discarded based on reception by the MAC before being stored in the reception FIFO. The following data are stored in the reception FIFO.

- MAC status of received frames
- Length of received frames
- Time stamp of received frames
- Target reception queue
- Received frame data

If the reception FIFO contains even one frame, the scheduler allocates storing in the reception queue (see section 45.3.2.2, Scheduling Reception and Transmission).

If there is even one empty descriptor in a queue for which reception has started, the storage of frame data starts. Received frames for a queue that is already full (there is no empty frame descriptor or the unread frame counter stop level has been reached) are discarded from the reception FIFO. This ensures that one queue being full does not prevent the processing for data in the other queues.

(1) Storing Frame Data in the Descriptor Data Area

Frame data for storage are assumed to be in either of the two patterns described below.

- The data for an entire frame will fit in the descriptor data area.
  - In this case, the descriptor type (DESCR.DT) is FSINGLE.
- Frame data is bigger than the descriptor, so the frame data is divided and stored in the descriptor data area.
  - In this case, FSTART is written to the descriptor type (DESCR.DT) bits of the first of the frame data to arrive and FMID and FEND are written to the type bits of descriptors for subsequent data.

The descriptor type is updated by the AVB-DMAC in the last step of descriptor processing, so software can always access the descriptor written-back to DESCR.DT.

Software can write FEMPTYxxx directly to the descriptor type field after processing the stored frame data. Do not change the descriptor or any part of the descriptor data area after FEMPTYxxx is written to DESCR.DT.

chanhvinh.com

(a) Single Frame

For a frame with an FSINGLE descriptor, all data for the frame are held at the position defined by DESCR.DPTR. DESCR.DS indicates the length of the received frame.

If DESCR.DS is bigger than the actual size of a received frame, the FEMPTY or FEMPTY_ND descriptor is stored in place of the FSINGLE descriptor.

Also, the FEMPTY_IS and FEMPTY_IC descriptors, which always hold the full frame data, are stored in place of the FSINGLE descriptor.

(b) Divided Frames

Divided frames can be processed in the same way as a single frame. A frame stored with divided descriptors must be recombined before use. DESCR.EI and DESCR.TS are only valid in the last descriptor of the sequence for a divided frame.

CAUTION

If the data area size setting in DESCR.DS is not a multiple of four bytes, only the number of bytes set in DESCR.DS is fetched from the reception FIFO and the remaining bytes are used by the next descriptor.

After a received frame is divided into different descriptors, each frame data is processed separately, and the descriptor type is assigned by software after processing. Accordingly, an error frame (FEMPTYxxx instead of FMID or FEND) may exist while a descriptor chain is being processed. In such a case, the CPU must postpone processing of the error frame until the corresponding descriptor has been processed.

(c) No Data

The application specification may lead to some types of received frames being unimportant (for example, when the application only requires stream data from the Ethernet frames). Storing frames in divided form makes separating out the unnecessary parts of Ethernet frames possible.

If part of a divided frame is not required, use the FEMPTY_ND descriptor so that unnecessary data is not stored in the on-chip RAM.

After processing an FEMPTY_ND descriptor, the AVB-DMAC sets DESCR.DS to 0. DESCR.DS = 0 is for the specific flag indicating that the FEMPTY_ND descriptor has been processed.

chanhvinh.com

(2) Incremental Data Areas

Secure space for storing received data in the on-chip RAM. Even when all descriptor data areas of a chain are placed in the contiguous area of the on-chip RAM area, a received frame being shorter than the descriptor data area will lead to an empty space. Figure 45.25 shows an example of settings and the memory map.

In some cases, contiguous data arrangement may be effective (e.g. when received data are to be processed other than by hardware as the A/V codec module). When the length of frames differs for each received frame (e.g. when payloads vary between having one or two A/V packages), the use of a pointer in the descriptor produces empty spaces in the data area. This necessitates additional processing such as copying the frame data to remove the empty spaces.

Accordingly, and to prevent copying data, the AVB-DMAC supports an “incremental data area” function.

When incremental data areas are in use, different descriptors use a contiguous data area. The first descriptor (FEMPTY_IS) from the top defines the base address of the incremental data area and is used to, together with the subsequent descriptor (FEMPTY_IC) within the descriptor chain, hold received data. Figure 45.26 shows an example of settings and the memory map.

When an incremental data area is used, it is also possible to divide a frame into various descriptors in a way that reflects its structure (e.g. one descriptor for the Ethernet header and one for the data payload).

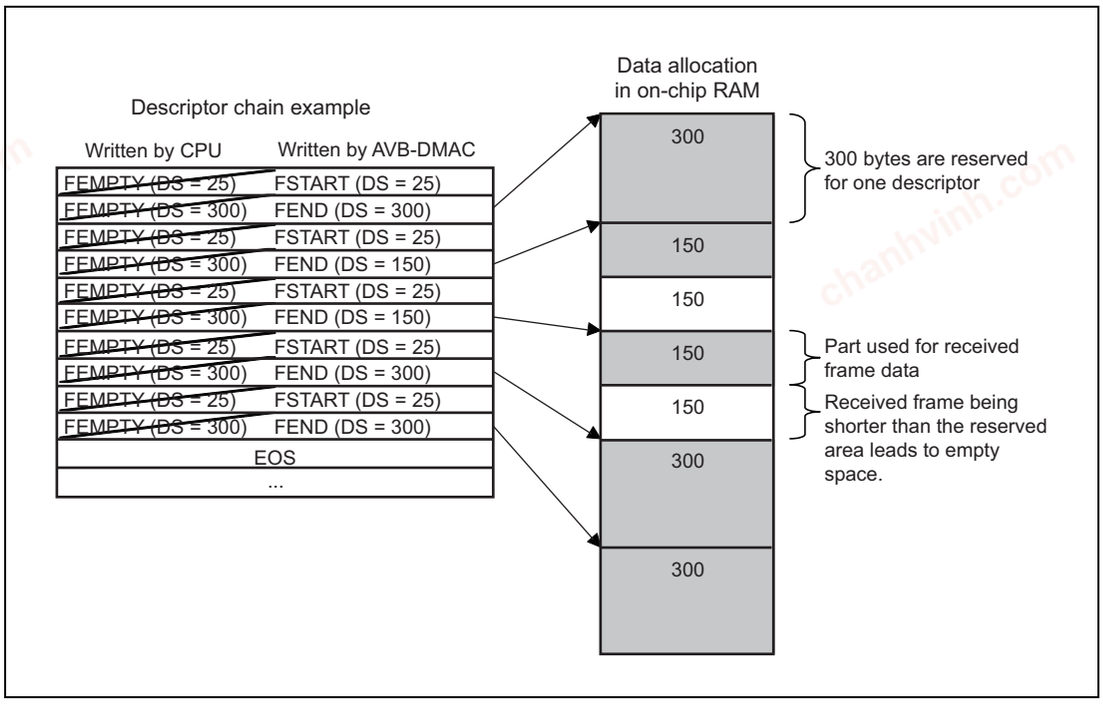


Figure 45.25 When Individual Descriptor Data Areas are Used

chanhvinh.com

Figure 45.25 and Figure 45.26 show how control of the data storage areas by a descriptor chain varies according to whether incremental data areas are not in use or in use. The chains are configured for storing received frames consisting of a 25-byte header (in this example, stored in the on-chip RAM area outside the drawing range) and a 150- or 300-byte payload (whether one or two 150-byte data depending on the data source).

In Figure 45.25, an EOS descriptor is added as an example of a re-synchronization point. If a frame larger than 325 bytes is received, the frame will be divided into three descriptors, meaning that synchronization of the header and data sequences is lost. However, a frame is not divided across an EOS descriptor, so adding EOS descriptors can prevent the header and data sequences losing synchronization. While an incremental data area is in use, EOS descriptors are not required because the incremental descriptors always hold all data being processed.

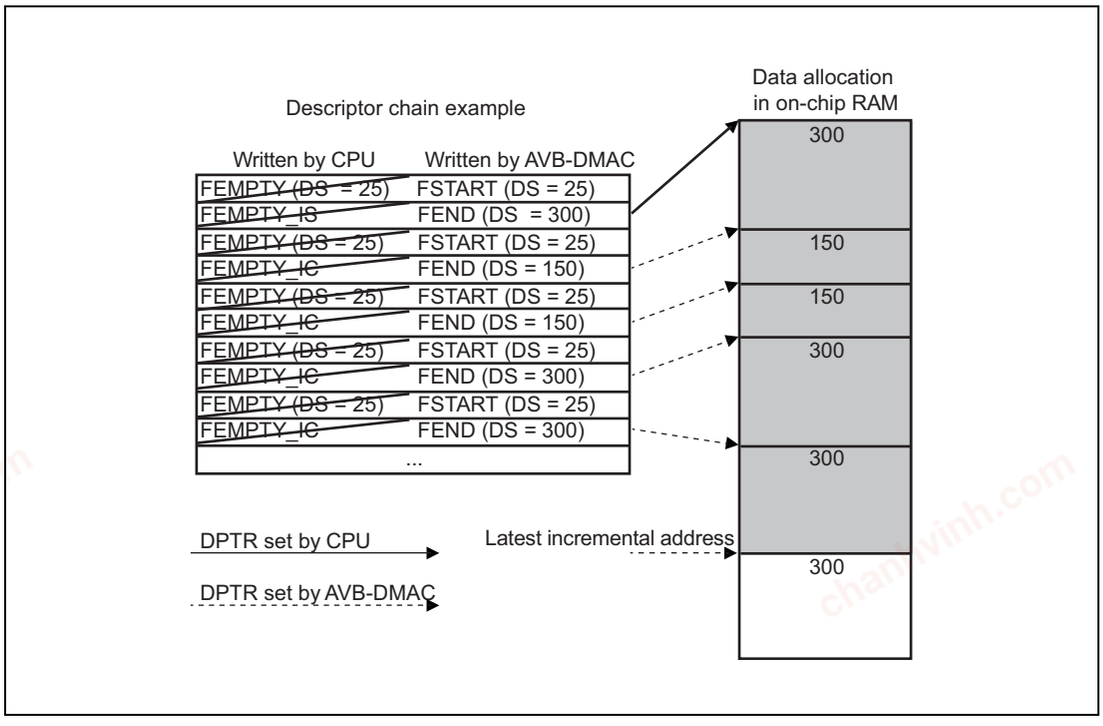


Figure 45.26 When Incremental Data Area is Used

As Figure 45.26 shows, when data are stored in an incremental data area, the descriptor pointers in the FEMPTY_IC descriptors (DESCR.DPTR) are updated. Accordingly, the FEND or FSINGLE descriptor written-back by the AVB-DMAC after processing is in the same format as the FEMPTY descriptor after processing.

Software captures contiguous received data, which has no empty spaces, from an incremental data area. The incremental data area must be aligned with a 4-byte boundary. Therefore, when the amount of received data for storage in an incremental data area is not a multiple of four bytes, from one to three bytes of empty space is placed at the end of the incremental data area. DESCR.DS can be read to check for such empty spaces.

For the normal descriptors (FEMPTY and FEMPTY_ND), controlling the amount of received data due to DESCR.DR is possible, but for the incremental descriptors (FEMPTY_IS and FEMPTY_IC), controlling the amount of received data to be stored is not possible. All received data are always stored in an incremental descriptor.

chanhvinh.com

(a) Setting Up an Incremental Data Area

If a descriptor chain has N descriptors (one FEMPTY_IS and N-1 FEMPTY_IC), an area for storing N times of the maximum received data as the incremental data area must be secured.

As Figure 45.26 shows, DESCR.DPTR of an FEMPTY_IS descriptor indicates the base address of the incremental data area. The subsequent FEMPTY_IC descriptor in the chain indicates that data must be stored in the incremental data area.

(b) Processing an Incremental Data Area Based on Descriptors

Since data processing by the CPU is the same regardless of how the AVB-DMAC stores the data, data stored in an incremental data area do not require any special handling.

(c) Padding

Use padding for reception of frame data that are not aligned correctly. Padding can be set individually for each descriptor. Accordingly, in the reception of divided frames, padding can be set to only those frames that require it (e.g. A/V payload data.)

Padding can also be used to effectively use an incremental data area (e.g. to prevent inefficient access by aligning received data with 32-byte boundaries.

Padding can only be used in an incremental data area.

The value H'0000 0000 is always used in padding.

Padding is the addition of the number of words (from one to seven 32-bit words) set in the stored padding counter bit in the receive padding configuration register (RPC.PCNT). This padding is repeatedly inserted for each received data count set to the stored data counter bit (RPC.DCN) (from one to 255 32-bit words). When the stored data counter (RPC.DCNT) reaches 0, however, padding is not repeated.

The first word of padding is always inserted at the position specified by DESCR.DPTR. When divided frames are in use, a padding word can be inserted in accord with each descriptor (e.g. the first descriptor handles a 42-byte header data and the second descriptor holds padded payload data in an incremental data area).

The next figure shows a general example of how padding is inserted and an example of setting up padding. In the figure, A indicates frame data received from the E-MAC, while B indicates frame data stored in the descriptor data area (32-bit word units).

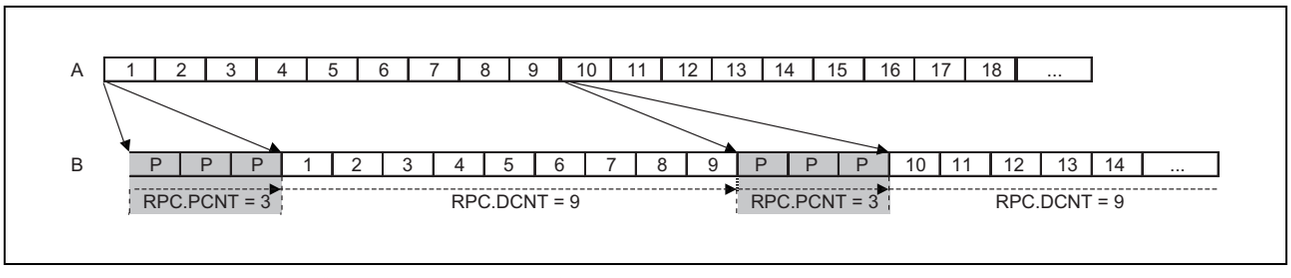


Figure 45.27 Example of a Padding Setting

Both padding and received frame data are counted in the descriptor size (DESCR.DS).

chanhvinh.com

(3) Support for Reception Time Stamps

Capturing reception time stamps is essential for IEEE 802.1AS time synchronization. Other types of received frames may also require that a reception time stamp be appended; this depends on the application. The AVB-DMAC supplies reception time stamps based on the gPTP timer by storing time stamps that have been captured when the start frame delimiter (SFD) for a received frame in the last frame data descriptor (FEND or FSINGLE). For the gPTP timer, see section 45.3.7.1, gPTP Timer.

When time stamps are to be stored, use extended descriptors for the entire reception queue. Furthermore, time stamps are always stored for reception queue 1 (network control). Time stamps for reception queue 0 (best effort) and reception queue r ($r \geq 2$; for stream data) can be selected by the time stamp enable bits in the receive configuration register (RCR.ETS0 or RCR.ETS2).

chanhvinh.com

45.3.4.4 Unread Frame Counters

Each reception queue has an unread frame counter (UFCVi). Use the unread frame counter configuration bits in the receive queue configuration register (RQCi.UFCCr) to select from among the four stop levels for each unread frame counter. The 0 setting disables the stop functions. For how to set this up, see Figure 45.28.

Operations of the AVB-DMAC (hardware) and CPU (software) drive an unread frame counter (UFC) in the following ways.

- The hardware indicates that it has added a new frame to the descriptor chain for the queue (this increments the counter).
- Software indicates how many frames from the descriptor chain it has processed by writing to the corresponding bits of the unread frame counter decrement register for the queue (this decrements the register by the number written).

The unread frame counter is based on the number of frames stored in the on-chip RAM and is only incremented by one even when a received frame is divided into different descriptors. Data failure in a descriptor chain requires care because this may fail in synchronization of the unread frame counter as described in section 45.3.4.4 (1) Loss of Unread Frame (UFC) Synchronization.

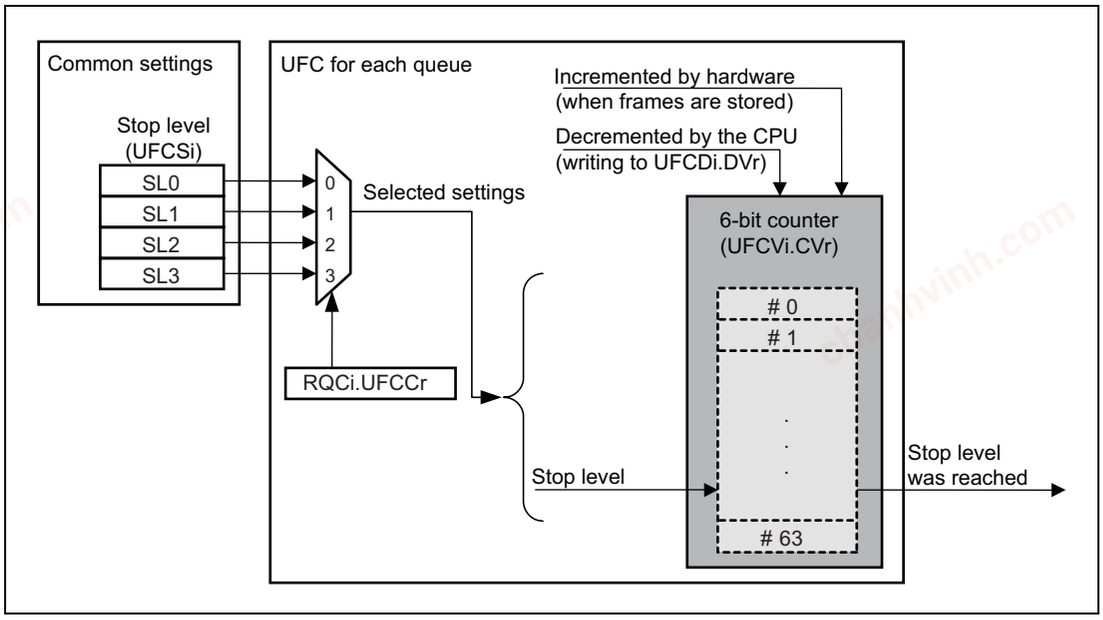


Figure 45.28 Overview of an Unread Frame Counter

Unless synchronization of hardware and software is lost, the current unread frame counter value (UFCVi.CVr) indicates the number of unread frames in the queue.

When the stop level has been reached, received frames are not stored in the descriptor chain. Setting 0 as the stop level disables the stop function. Otherwise, further received frames for the queue are discarded once its unread frame counter reaches the stop level. Activation of the unread frame counter stop function is indicating by setting of the receive queue full interrupt flag in the receive interrupt status register 2 (RIS2.QFFr).

Set the unread frame counter stop level configuration register (UFCS) for each reception queue that will use the unread frame counter function while the current operating mode is configuration mode.

chanhvinh.com

(1) Loss of Unread Frame (UFC) Synchronization

The unread frame counters do not recognize failure to store a frame in the on-chip RAM. In other words, the AVB-DMAC increments the counter for a queue each time it captures a frame for that queue from the reception FIFO whether or not it succeeds in storing the frame normally in the descriptor chain.

In general, synchronization of hardware and software may be lost under the following conditions.

- An unread frame counter reaching its maximum value
  When the value of a counter in an unread frame counter register i (UFCVi) (i = 0 to 4) reaches 63, synchronization for the corresponding queue can be lost.
  It can only be judged that a loss of synchronization has not occurred when the stop level is set to 63.
- A queue not having enough space for a descriptor or the associated data
  In this case, the corresponding receive queue full interrupt flag (RIS2.QFFr) in receive interrupt status register 2 (RIS2) is set.
  If an unread frame counter reaches its stop level, the receive queue full interrupt flag (RIS2.QFFr) in the receive interrupt status register 2 (RIS2) is set.
- A problem occurring during access to memory

The result of a failure in synchronization is the unread frame counter indicating that the corresponding descriptor chain contains more available frames than it actually does. To retrieve the correct starting point for operations, use the descriptor base address load request (DLR.LBAq) for the given queue.

chanhvinh.com

45.3.5 Transmission Control

Areas in the on-chip RAM for storing transmission descriptors must also be secured (for descriptors, see section 45.3.3, Descriptors).

The AVB-DMAC fetches data from the on-chip RAM in accord with the procedure the descriptor describes. The descriptor also retains tag information once the frame has been fetched for transmission. The tag information is used to maintain the relationships between state information and time stamps for the software and the AVB-DMAC. The status and time stamp information for transmitted frames remains accessible after their transmission is completed.

45.3.5.1 Transmission Modes

The AVB-DMAC has two modes of transmission.

- AVB transmission mode
  This mode is selected by the priority level setting for the transmission queue in the transmit configuration register (setting of the TGC.TQP[1:0] bits) being B'01 or B'11.
- Non-AVB transmission mode
  This mode is selected by the priority level setting for the transmission queue in the transmit configuration register (setting of the TGC.TQP[1:0] bits) being B'00.

(1) AVB Transmission Mode

AVB transmission supports the control of traffic through the output port to implement various traffic classes.

(a) Support for Traffic Classes and Associated Priority

When transmission is in AVB transmission mode, streams of traffic are transmitted in accord with the part of the AVB specification called Forwarding and Queuing for Time Sensitive Streams (FQTSS; for details on this, see the IEEE 802.1Q standard).

In the AVB specification, at least one queue for a reserving stream under the Stream Reservation Protocol (SR stream) and at least one queue for a non-SR stream are present, and the queues for reserved traffic according to the SRP have highest priority.

The AVB-DMAC supports four traffic classes: SR class A, SR class B, network control (NC) traffic (gPTP frames), and best effort (BE) traffic. Allocating a specific queue to network control (NC) frames ensures the control of synchronization.

The AVB-DMAC realizes compliance with the AVB standards by handling queues with the following architecture (in terms of traffic classes).

- Four transmission queues (Q3, Q2, Q1, and Q0) are available.
- Q3 and Q2 are for SR streams (one each for class A and class B).
- Q1 is for low-bandwidth network control (NC) traffic (gPTP frames)
- Q0 is for other types of traffic (MSRPDU*1, MVRPDU*2, best effort (BE), etc.)

CAUTIONS

1. MSRPDU: Multiple Stream Registration Protocol Data Unit
2. MVRPDU: Multiple VLAN Registration Protocol Data Unit

chanhvinh.com

Fetching from queues proceeds in order of priority of the above traffic types. Three systems of priority are available through the setting of the transmit queue priority bits in the transmit configuration register (TGC.TQP[1:0]). In the default priority scheme, which is called AVB mode 1 (selected by TGC.TQP[1:0] = B'01), operation of the AVB-DMAC is fully in accord with the AVB specification. AVB mode 2 (transmit queue priority bits (TGC.TQP[1:0] = B'11) is an alternative priority scheme and varies from the AVB specification. Using this scheme thus requires more care.

Table 45.78 Default and Alternative Priority Orders in AVB Transmission Mode

Priority Schemes (AVB Mode)	Priority Order of Queues
AVB mode 1	Q3 (SR class A) > Q2 (SR class B) > Q1 (NC) > Q0 (BE)
AVB mode 2	Q1 (NC) > Q3 (SR class A) > Q2 (SR class B) > Q0 (BE)

(b) Transmission Selecting Algorithm and CBS

The algorithm the AVB-DMAC applies to select frames for transmission is in accord with the IEEE 802.1Q standard. For AVB mode, the CBS (credit-based shaping) algorithm is applied to the class A and class B SR queues (Q3 and Q2). Use of the CBS enables correct handling of the priorities of transmission from the SR queues. For the CBS algorithm, see section 45.3.6, CBS (Credit-Based Shaping).

When the following conditions are both satisfied, transmission from an SR queue (Q3 or Q2) proceeds at the specified time.

- The queue contains at least one frame ready for transmission.
- The queue has credit.
- Unless an SR queue satisfies the above conditions, a higher priority queue is not present (not ready for transmission).

A non-SR queue (Q1 or Q0) is selected if the conditions below both hold.

- The queue contains at least one frame ready for transmission.
- As well as the above condition, a higher priority queue is not present (not ready for transmission).

chanhvinh.com

Figure 45.29 and Figure 45.30 are flowcharts of selection for transmission in AVB mode 1 and AVB mode 2.

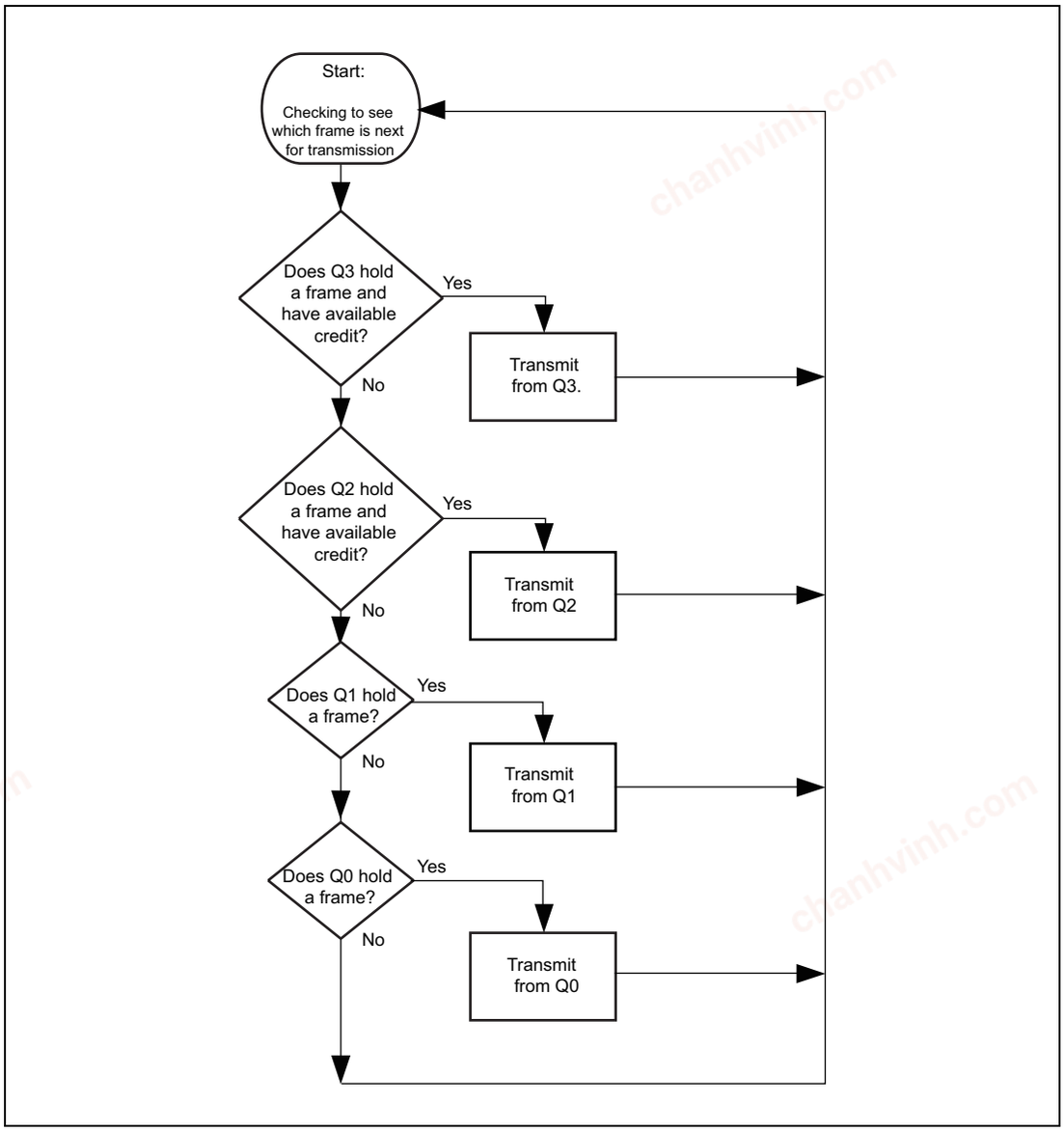


Figure 45.29 Flow of Selection for Transmission in AVB Mode 1

chanhvinh.com

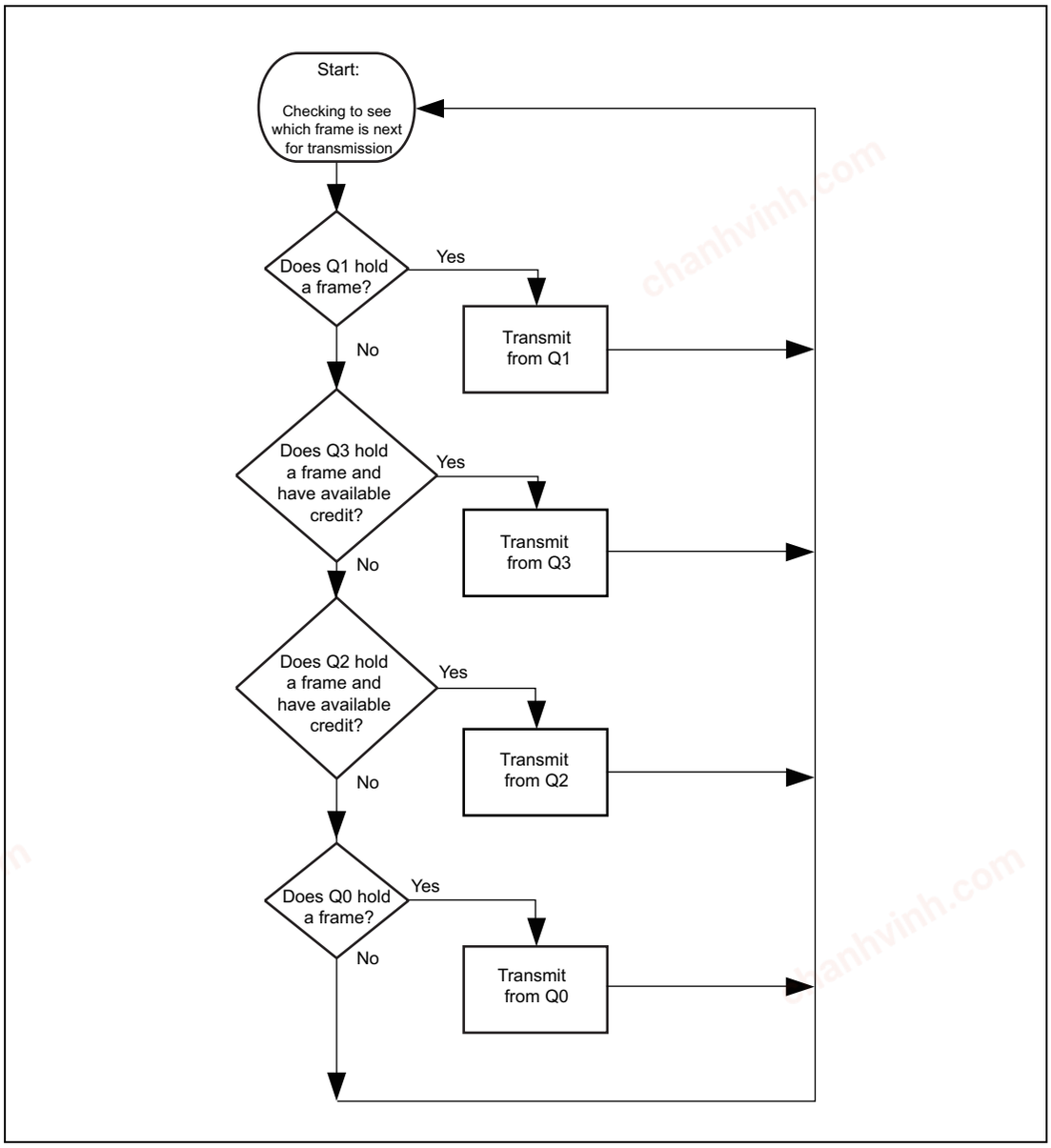


Figure 45.30 Flow of Selection for Transmission in AVB Mode 2

chanhvinh.com

(2) Non-AVB Transmission Mode

In non-AVB transmission mode, an absolute priority scheme is used. The SR class is not supported and the CBS algorithm is not used.

In non-AVB transmission mode (when the transmit queue priority bits in the transmit configuration register (TGC.TQP[1:0]) are B'00), data is fetched for transmission in a strict order of priority (Q3 > Q2 > Q1 > Q0).

Figure 45.31 shows the flow of selection in non-AVB transmission mode.

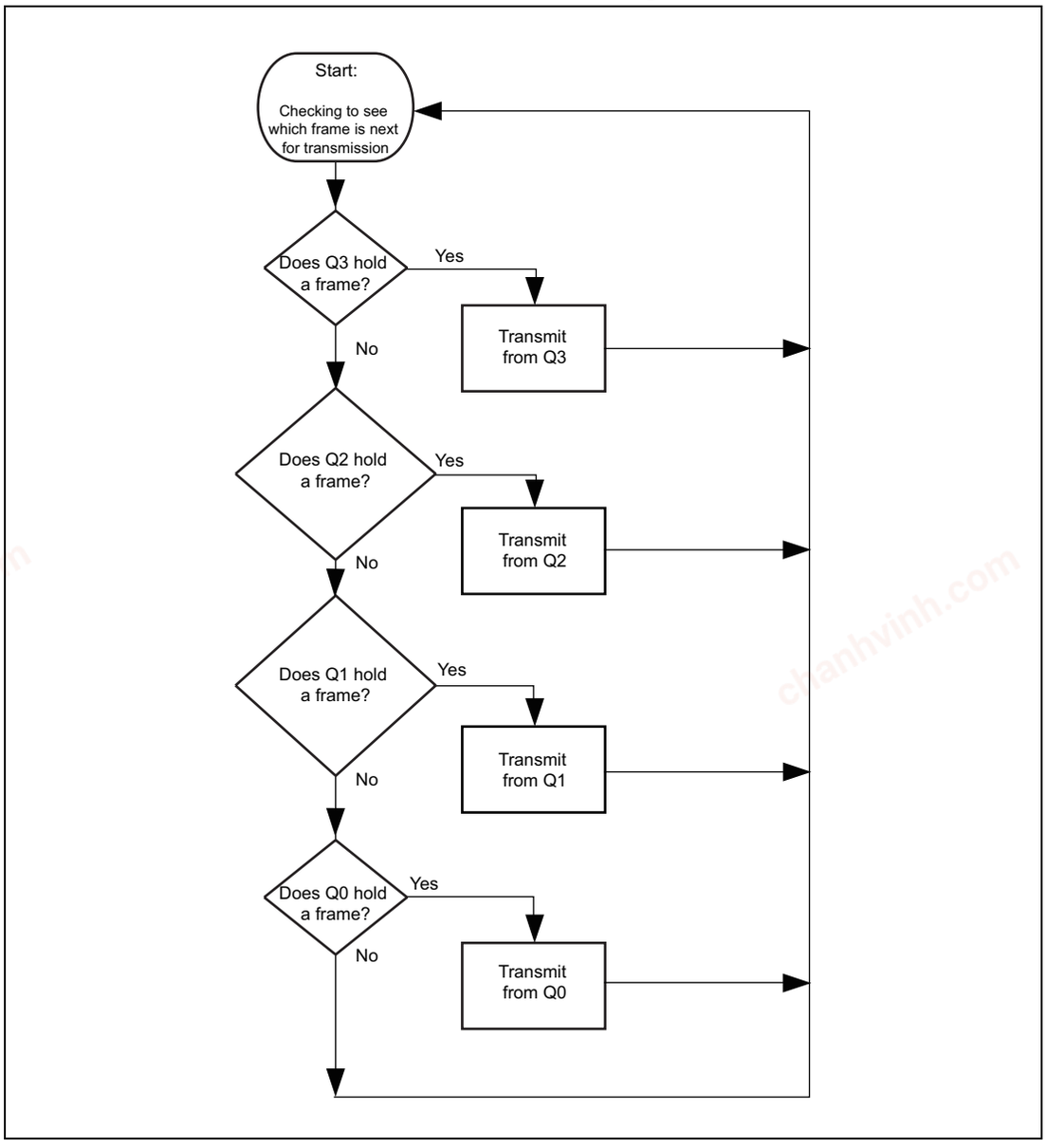


Figure 45.31 Flow of Selection for Transmission in Non-AVB Mode

chanhvinh.com

(3) Setting the Size of the Transmission FIFO

The transmission FIFO is made up of 124 clusters. Each cluster can hold up to 128 bytes.

The size of the part of the transmission FIFO for use by each of the four transmission queues can be set by the corresponding transmit queue configuration q bits in the transmit control register (TGC.TBDq). The maximum number of clusters required can be determined from the maximum length of frames for transmission from the queue q.

Even if queue t already contains data of the frame size specified by TGC.TBDt, the AVB-DMAC fetches the next frame to queue t after the E-MAC starts transfer data from queue t. Therefore, a frame data area of the total of maximum frame sizes set for each transmission queue plus an area for 1 frame must be secured for a cluster.

General Usage Examples:

Q0: Frames containing up to 1500 bytes → 1500/128 = 11.7 → 12 clusters

Q1: Frames containing up to 1024 bytes → 1024/128 = 8.0 → 8 clusters

Q3: Frames containing up to 1996 bytes → 1996/128 = 15.6 → 16 clusters

Q4: Frames containing up to 1996 bytes → 1996/128 = 15.6 → 16 clusters

When the depth of all transmission queues is 2, only the following number of clusters is required.

2 * (12 + 8 + 16 + 16) + 16 = 2 * 52 + 16 = 120

Adjust the frame length of each queue so that the number of clusters to be used is no greater than 124.

chanhvinh.com

45.3.5.2 Setting Up Transmission Descriptors

(1) Transmission Descriptor Type

The type of a descriptor is defined by the descriptor type (DESCR.DT) field.

Table 45.79 shows the descriptor types used in transmission. In the table, entries in the write-back column indicate how the AVB-DMAC changes the DESCR.DT field upon completion of descriptor processing.

Table 45.79 Descriptor Types in Transmission

Descriptor Type (DESCR.DT)	Operation	Write-back
Frame Start (FSTART)	The AVB-DMAC fetches the first of the data for the divided frame and proceeds to the next descriptor.	FEMPTY
Frame Middle (FMID)	The AVB-DMAC fetches the second or subsequent data for the divided frame and proceeds to the next descriptor.	FEMPTY
Frame End (FEND)	The AVB-DMAC fetches the last of the data for the divided frame. When the frame of data that has been fetched to the transmission FIFO is ready for transmission by the E-MAC, the AVB-DMAC proceeds to the next descriptor.	FEMPTY
Frame Single (FSINGLE)	The AVB-DMAC fetches the frame of data. When the frame of data that has been fetched to the transmission FIFO is ready for transmission by the E-MAC, the AVB-DMAC proceeds to the next descriptor.	FEMPTY
Link (LINK)	Processing proceeds to the descriptor specified by DESCR.DPTR.	LEMPTY
Fixed Link (LINKFIX)	Same as LINK	Not changed
End Of Set (EOS)	Clears the transmit start request bit (TCCR.TSRQt), which stops transmission queue. When the TCCR.TSRQt is again set to 1 (a new transmission start request is issued), processing proceeds to the next descriptor.	EEMPTY
Frame Empty (FEMPTY)	Clears the transmit start request bit (TCCR.TSRQt), which stops transmission queue. When the TCCR.TSRQt is again set to 1 (a new transmission start request is issued), processes this descriptor again.	Not changed
Link Empty (LEMPTY)	Same as FEMPTY	Not changed
EOS Empty (EEMPTY)	Same as FEMPTY	Not changed

chanhvinh.com

(2) Configuration of Transmission Frame Data Descriptors

Figure 45.32 shows the configuration of descriptors for use with transmission queues. The transmission-specific fields (DESCR.TSR and DESCR.TAG) are described in Table 45.80.

For the other fields and the descriptor types, see section 45.3.3.6, Descriptor Type.

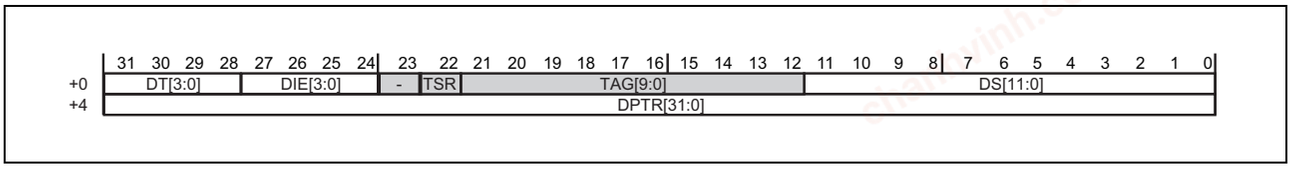


Figure 45.32 Configuration of Descriptor for a Transmitted Frame

Table 45.80 Configuration of a Transmission Descriptor

Bit Name	Function
TSR	Time Stamp Store Request This bit specifies whether the transmission time stamp together with tag information is to be stored within the EthernetAVB module. 0: The time stamp status FIFO within the EthernetAVB module does not retain a transmission time stamp. 1: The time stamp status FIFO within the EthernetAVB module retains a transmission time stamp. Only control this bit while the current DESCR.DT is FEND or FSINGLE.
TAG	Frame Tag This TAG field is used to associate each frame data status with a time stamp. Frame TAG is not required but is recommended. Only control this bit while the current DESCR.DT is FEND or FSINGLE.

For the time stamp FIFO function, see section 45.3.5.4, Time Stamping in Transmission.

chanhvinh.com

45.3.5.3 Transmission

(1) Transmitting Frames

Setting the transmit start request bit in the transmit configuration control register (TCCR.TSRQt) starts the transfer of frames from the corresponding transmission queue.

The descriptor in the current descriptor address for the queue (CDARq.CDA) is read first.

If this descriptor is a descriptor for frame transmission (FSINGLE, etc.), the AVB-DMAC fetches the frame data from the data area indicated by the descriptor, writes FEMPTY back to the descriptor type (DESCR.DT) bits to indicate completion of this processing, then proceeds to processing of the next descriptor.

If the descriptor is not for transmission, processing is as dictated by the given descriptor (for these descriptors, see the descriptions in section 45.3.3, Descriptors).

If a base address load request is issued for a descriptor chain while it is being processed (by setting 1 in the LBAq bit for transmission queue q that is currently being processed in the descriptor base address load request register, DLR), processing proceeds to the new descriptor chain (descriptor base table address bits (DBAT.TA) + 8*q). Loading the base address does not interrupt frame fetching, but note that frames that have not been fetched from the old chain remain where they are.

Figure 45.33 shows descriptor processing during transmission. “Fxxx” in the figure indicates the frame data descriptors (FSTART, FMID, FEND, FSINGLE).

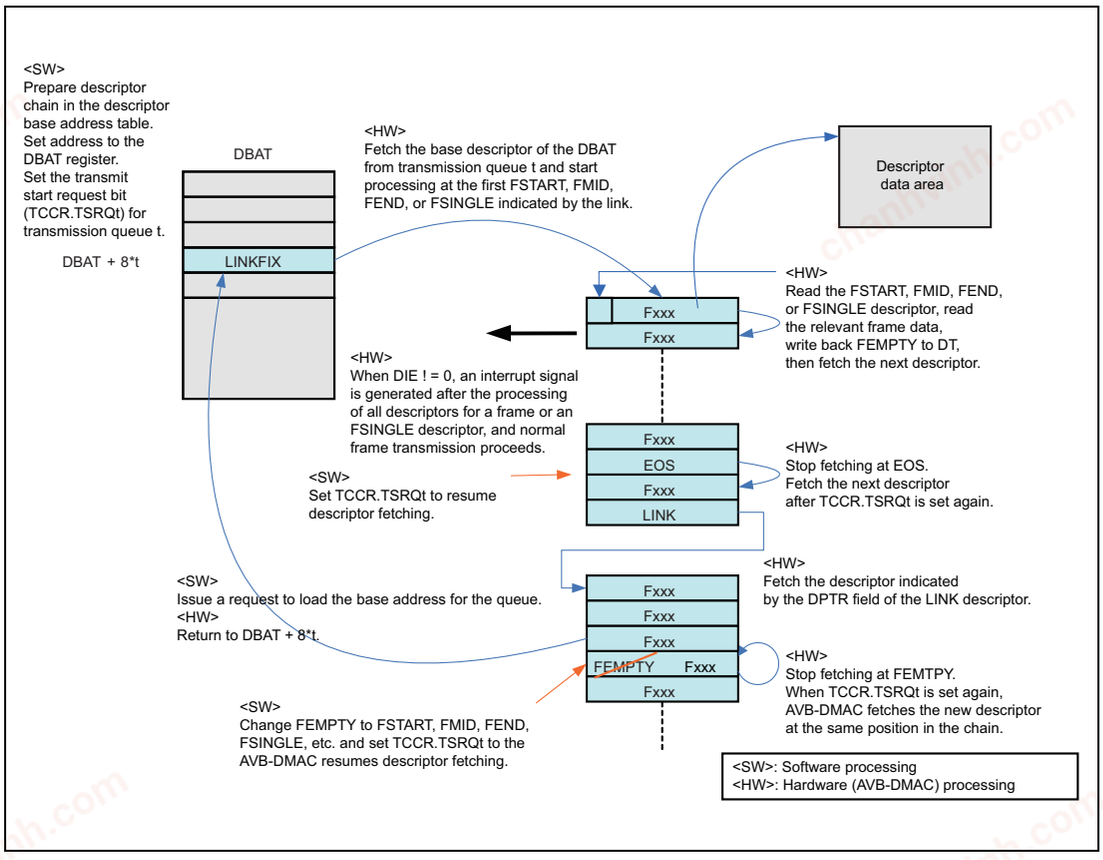


Figure 45.33 Descriptor Processing During Transmission

chanhvinh.com

(2) Examples of Descriptor Usage

(a) Immediate Frame Transmission

Immediate frame transmission is a method in which fetching by the AVB-DMAC starts whenever software adds data to a queue. FEMPTY descriptors are used as stop points to keep the hardware and software in synchronization.

As preparation in transmission, software creates descriptor chains that have FEMTPY descriptors.

Figure 45.34 shows the flow for subsequent process.

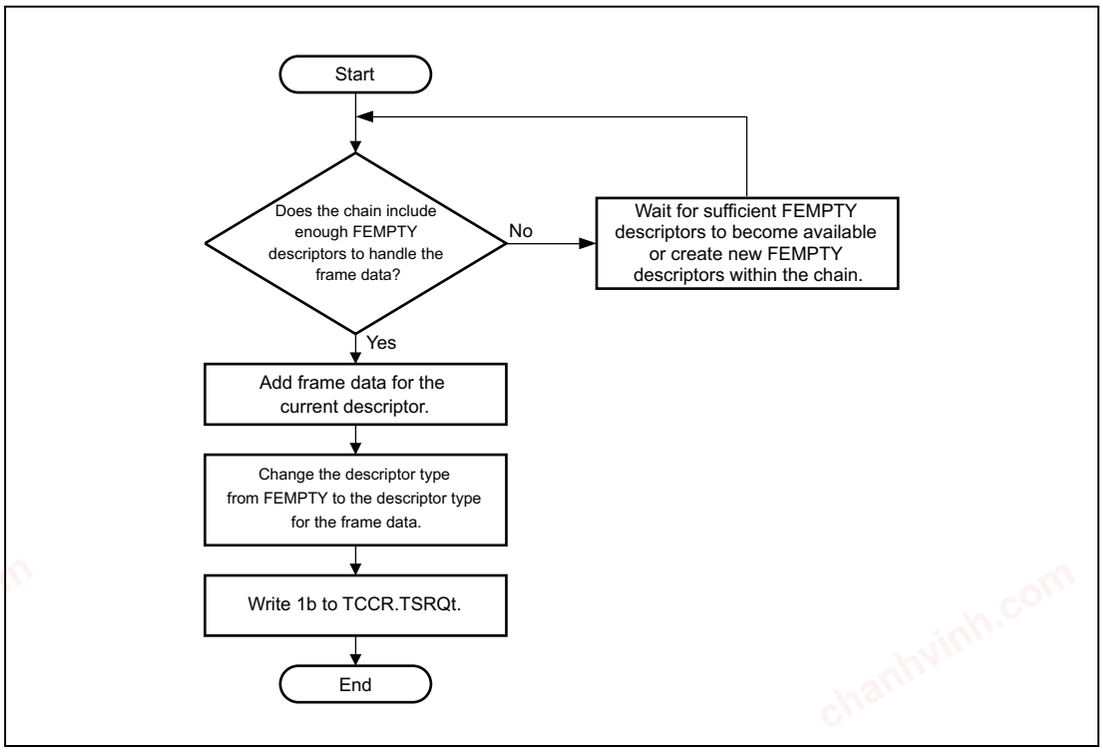


Figure 45.34 Software Flow for Immediate Frame Transmission

When processing a divided frame, change the types of descriptors in the backward direction, i.e., from FEND to FSTART. This guarantees all the descriptors including frame data to be ready when the AVB-DMAC starts fetching divided frames.

Figure 45.35 shows software and AVB-DMAC operations for divided frame transmission. In the figure, software and hardware independently perform processing. This allows software to prepare the descriptor for frame n + 1 even when the AVB-DMAC is currently fetching frame n.

chanhvinh.com

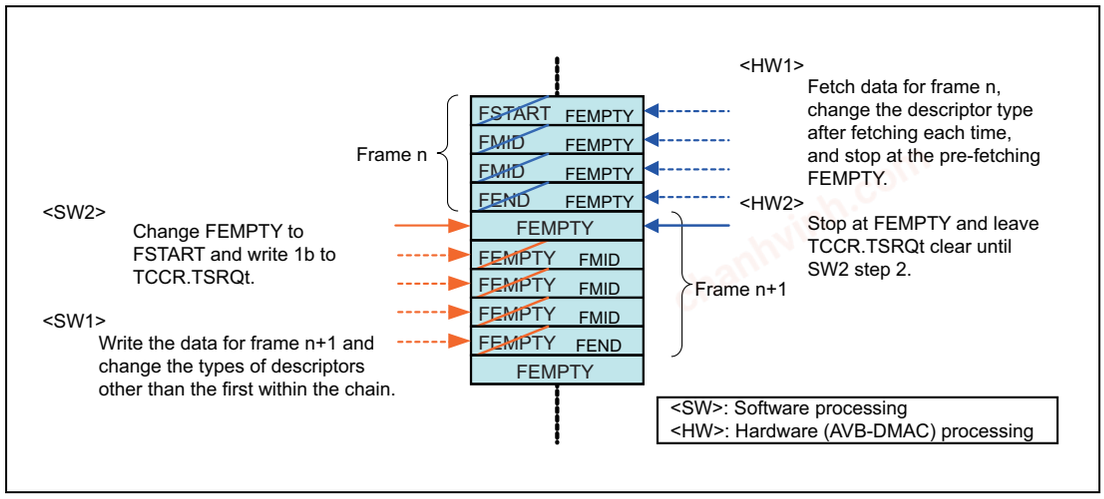


Figure 45.35 Software and AVB-DMAC Operation Examples for Immediate Frame Transmission (divided frames)

chanhvinh.com

(b) Frame Set Transmission with Switching of the Active Descriptor Chain

This method is used when data are transmitted with a delay for software control to secure bandwidth or for other reasons, rather than immediately transmitted. EOS descriptors are used for the stop points.

As preparation in transmission, software creates descriptor chains that have FEMTPY descriptors.

Figure 45.36 shows the flow for subsequent process.

In the first task, software creates and adds a transmission frame to the descriptor chain, which is followed by EOS descriptor insertion.
In the second task, software processes the transmission trigger for the transmission descriptor prepared. This guarantees transmission of a frame set per trigger.
No particular operation is necessary for synchronization between two tasks. If not enough frame sets are prepared when the transmit start request bit in the transmit configuration control register (TCCR.TSRQt) is set, only the prepared frame sets are transmitted. If nothing is prepared in the queue, only clearance of a transmission request is performed.

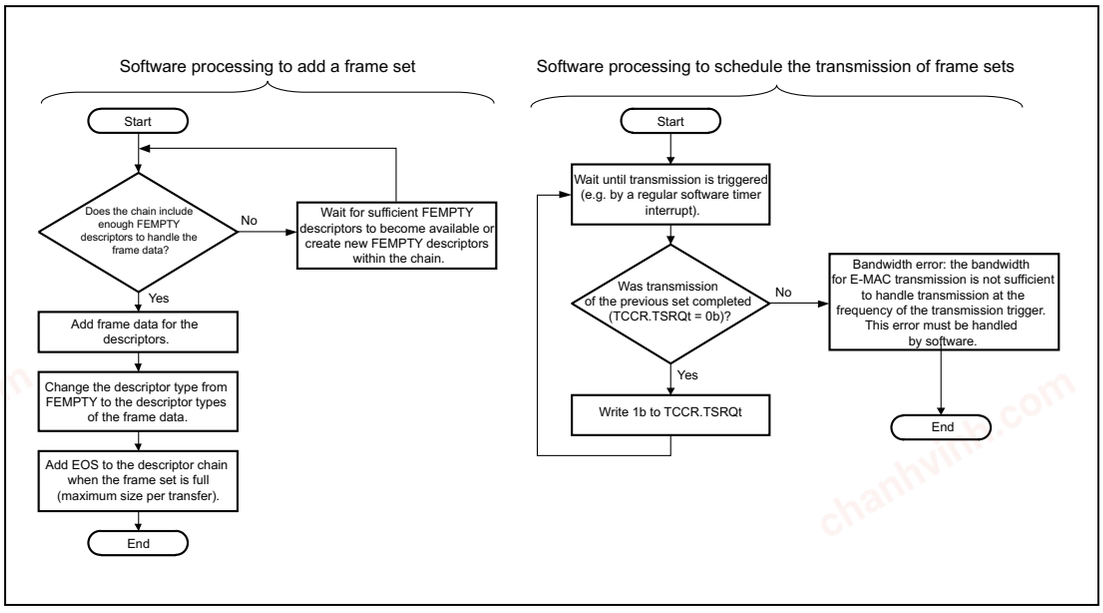


Figure 45.36 Software Flow for Frame Set Transmission with Switching of the Active Descriptor Chain

In a given time, the AVB-DMAC does not use the descriptor chain area currently being updated by software; therefore, descriptor types for a divided frame can be changed in any order.

Figure 45.37 shows software and AVB-DMAC operation examples in this method.

chanhvinh.com

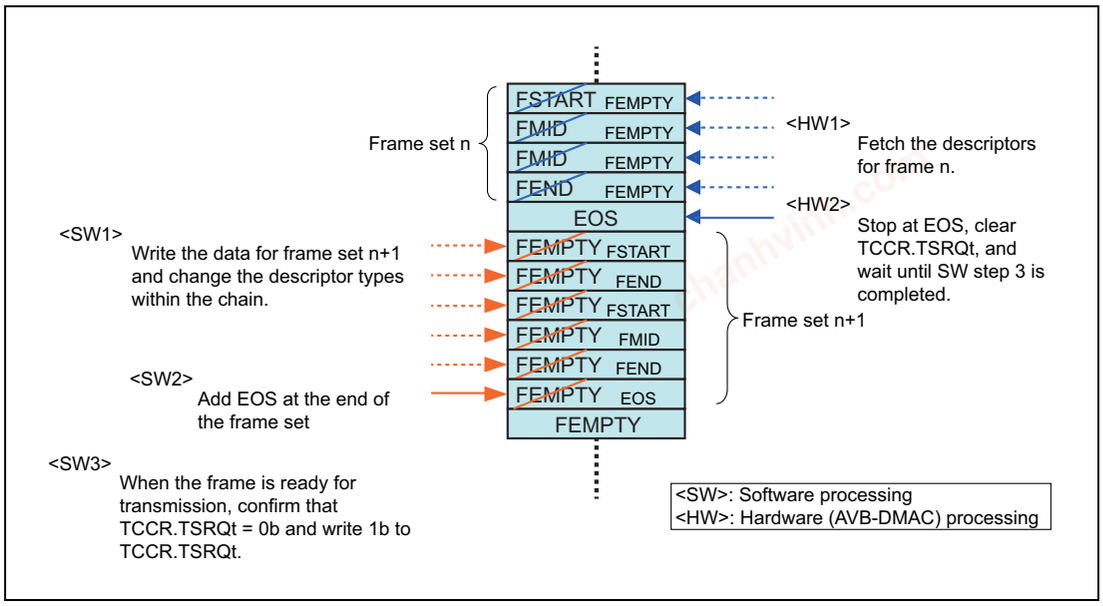


Figure 45.37 SW and AVB-DMAC Operation Examples for Frame Set Transmission with Switching of the Active Descriptor Chain

chanhvinh.com

(c) Frame Set Transmission Using a Shadow Descriptor Chain

This method is used when data are transmitted with a delay for software control to secure bandwidth or for other reasons, rather than immediately transmitted. Two or more descriptor chains are used. The chains are classified as active or shadow chains. Software prepares frame data in shadow chain and descriptor while hardware processes active chain. EOS or FEMPTY descriptors are used for the stop points.

As preparation in transmission, software creates a shadow descriptor chain that has a FEMPTY descriptor, and creates a frame data descriptor in an active chain. Software then sets the transmit start request bit (TCCR.TSRQt) to start transmission of the active chain. During active chain transmission, software can prepare frame data for the shadow chain.

Figure 45.38 shows the flow for software implementing this method.

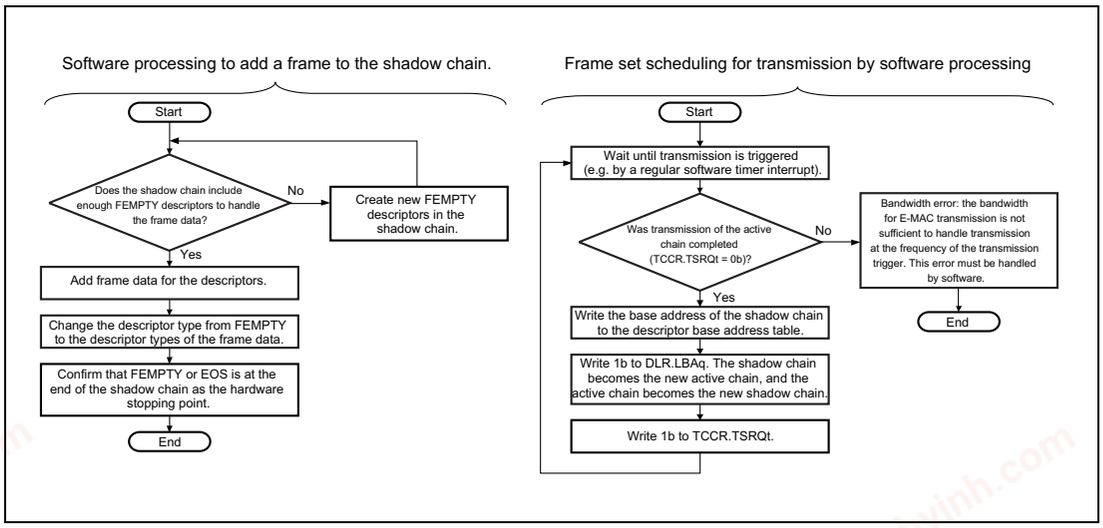


Figure 45.38 Software Flow for Frame Set Transmission Using the Shadow Descriptor Chain

Figure 45.39 shows software and AVB-DMAC operation examples in this method.

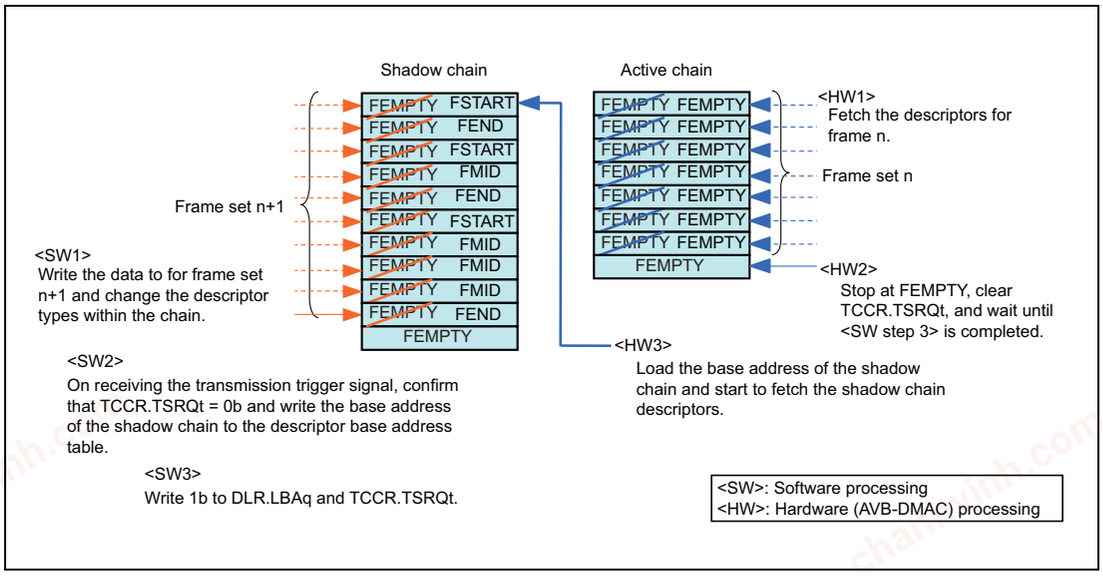


Figure 45.39 SW and AVB-DMAC Operation Examples for Frame Set Transmission Using the Shadow Descriptor Chain

chanhvinh.com

45.3.5.4 Time Stamping in Transmission

Transmission time stamps are important in performing timing and synchronization processing of the IEEE 802.1AS standard. Reference to the transmission time stamps can also be useful to other applications and so on. The AVB-DMAC supplies the time-stamp values based on the gPTP timer by storing the time stamps captured at the same time as sending of the Start of Frame Delimiter (SFD) for transmitted frames.

When the time stamp storage request field (DESCR.TSR) is set to 1, selecting storage of a time stamp, the tag number defined in the tag field (DESCR.TAG) of the last descriptor in a set (FEND) or of an FSINGLE descriptor for the frame being transmitted is stored with the time stamp. This eases identification and association. The time-stamp FIFO is accessible at any time.

The method of using this function is described below:

1. Create descriptor and frame data in the on-chip RAM for the frame requiring time stamping. Write the tag number of the frame to the frame tag field (DESCR.TAG) and set the time stamp storage request field (DESCR.TSR) to 1.
2. The AVB-DMAC fetches and analyzes the descriptor. When the time stamp storage request field (DESCR.TSR) is 1, the AVB-DMAC recognizes that transmitting this frame also requires storage of the time stamp.
3. The AVB-DMAC fetches the frame data and stores the frame data in the transmission FIFO for scheduling.
4. Under the control of priority settings according to credit-based shaping (CBS) or another scheme, the transmission scheduler decides it is time to transmit frame.
5. The EthernetAVB starts transmission of frame.
6. The gPTP time stamp is captured at the start of sending the frame delimiter (SFD) for transmission and stored with the tag frame field (DESCR, TAG) in the time-stamp FIFO upon completion of the frame transmission. When the time stamp update interrupt is enabled, update of time stamp is notified by an interrupt.
7. The time stamp can now be acquired from the time-stamp FIFO.

Use the time-stamp FIFO for the timing and synchronization of frames with IEEE 802.1AS compliance.

Time stamping can also be used with other frames, but take care not to allow the time-stamp FIFO to overflow. When the time-stamp FIFO is full, further time stamps in subsequent transmissions are not stored.

Figure 45.40 shows software flow in time-stamp FIFO operation. The time stamp FIFO update interrupt status bit (TIS.TFUF) and the time stamp FIFO warning interrupt status bit (TIS.TFWF) are used for the start flag of the START block in the figure.

chanhvinh.com

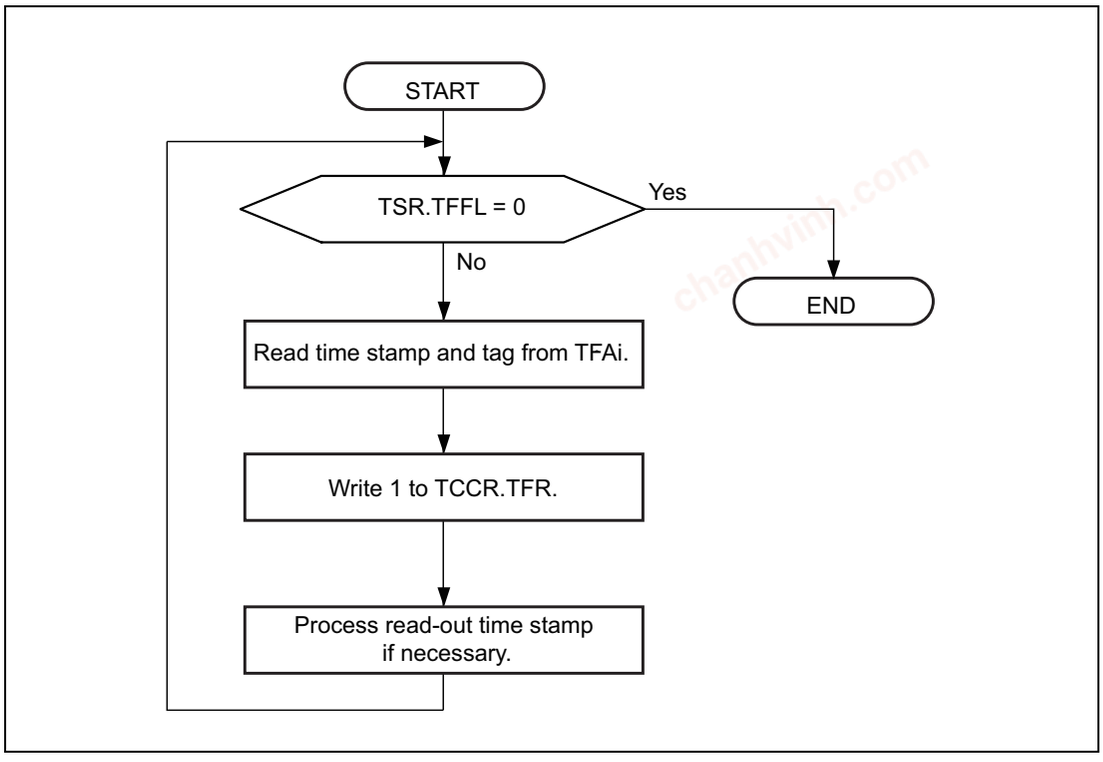


Figure 45.40 Flow of Transmission Time Stamping

When all the entries in the time-stamp FIFO are released, the time stamp FIFO update interrupt status bit (TIS.TFUF) and time stamp FIFO warning interrupt status bit (TIS.TFWF) are cleared by the AVB-DMAC. Therefore, clearing the interrupt flags is not necessary.

When multiple frames are transmitted with time stamps but from different queues, the order of storage in the time-stamp FIFO buffer depends on the order of frames in transmission.

chanhvinh.com

45.3.6 CBS (Credit-Based Shaping)

In AVB transmission mode (i.e. when the transmit queue priority field in the transmit configuration register (TGC.TQP) is B'01 or B'11), transmission queues Q3 and Q2 are respectively assigned to class A and class B stream traffic and the CBS (Credit Based Shaping) algorithm is used to select the transmission queues in order to satisfy the Forwarding and Queuing for Time Sensitive Streams (FQTSS) specification (see section 8.6.8 or section 34 in IEEE 802.1Q).

The CBS algorithm is based on the concept of transmission credit for each queue. Credit can be thought of as the degree to which a queue has the "right" to transmit at a given time. In addition, in AVB transmission mode as specified in IEEE 802.1Q, queues that are subject to the CBS algorithm are able to transmit when the following conditions are met.

- At least one frame is stored in the queue.
- The credit for the queue is 0 or a positive value.

The credit for a transmission queue is incremented while one or more frames from the queue are present in the transmission FIFO but transmission of these frames is not proceeding. This state is indicated by the transmission process status bit for queue t in the AVB-DMAC status register (CSR.TPOt) being clear (0). The credit is decremented while transmission of a frame from the queue is in progress. This mechanism is used to control transmission so that the amount of data for transmission for each queue does not exceed the specified maximum bandwidths.

IEEE 802.1Q defines the following parameters for queues under the control of the CBS algorithm.

portTransmitRate: Maximum transmission data rate of an external port. The E-MAC determines this parameter.

bandwidthFraction: Maximum fraction of portTransmitRate that can be used for a queue.

idleSlope: Rate of change of credit for a queue when transmission of frames from the queue is not proceeding so the credit value (in bits per second) is increasing. idleSlope is also equal to the maximum fraction of the total bandwidth (portTransmitRate) that is available to the given queue under a specified condition (frames from the queue can be placed in a continuous stream. See Annex L of IEEE 802.Q.
idleSlope = bandwidthFraction * portTransmitRate

sendSlope: Rate of change of credit (in bits per second) for a queue while transmission of a frame from the queue is in progress so the credit value is decreasing.
The value of sendSlope is defined as follows:
sendSlope = idleSlope - portTransmitRate

Furthermore, the values below are used to define individual traffic classes (or queues for the classes) under control of the algorithm. See Annex L of IEEE 802.Q.

maxFrameSize: Maximum size of frames (in bits) of the corresponding traffic class that can be transmitted from a port

maxInterferenceSize: Maximum burst size (in bits) by which delays for the corresponding traffic class can be allowed

hiCredit: Maximum credit value (positive number). Can be calculated by using the following equation: hiCredit = maxInterferenceSize * (idleSlope / portTransmitRate)

loCredit: Minimum credit value (negative number). Can be calculated by using the following equation: loCredit = maxFrameSize * (sendSlope / portTransmitRate)

chanhvinh.com

Figure 45.41 shows how the CBS algorithm works and the meaning of the above parameters. When there is no frame to be transmitted in the queue in the transmit-enabled state, the credit value is 0 (after Frame 0 transmission in the figure).

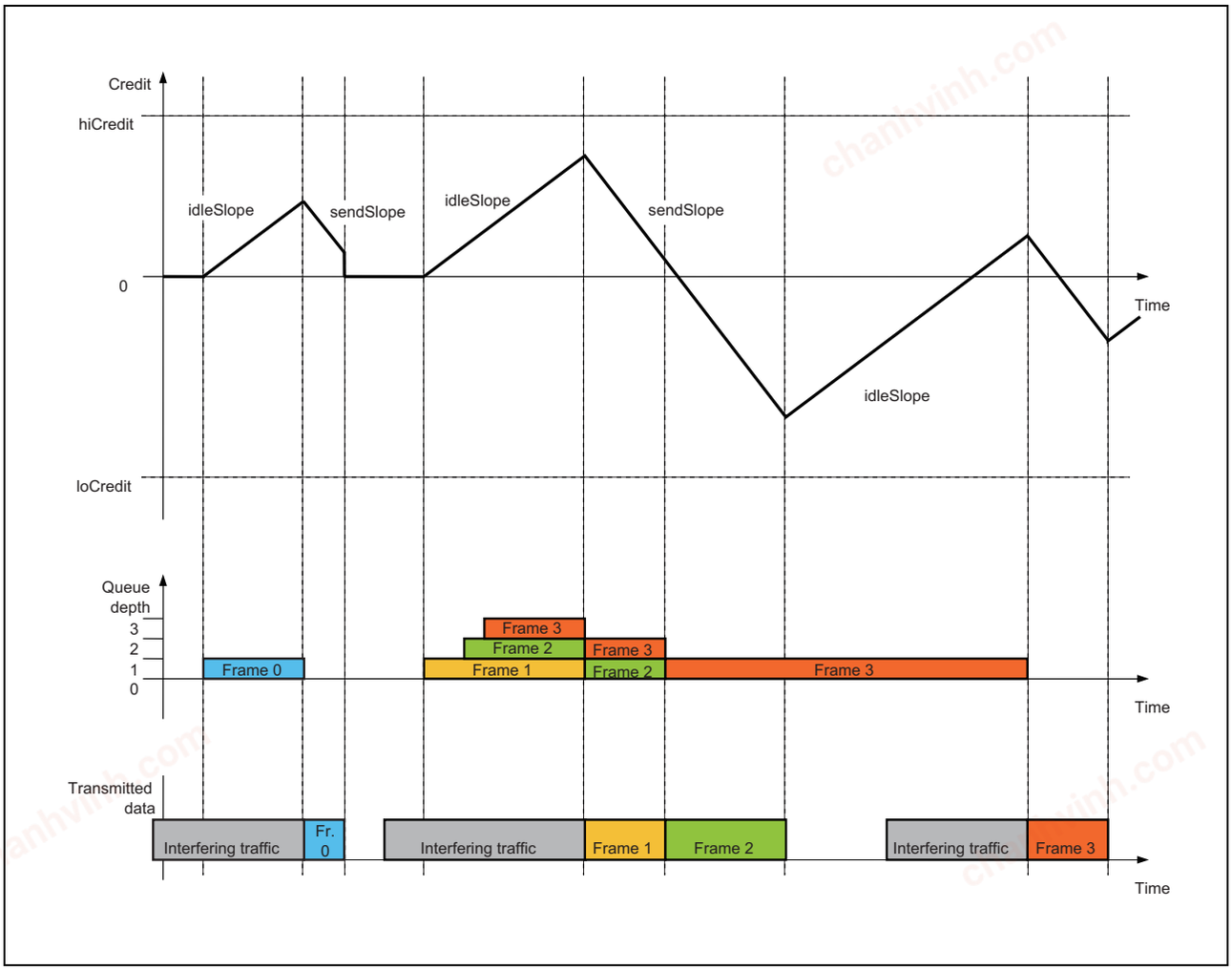


Figure 45.41 CBS (Credit-Based Shaping) Operation

chanhvinh.com

Figure 45.42 shows the operation of CBS (Credit Based Shaping) in the AVB-DMAC.

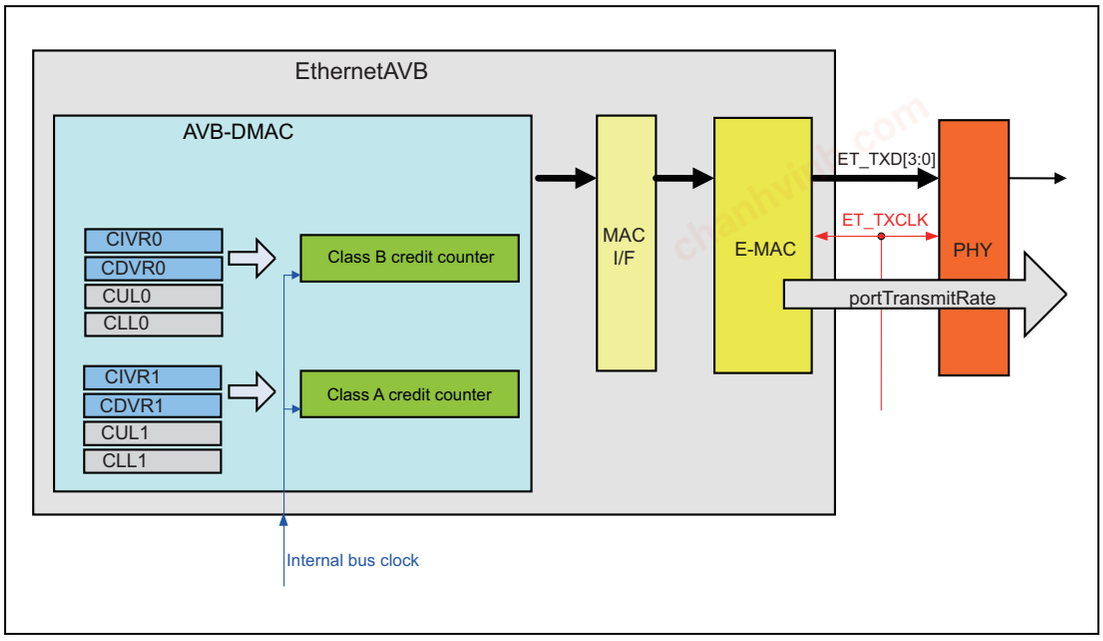


Figure 45.42 CBS (Credit-Based Shaping) Operation in the AVB-DMAC

The CBS operation in the AVB-DMAC is based on “credit counters” for the respective traffic classes (SR class A and class B). The following parameters apply for these classes.

CBS increment value (CIV): Signed positive number
The credit is incremented by this amount every internal bus clock cycle while a frame from the queue is pending but transmission has not started (idleSlope).

CBS decrement value (CDV): Signed negative number
The credit is decremented by this amount every internal bus clock cycle while transmission of a frame from the queue is proceeding (sendSlope).

The CBS increment value (CIV) and CBS decrement value (CDV) are defined as follows.

CIV = idleSlope * Mfactor

CDV = sendSlope * Mfactor

Mfactor is a multiplier factor to ensure accuracy for CIV and CDV. CIV and CDV are finally calculated by using the following equations.

CIV = (portTransmitRate/Bϕ) * bwFraction * Mfactor

CDV = (portTransmitRate/Bϕ) * (bwFraction - 1) * Mfactor

The credit counters are driven by the internal bus clock (Bϕ), so calculating the slope parameters for CBS requires (1/Bϕ).

Mfactor must be calculated for the CBS parameters. All queues for the same class must have the same CBS parameters. Mfactor for a certain class can be changed during operation, unless transmission is pending for that class (i.e. the transmit process status bit in the AVB-DMAC status register (CSR.TPOt) = 0). At that time, the credit counter values for class are 0. Note that the credit value will

chanhvinh.com

not match a new incrementation or decrementation parameter if Mfactor is changed while the credit counter value is non-zero. Mfactor configuration register is not present in the AVB-DMAC.

Set the CIV and CDV parameters in the CBS increment value registers (CIVRc) and the CBS decrement value registers (CDVRc). They should be dynamically updated when streams are registered and erased in accord with IEEE 802.1Qat.

Although the credit counters operate with the internal bus clock (Bφ), a transmission clock is used in frame transmission. Therefore, time lags occur when frame transmission starts and ends.

For this reason, an extra credit is stored after a frame is transmitted. The maximum value of the credit to be stored is the credit value incremented for one cycle of the Bφ clock and equals the Tx port transmission rate in Mbps * Bφ.

Bandwidth needs to be secured by software taking this into account.

The AVB-DMAC also has CBS upper limit registers (CULc) (the upper limit registers for classes A and B) and CBS lower limit registers (CLLc) (the lower limit registers for classes A and B). Multiply the upper limit (hiCredit) and the lower limit (loCredit) by Mfactor for each class as defined above so that the resulting values agree with the credit values.

CUL = hiCredit * Mfactor = maxInterferenceSize * bwFraction * Mfactor

CLL = loCredit * Mfactor = maxFrameSize * (bwFraction - 1) * Mfactor

Example:

Assume that portTransmitRate = 100 Mbps, Bϕ = 130 MHz and bwFraction = 3%. Then idleSlope and sendSlope represented as one bit vs. cycles of the peripheral bus clock are as follows.

idleSlope = (portTransmitRate/Bϕ) * bwFraction = 100/130 (Mbps/MHz) * 3% = 0.023 (bit/Bϕ)

sendSlope = idleSlope - (portTransmitRate / Bϕ) = -0.746 (bit/Bϕ)

Let Mfactor be 100, then CIV and CDV parameters are determined as follows.

CIV = idleSlope * Mfactor = 23

CDV = sendSlope * Mfactor = -74.6

chanhvinh.com

45.3.6.1 Restrictions on CIV, CDV and Mfactor

The maximum value (the minimum value for negative numbers) up to which the credit counter will not overflow determines the maximum values of CIV and CDV that can be set in the CBS registers. This maximum credit value is equivalent to the worst case of the hiCredit value, and the maximum values for class A and class B are calculated as follows.

<Conditions>

- Class A maximum value (hiCredit_max_classA)
  classA bwFraction ≅ 100%
  Maintaining the proper relations in the transmission priority order requires waiting for a period equivalent to the one maximum sized frame.
  hiCredit_max_classA ≅ maxInterferenceSize for class A = Waiting for a period equivalent to one maximum sized frame = header + maximum size payload + CRC (2000 bytes) + preamble (8 bytes) + IFG (12 bytes) + processing_delay (≅ 80 bytes) ≅ 2100 bytes
- Class B maximum value (hiCredit_max_classB)
  classB bwFraction ≅ 100%
  Maintaining the proper relations in the transmission priority order requires waiting for a period equivalent to the respective one maximum sized frame in the class A transmission queue and other transmission queues.
  hiCredit_max_classB ≅ maxInterferenceSize for class B = Waiting for a period equivalent to two maximum sized frames = 2* hiCredit_max_classA ≅ 4200 bytes

In bit units, calculated as follows:

hiCredit_max_classA = 16800

hiCredit_max_classB = 33600

The 32-bit signed counter can handle from -2^{31} to $2^{31}-1$, so the specifiable maximum values of Mfactor without overflow are:

Mfactor_max_classA = $2^{31}-1$ / hiCredit_max_classA ≅ 127826 and
Mfactor_max_classB = $2^{31}-1$ / hiCredit_max_classB ≅ 63913.

A high degree of accuracy can be achieved even with a low bandwidth. In class B, bandwidthFraction = 0.05% and the bandwidth error < 0.1%.

The maximum value of CIV is calculated from the following equation.

CIV = idleSlope x Mfactor = (portTransmitRate / Bϕ) * bandwidthFraction × Mfactor

When Mfactor is the maximum value and bandwidthFraction is the maximum value (up to 100%):

CIV_max_classA = (portTransmitRate / Bϕ) * Mfactor_max_classA and
CIV_max_classB = (portTransmitRate / Bϕ) * Mfactor_max_classB.

The maximum values when portTransmitRate = 100 Mbps and Bϕ = 130 MHz are as follows:

CIV_max_classA ≅ 98328

CIV_max_classB ≅ 49164

chanhvinh.com

Table 45.81 shows examples of values for portTransmitRate and internal bus clock frequency. The values in this table are just the results of calculation, so when actually setting values, the CIV values must be limited so that the 32-bit credit counter will not overflow. In the AVB-DMAC, the CIV parameters are implemented as 16 bits + a sign bit, so CIV ≤ 65535 should be applied to both class A and class B.

Table 45.81 Example of Maximum Values for Class A and Class B CIV Parameters

portTransmitRate	Bϕ [MHz]	CIV_max_classA	CIV_max_classB
100 Mbps	100	127826	63913
100 Mbps	125	102260	51130
100 Mbps	133	96109	48054

chanhvinh.com

45.3.6.2 Credit Incrementation (IFGs)

In the CBS credit counter in the AVB-DMAC, the inter-frame gap (IFG) after a frame is transmitted is not treated as part of frame transmission. During an IFG, the credit is incremented for all SR queues that have pending frames or negative credit. Figure 45.43 illustrates credit operations during IFGs.

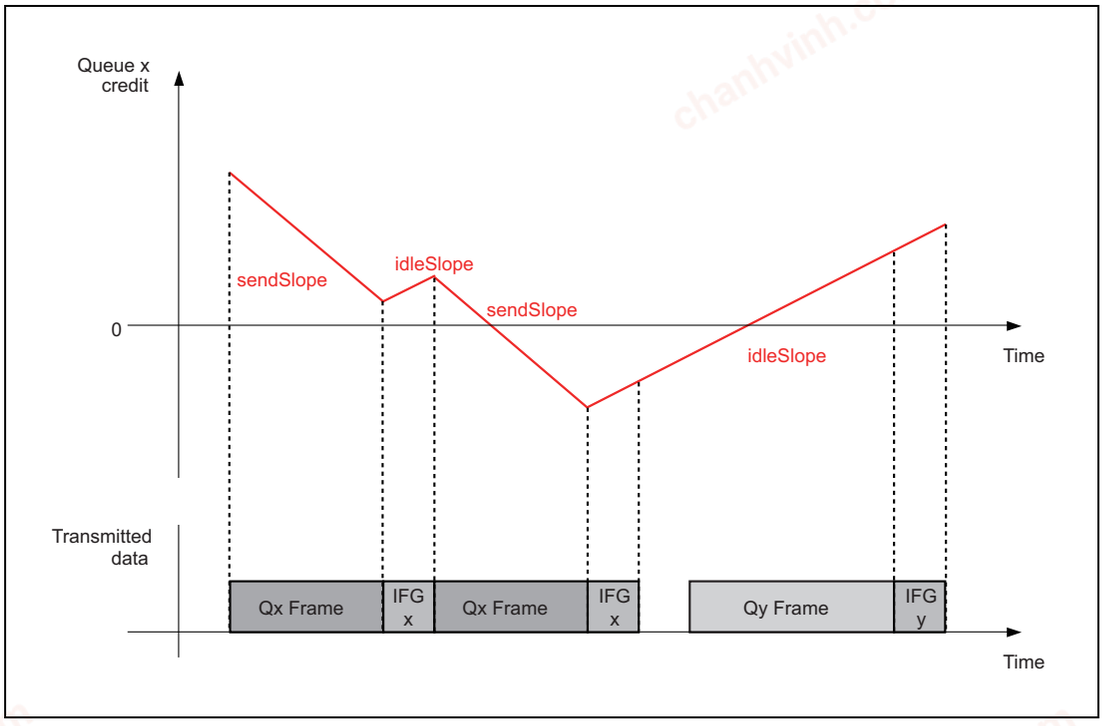


Figure 45.43 Credit Operations during IFGs

The IFG need not be included in calculation of the bandwidth for the specified SR class when deciding the idleSlope, sendSlope, and CIV and CDV parameters. However, IFG must also be included in the calculation in order to confirm that the total bandwidth allocated to all SR classes does not exceed 100% of portTransmitRate.

chanhvinh.com

45.3.6.3 CBS Setting Example

The case of a class A 48-kHz stereo audio stream among Ethernet frames is described as an example.

After every class A measurement interval (125 us), 80 octets consisting of two sets of six 32-bit samples plus a 32-octet header are stored as audio data within a frame. The IEEE 802.3 also imposes a 42-octet media-specific frame overhead (an 8-octet preamble, 14-octet IEEE 802.3 header, 4-octet IEEE 802.1Q priority/VID Tag, 4-octet CRC, and 12-octet IFG) are also added. Accordingly, the total frame size is 80 + 42 = 122, and one such frame is transmitted after every class measurement interval.

This represents a total bandwidth of about 7.8 Mbits per second (122 octets × 8 bits per octet × 8000 frames per second) for this class. The E-MAC runs at 100 Mbps (portTransmitRate), so the allocation of the total bandwidth to each class A queue corresponds to 7.8%. If other traffic classes are to share the total transmission bandwidth, checking that the total allocation of bandwidth with this 7.8% allocation does not exceed 100% of portTransmitRate is required.

To obtain the CIV and CDV parameters for a given class, the IFG must not be taken into account in calculation of the frame size. For this case, therefore, we obtain the total bandwidth for the class = about 7.04 Mbps (110 octets × 8 bits per octet × 8000 frames per second) = 7.04% of portTransmitRate.

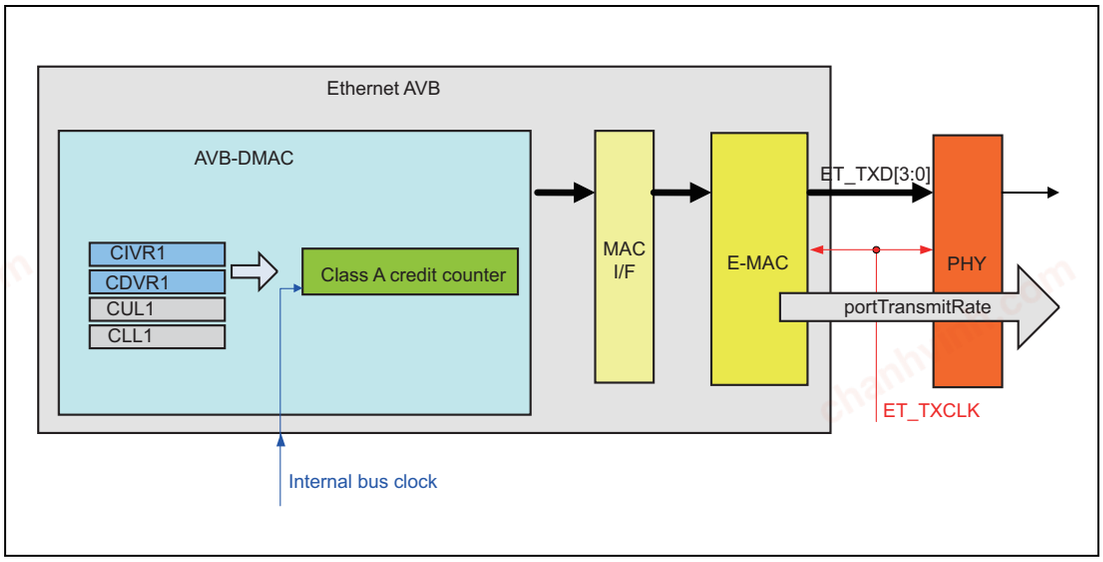


Figure 45.44 Example of CBS Settings

In this setting example, operating frequency and so on in each section are as follows:

- the E-MAC runs at 100 Mbps, so portTransmitRate = 100 Mbps and
- high-speed internal bus clock (operating clock for the credit counter) frequency = 133 MHz,

securing a bandwidth of 7.04 Mbits/sec for class A requires configuring the CBS parameters as follows.

- bandwidthFraction = 7.04%
- idleSlope = (portTransmitRate / Bϕ) * bandwidthFraction ≅ 0.05293 bits per Bϕ
- sendSlope = idleSlope – (portTransmitRate / Bϕ) ≅ -0.69895 bits per Bϕ

When Mfactor = 10000, the parameters are as follows.

- CIV = idleSlope × Mfactor = 529 bits per Bϕ

chanhvinh.com

- CDV = sendSlope × Mfactor = -6989 bits per Bϕ

These are the final values for setting in the CIVR1 and CDVR1 registers.

chanhvinh.com

45.3.7 IEEE802.1: gPTP

45.3.7.1 gPTP Timer

An 84-bit timer is provided to support the gPTP function. Figure 45.45 shows the overview of the timer.

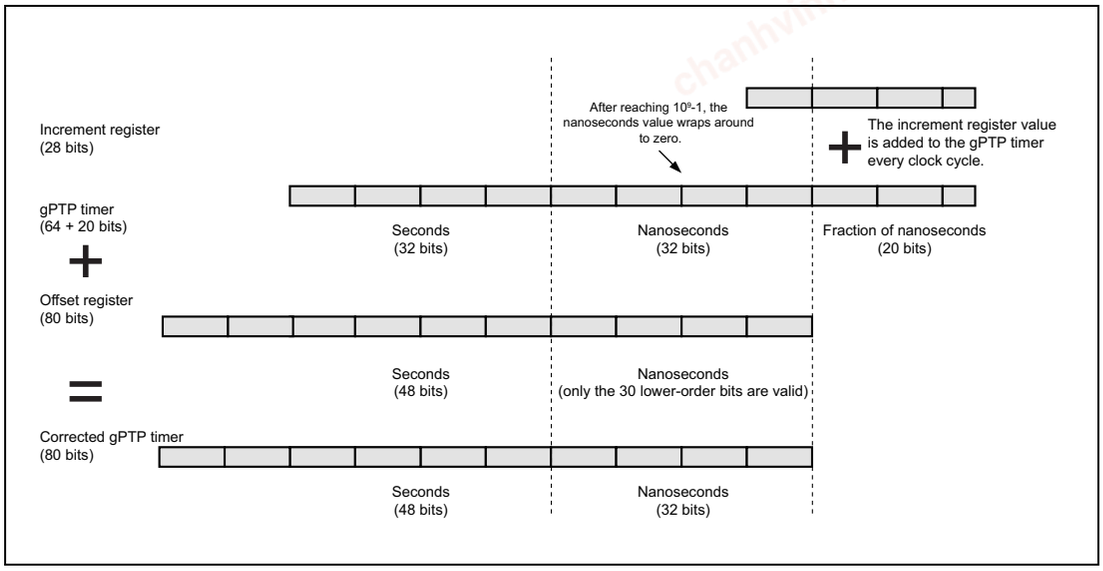


Figure 45.45 gPTP Timer

The higher-order 32 bits indicate seconds. For the next 32 bits indicate nanoseconds. The lower-order 20 bits indicate fractional nanoseconds. Software can only read the 32 higher-order bits, indicating seconds, and the subsequent 32-bits, indicating nanoseconds. The 20 lower-order 20 bits, representing less than 1 ns, are provided within the AVB-DMAC to maintain accuracy in time measurement, and cannot be referenced.

The timer can be reset by setting the timer control request bits in the gPTP configuration control register (GCCR.TCR[1:0]) to B'01. These bits are set to B'00 on completion of normal resetting of the timer.

After the timer starts, the value in the gPTP timer increment value bit (GTI.TIV) is added to the value of the gPTP timer every clock cycle. When the value of the nanosecond part reaches 10^9 - 1, it returns to 0 when the next 1 nanosecond is counted. The fractional nanosecond value is represented by 20 bits and can be adjusted in precision of $1/2^{20}$ (1/1048576) nanosecond.

Before setting a value in the gPTP timer increment value bit (GTI.TIV), set the timer increment value setting request bit in the gPTP configuration control register (GCCR.LTI) to 1. If this bit is not set to 1, new values that are written will not be reflected in the register. This bit returns to 0 after the setting is completed.

An offset to the gPTP timer is also available. If this is required, set the value in the timer offset value bit (GTO.TOV). Before setting a value in this register, set the timer offset value setting request bit in the gPTP configuration control register (GCCR.LTO) to 1. If this bit is not set to 1, new values that are written will not be reflected in the register. This bit returns to 0 after the setting is completed. Set the value from 0 to 10^9 - 1 to the timer offset value bits (GTO0.TOV[31:0]) in the nanoseconds portion. When adding an offset, take care that the calculation result with the offset added does not exceed 80 bits.

chanhvinh.com

The value of the gPTP timer can be read from the gPTP timer capture value bit (GCTi.CTV) by setting the timer control request bits in the gPTP timer control register (GCCR.TCR[1:0]) to B'11. Set the timer capture source select bits in the gPTP configuration control register (TCCR.TCSS) to select the timer value for capture as the value of the gPTP timer, the corrected value of the gPTP timer (value with the offset added), or the AVTP presentation time. Setting the timer control request bits in the gPTP configuration control register (GCCR.TCR[1:0]) to B'11 initiates the capture. Once normal capture of the timer is complete, the value of the timer control request bits in the gPTP configuration control register (GCCR.TCR[1:0]) returns to B'00.

The timer for gPTP operates as a free-running timer but can be synchronized with the Grandmaster clock.

The timer clock source can be selected from the internal bus clock and Ethernet transmission clock by setting the gPTP clock select bits (CCC.CSEL[1:0]).

45.3.7.2 Free-Running Operation

The IEEE 802.1 AS standard does not prescribe the adjustment of local clocks to the Grandmaster clock. To avoid negative effects from the correction procedure, we recommend the use of a free-running timer.

As a free-running timer, the timer counts the local clock based on the local time. The gPTP timer increment value bit (GTI.TIV) is set to 1 ns (the setting value = H'0010 0000) and the timer offset value bit (GTOi.TOV) is set to 0. The ratio information captured at the time of the gPTP delay measurement and synchronization procedures is used to correct the frequency ratio to that of the Grandmaster clock. The Grandmaster clock can be calculated from the local clock by using the information collected during the gPTP measurement and synchronization procedures.

45.3.7.3 Synchronization with the Grandmaster Clock

In situations requiring physical synchronization of the local clock with the Grandmaster clock, the fractional nanoseconds value (the 20 lower-order bits of the gPTP timer) is used to make the adjustment. Specifically, the increment value is finely adjusted to correct for deviations of the clock frequency from that of the Grandmaster clock.

Use the timer offset value bit (GTOi.TOV) to correct for offsets for comparing with the absolute time and so on from start-up. The sum of the timer value and the offset register is the corrected gPTP timer value.

Set the value from 0 to 10^9 - 1 to the timer offset value bits (GTOi.TOV[31:0]) for setting the nanoseconds portion of the offset.

The following equation gives a method of calculating the increment (GTI.TIV) from the frequency of the gPTP clock and its deviation from that of the Grandmaster clock. Variable d is the deviation (d = 10^{-6} for 1 ppm).

$$GTI.TIV = round\left(\frac{2^{20}\,GHz}{f_{GPTP}} * (1+d)\right)$$

After adjusting for the current deviation of clock frequency, re-set the gPTP timer increment register (GTI.TIV).

After calculating the new offset value, re-set the timer offset configuration register (GTOi.TOV).

- Sample setting 1

When f_{GPTP} = 100 MHz ± 0 ppm (device used is guaranteed by the Grandmaster clock)

chanhvinh.com

gPTP is updated every 10 ns.
Set the gPTP timer increment value bits (GTI.TIV) to H'00A0_0000.

- Sample setting 2

When f_{GPTP} = 100 MHz - 10 ppm
gPTP is updated every 10 ns.
Set the gPTP timer increment value bits (GTI.TIV) to H'00A0_0069 (10485864.8576 before rounded off). Rounding off the value causes a frequency error less than 1 ppm.

45.3.7.4 Support Provided by the gPTP Timer in Transmission and Reception

The timer value described above is used in the time-stamp values captured when start frame delimiters are detected in reception and generated in transmission.

Captured time stamp values for received frames are stored in the corresponding descriptors. Those for transmitted frames are stored with tag information in the time-stamp FIFO. The time stamp values are thus correlated with both transmitted and received frames.

Note that the use of corrected gPTP timer values can introduce an error due to the offset correction in the gPTP synchronization procedure.

Errors due to SFD notification and the interface between the timer modules must also be taken into account. Although SFDs for transmission and reception are detected in synchronization with the transmission and reception clocks, respectively, the gPTP operates with a gPTP clock. Therefore, an error of ± 1 GTI.TIV + 1 Bφ is generated.

chanhvinh.com

45.3.8 Support for IEEE 1722

For IEEE 1722, the following two functions are supported.

- Output and capture of values in the IEEE 1722 AVTP (Audio/Video Transport Protocol) presentation time format
- Comparison of IEEE 1722 AVTP presentation time stamps

The 32-bit AVTP time stamp field of IEEE 1722 frames holds the AVTP presentation time when the AVTP time-stamp enable bit in the frame is 1. The AVTP time stamp field is generated from the pPTP timer and is given as seconds (gPTP_seconds) and nanoseconds (gPTP_nanoseconds) according to the following equation.

AVTP time stamp = (gPTP_seconds * 10^9 + gPTP_nanoseconds) modulo 2^{32}

The AVTP presentation time can be read from the gPTP timer capture value bit (GCTi.CTV). Set the timer capture source select bits in the gPTP configuration control register (TCCR.TCSS) to select the timer value for capture as the AVTP presentation time. Setting the timer control request bits in the gPTP configuration control register (GCCR.TCR[1:0]) to B’11 initiates the capture. The value stored as the capturing result is obtained by adding the maximum transit time defined in the maximum transit time bits (GMTT.MTTV) to the corrected gPTP timer value. The AVTP presentation time wraps around approximately every four seconds.

CAUTION

The AVTP presentation time captured in GCTi.CTV is only valid when the corrected gPTP timer value is in synchronization with the Grandmaster clock. That is, the timer increment and timer offset values for the corrected gPTP timer value must be adjusted by the synchronization procedure so that the corrected gPTP clock is adjusted to match the time kept by the Grandmaster clock.

chanhvinh.com

45.3.9 Flow Control

The E-MAC supports flow control for full-duplex operation in compliance with the IEEE 802.3 standards. This flow control is applicable to both reception and transmission. In regard to the transmission of PAUSE frames, flow control operates in the following ways.

(1) Automatic PAUSE Frame Transmission

For the received frames, PAUSE frames are automatically transmitted when the amount of data having been written in the reception FIFO reaches the value specified by the receive FIFO caution level bits in the receive configuration register (RCR.RFCL). The TIME parameter contained in the PAUSE frame can be specified by the automatic PAUSE frame register (APR). If the maximum count of PAUSE frame retransmission times is not specified, PAUSE frames are automatically transmitted repeatedly until the data is read from the reception FIFO and the remaining data amount becomes smaller than the value specified by the RCR.RFCL. As the maximum count of PAUSE frame retransmission times, any value from 1 to 65535 can be specified by the PAUSE frame retransmission count register (TPAUSER). In this case, PAUSE frames are automatically transmitted repeatedly until the remaining data amount becomes smaller than the value specified by the RCR.RFCL or the count of retransmission times reaches the value specified by the TPAUSER. The retransmit counter is cleared to 0 when the next PAUSE frame is transmitted once the data amount in the reception FIFO becomes smaller than the value specified by the RCR.RFCL.

Automatic PAUSE frame transmission is enabled when the operating mode bit for flow control in transmission in the E-MAC mode register (ECMR.TXF) is 1.

(2) Manual PAUSE Frame Transmission

PAUSE frames can also be transmitted in response to software operations. Writing a timer value to the manual PAUSE frame register (MPR) starts the transmission of a PAUSE frame. This only causes the transmission of one PAUSE frame.

(3) PAUSE Frame Reception

After reception of a PAUSE frame, transmission of the next frame does not proceed until the time indicated by the Timer value elapses. However, transmission of a frame currently being transmitted continues. PAUSE frames are only received while the operating mode for flow control in reception bit in the E-MAC mode register (ECMR.RXF) is set to 1. The number of received PAUSE frames is counted.

(4) 0 TIME PAUSE Frame Control

The setting of the PAUSE frame usage with TIME = 0 enable bit in the E-MAC mode register (ECMR.ZPF) enables or disables the transmission of PAUSE frames with the TIME parameter value 0. The setting of the PAUSE frame reception with time = 0 bit in the E-MAC mode register (ECMR.RZPF) enables or disables the reception of PAUSE frames with the TIME parameter value 0.

- Operation for transmission
  When the 0-time PAUSE frame control is enabled, PAUSE frame with the TIME parameter value 0 is transmitted when the capacity of the reception FIFO is less than the value of the receive FIFO caution level bits in the receive configuration register (RCR.RFCR) while the time indicated by the TIME parameter value has not elapsed.
  When the 0-time PAUSE frame control is disabled, PAUSE frames with the TIME parameter value 0 are not transmitted.

- Operation for reception

chanhvinh.com

When the 0-time PAUSE frame reception is enabled, frame transmission wait state is released when PAUSE frame with the TIME parameter value 0 is received.
When the 0-time PAUSE frame reception is disabled, received PAUSE frames with the TIME parameter value 0 are discarded.

chanhvinh.com

45.3.10 Interrupts

The EthernetAVB module has four interrupts: three interrupts from the AVB-DMAC and one interrupt from the E-MAC. The three interrupts from the AVB-DMAC are generated when the AVB-DMAC is placed in operation mode, and the one interrupt from the E-MAC is generated when the AVB-DMAC is placed in configuration, operation, or standby mode.

Table 45.82 is a list of the interrupts.

Table 45.82 EthernetAVB Interrupts

Interrupt Source Name	Remarks
AVB_DATA	Transmit/receive data management interrupt
AVB_ERROR	Error management interrupt
AVB_MANAGE	Other management (FIFO caution level, etc.) interrupt
AVB_MAC	E-MAC interrupt

The AVB-DMAC related interrupts include descriptor interrupts (15 sources), error interrupts (5 sources), reception interrupts (37 sources), transmission interrupts (2 sources), and gPTP interrupts (3 sources). From the CPU's perspective, each appears as one of the above four interrupt sources.

The states of an AVB-DMAC-related interrupt sources can be checked in the following registers.

- Descriptor interrupt status register (DIS)
- Error interrupt status register (EIS)
- Receive interrupt status register (RISi)
- Transmit interrupt status register (TIS)
- gPTP interrupt status register (GIS)

The interrupts are controlled by the corresponding interrupt enable bits. However, the status flags operate independently of the settings of the enable bits.

The states of grouped interrupts can be checked by reading the interrupt summary status register (ISS) and the queue full error interrupt status bit in the error interrupt status register (EIS.QFS).

45.3.10.1 Transmit/Receive Data Management Interrupt

The management interrupt for transmission and reception is conveyed when the interrupt conditions corresponding to the following sources are satisfied.

- Receive frame interrupt in the receive interrupt status register 0 (RIS0.FRFr)
- Descriptor interrupt in the descriptor interrupt status register (DIS.DPFi)

chanhvinh.com

45.3.10.2 Error Management Interrupt

The error management interrupt is conveyed when interrupt conditions corresponding to the following sources are satisfied.

- Time stamp FIFO full error interrupt in the error interrupt status register (EIS.TFFF)
- CBS limitation value error interrupts in the error interrupt status register (EIS.CULF1, EIS.CULF0, EIS.CLLF1, EIS.CLLF0)
- Receive FIFO full interrupt in the receive interrupt status register 2 (RIS2.RFFF)
- Receive queue full interrupt in the receive interrupt status register 2 (RIS2.QFFr)

45.3.10.3 Other Management (FIFO Warning, etc.) Interrupts

The other management (FIFO warning, etc.) interrupt is conveyed when interrupt conditions corresponding to the following sources are satisfied.

(1) Reception related interrupt

Receive FIFO warning interrupt in the receive interrupt status register 1 (RIS1.RFWF)

(2) Transmission related interrupts

Time stamp FIFO warning interrupt in the transmit interrupt status register (TIS.TSWF)

Time stamp FIFO update interrupt in the transmit interrupt status register (TIS.TSUF)

(3) gPTP related interrupts

Presentation time match interrupt in the gPTP interrupt status register (GIS.PTMF)

45.3.10.4 E-MAC Interrupt

The E-MAC interrupt is conveyed when the following E-MAC interrupt sources are generated.

- PAUSE frame retransmit retry over interrupt in the E-MAC status register (ECSR.PFROI)
- Illegal carrier detection interrupt in the E-MAC status register (ECSR.ICD)

chanhvinh.com

45.3.11 Flows of Operations

45.3.11.1 Flow of E-MAC Initialization

Figure 45.46 shows the flow of E-MAC initialization.

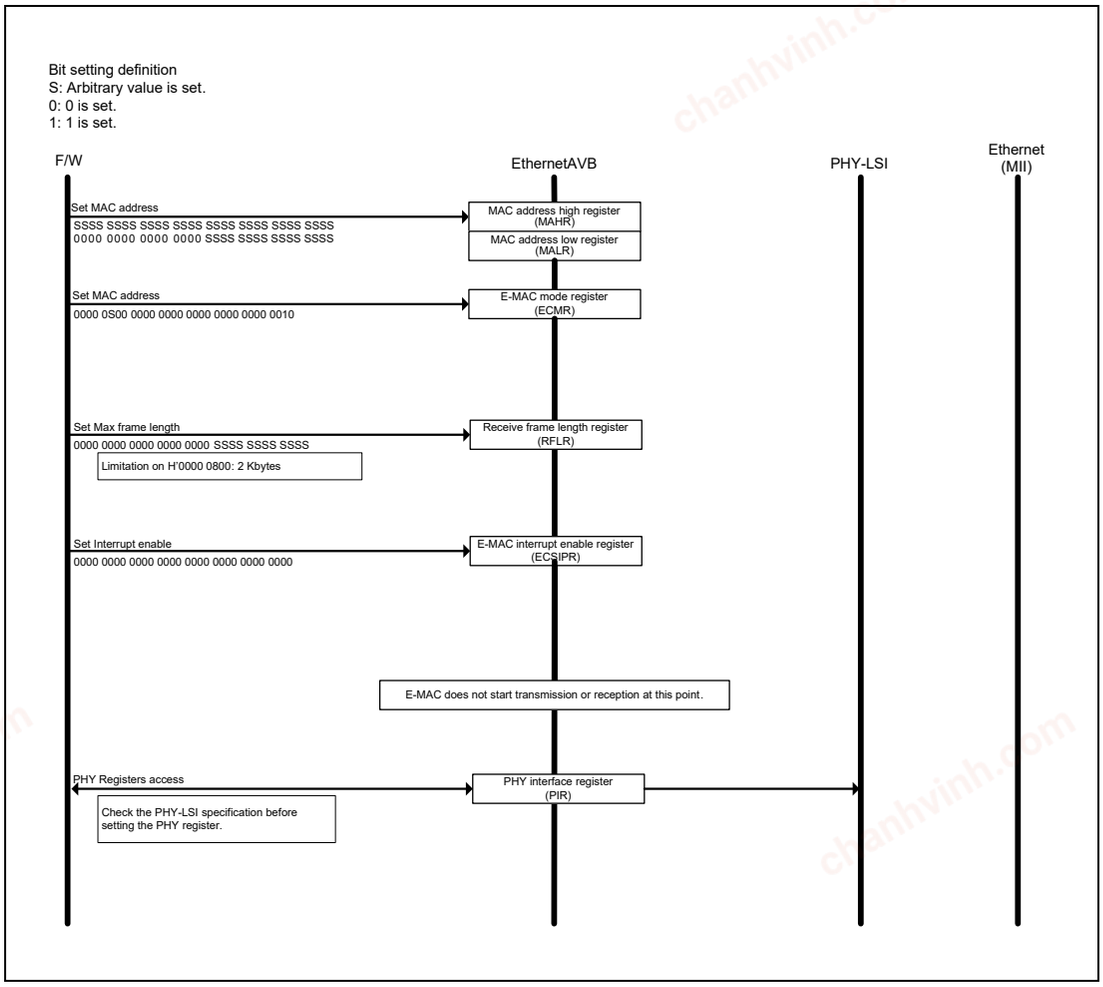


Figure 45.46 Flow of E-MAC Initialization

chanhvinh.com

45.3.11.2 Flow of AVB-DMAC Initialization

Figure 45.47 shows the flow of AVB-DMAC initialization.

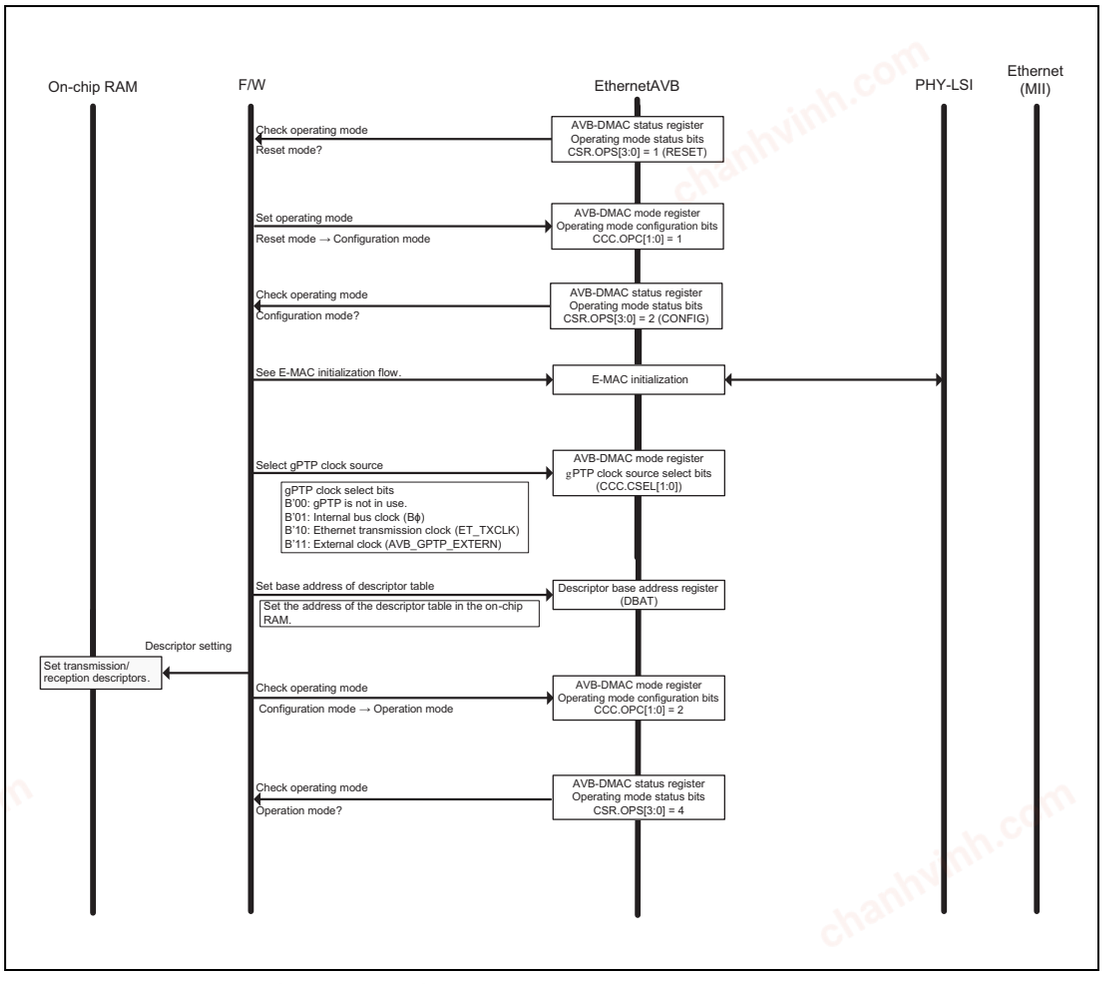


Figure 45.47 Flow of AVB-DMAC Initialization

chanhvinh.com

45.3.11.3 Flow for the AVB-DMAC in Reception

Figure 45.48 shows the flow for the AVB-DMAC in reception.

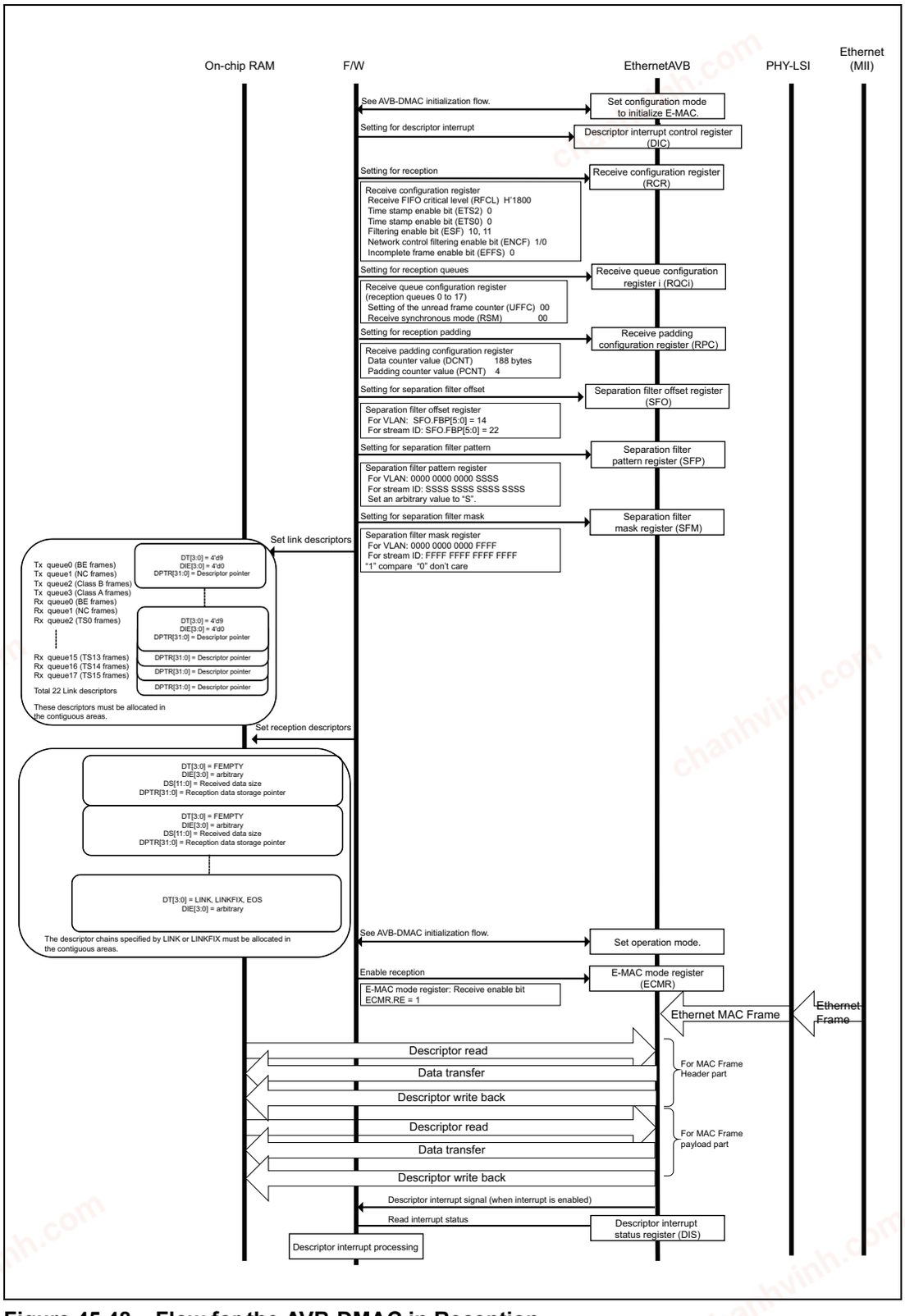


Figure 45.48 Flow for the AVB-DMAC in Reception

chanhvinh.com

45.3.11.4 Flow for the AVB-DMAC in Transmission

Figure 45.49 shows the flow for the AVB-DMAC in transmission.

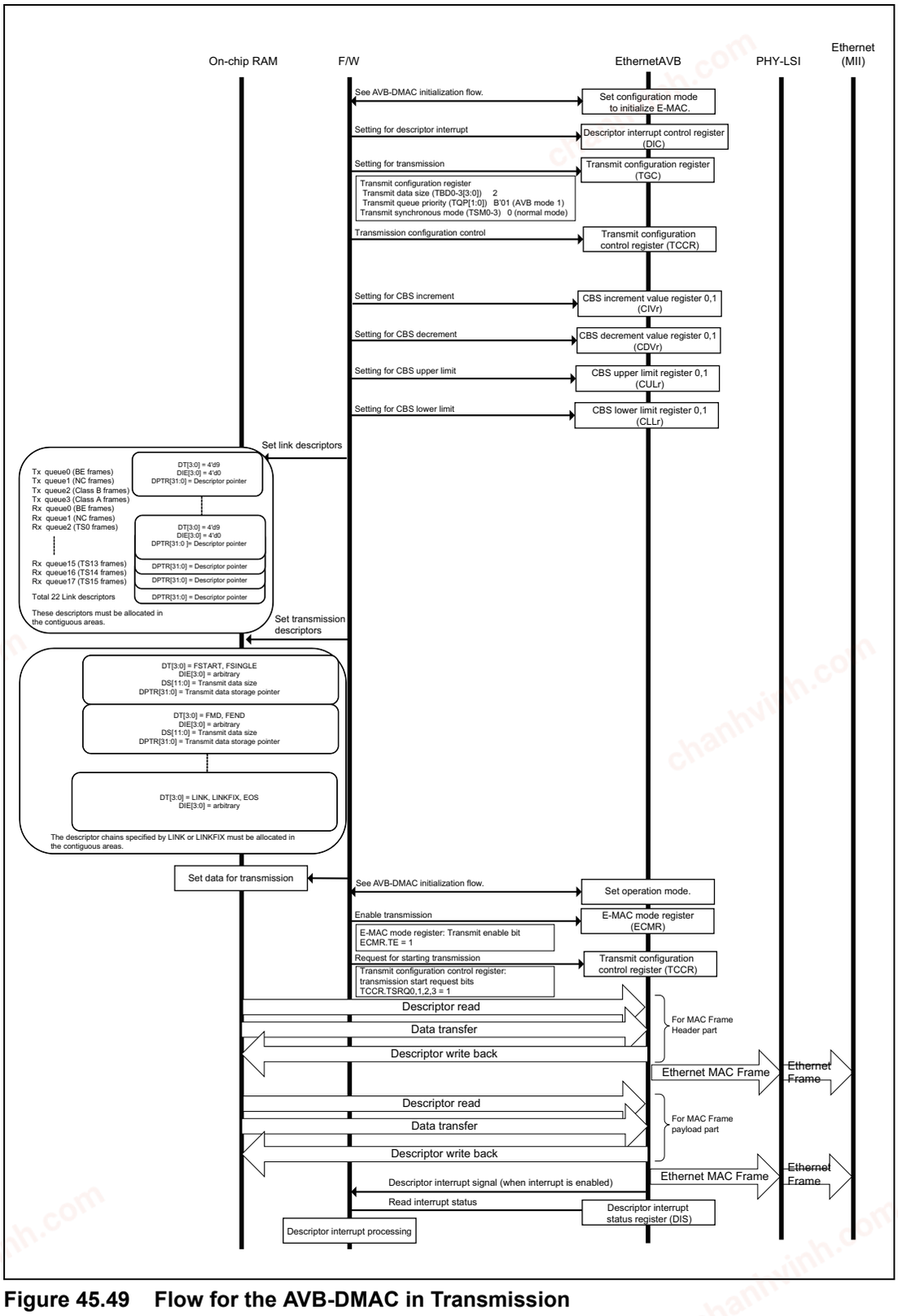


Figure 45.49 Flow for the AVB-DMAC in Transmission

chanhvinh.com

45.3.11.5 Flow for Stopping AVB-DMAC Operation in Reception

Figure 45.50 shows the flow for stopping AVB-DMAC operation in reception.

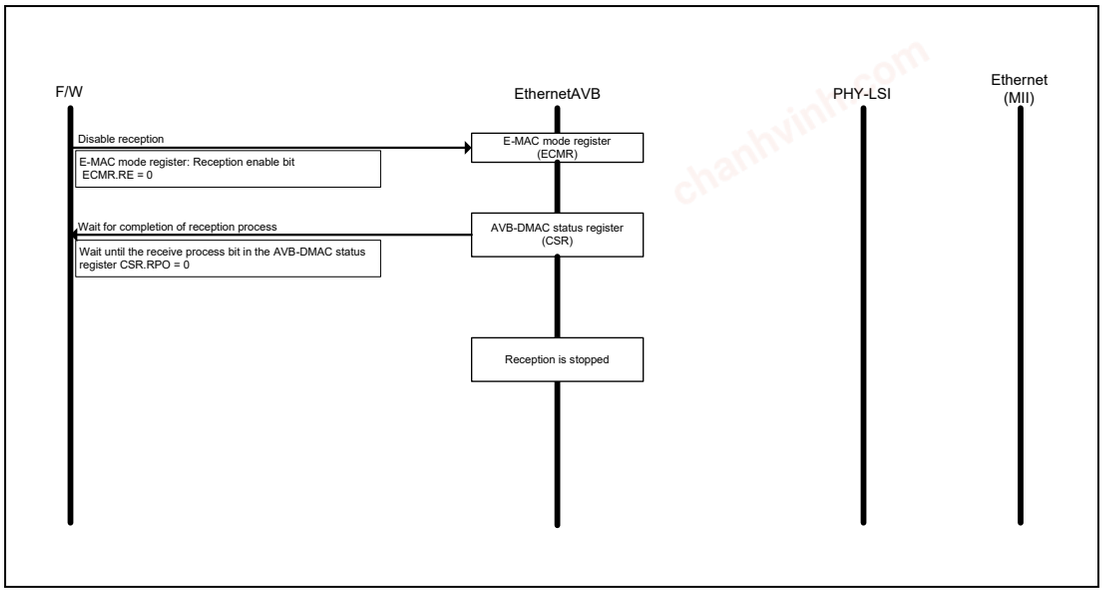


Figure 45.50 Flow for Stopping AVB-DMAC Operation in Reception

45.3.11.6 Flow for Stopping AVB-DMAC Operation in Transmission

Figure 45.51 shows the flow for stopping AVB-DMAC operation in transmission.

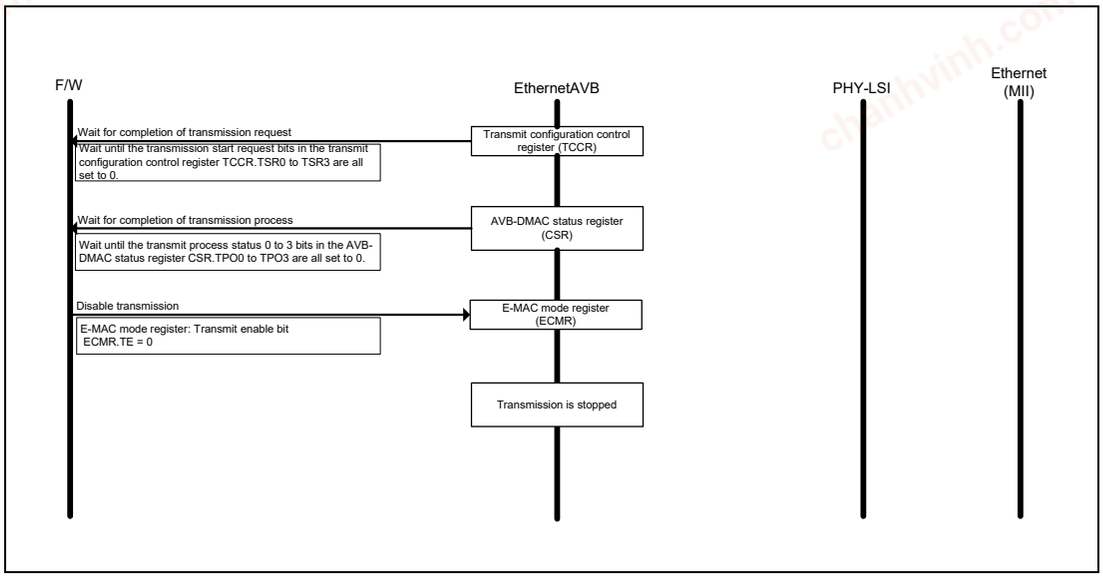


Figure 45.51 Flow for Stopping AVB-DMAC Operation in Transmission

chanhvinh.com

45.3.11.7 Flow for Stopping and Resetting the AVB-DMAC

Figure 45.52 shows the flow for stopping and resetting the AVB-DMAC.

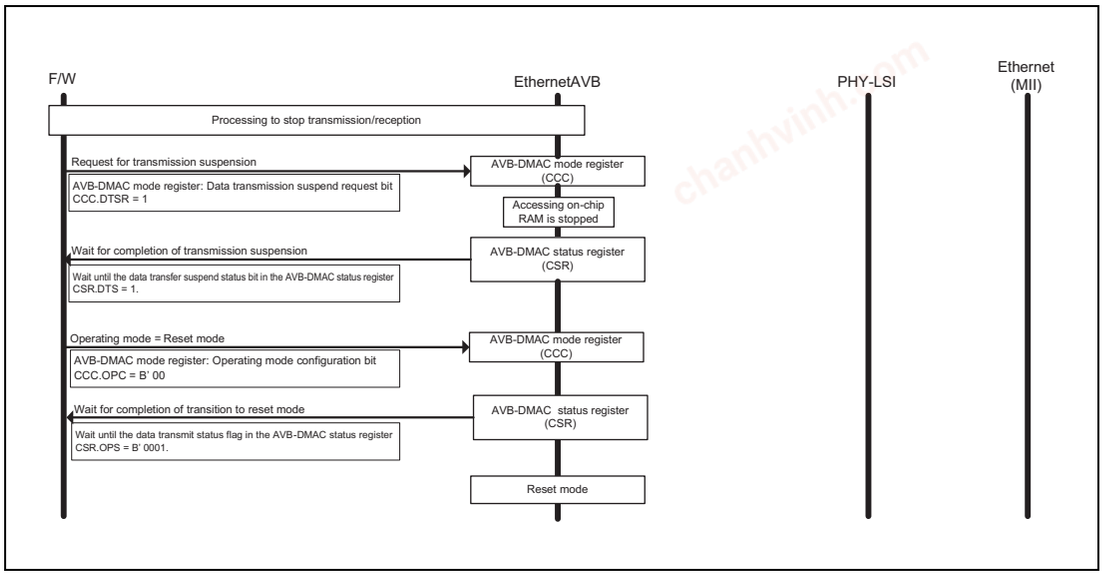


Figure 45.52 Flow for Stopping and Resetting the AVB-DMAC

45.3.11.8 Flow for Emergency Stopping the AVB-DMAC

Figure 45.53 shows the flow for emergency stopping the AVB-DMAC.

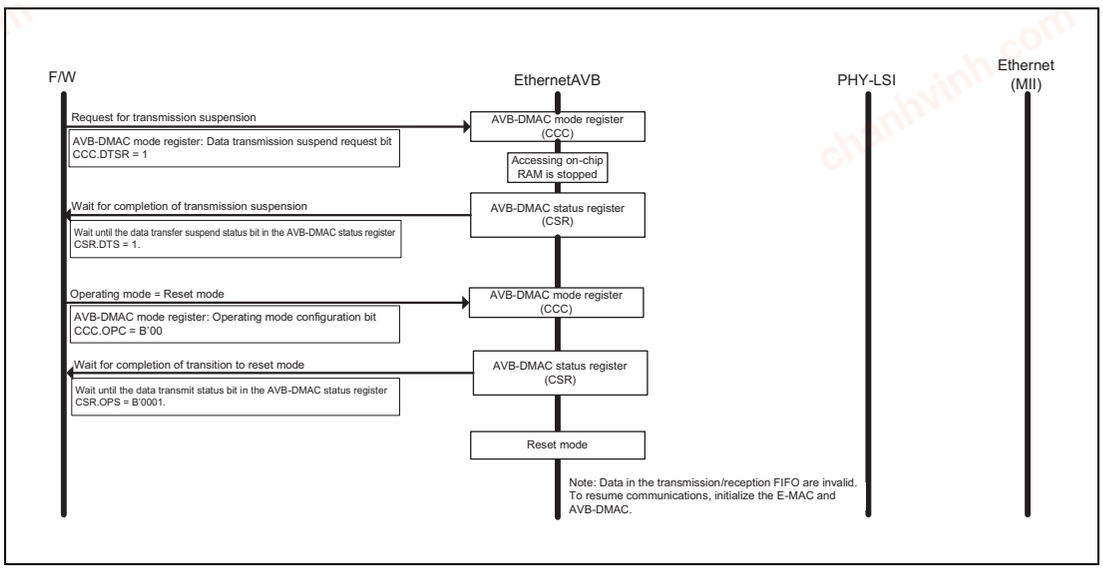


Figure 45.53 Flow for Emergency Stopping the AVB-DMAC

chanhvinh.com

45.3.11.9 Flow of gPTP Initialization

Figure 45.54 shows the flow of gPTP initialization.

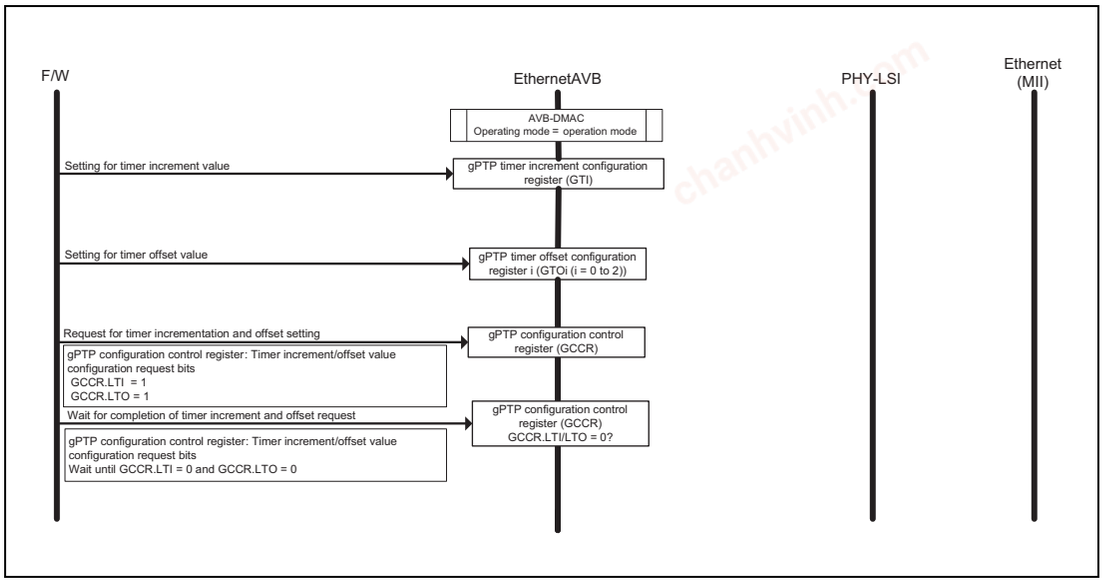


Figure 45.54 Flow of gPTP Initialization

chanhvinh.com

45.3.11.10 Flow of gPTP Time Stamping in Transmission

Figure 45.55 shows the flow of gPTP time stamping in transmission.

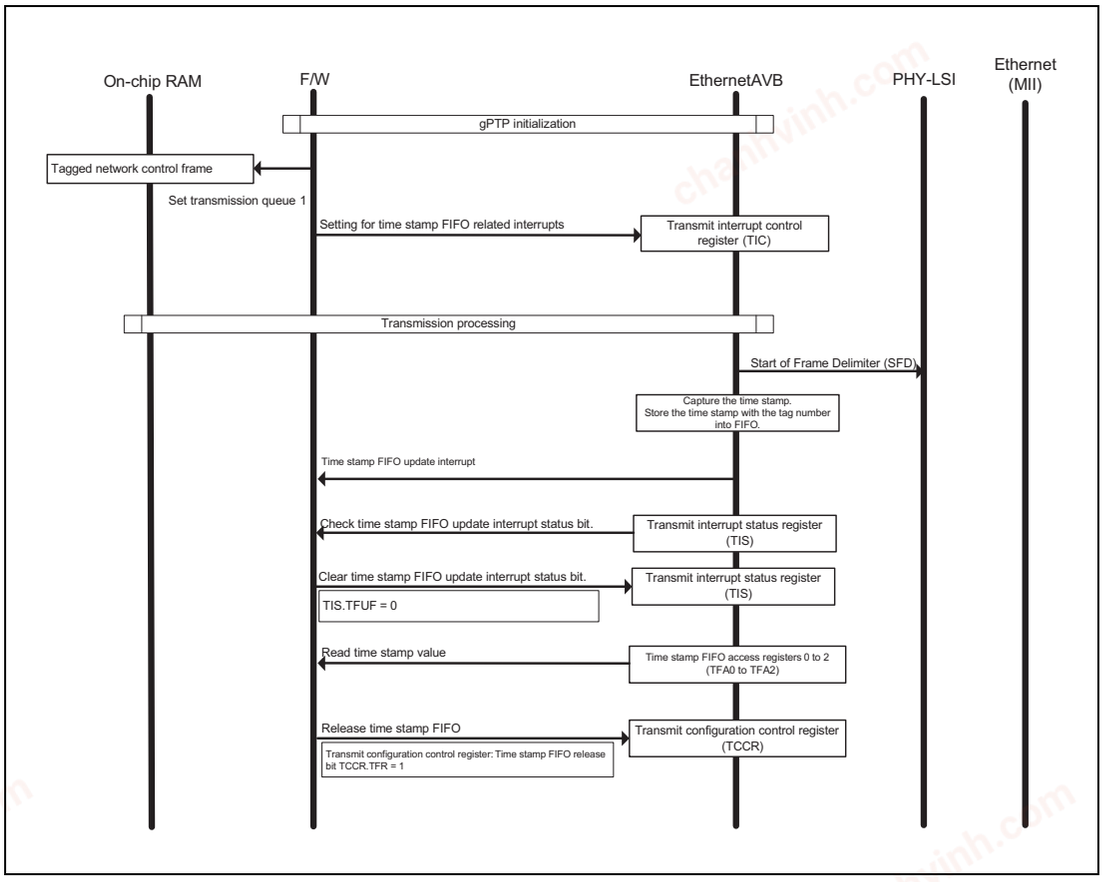


Figure 45.55 Flow of gPTP Time Stamping in Transmission

chanhvinh.com

45.3.11.11 Flow of gPTP Time Stamping and Synchronization in Reception

Figure 45.56 shows the flow of gPTP time stamping and synchronization in reception.

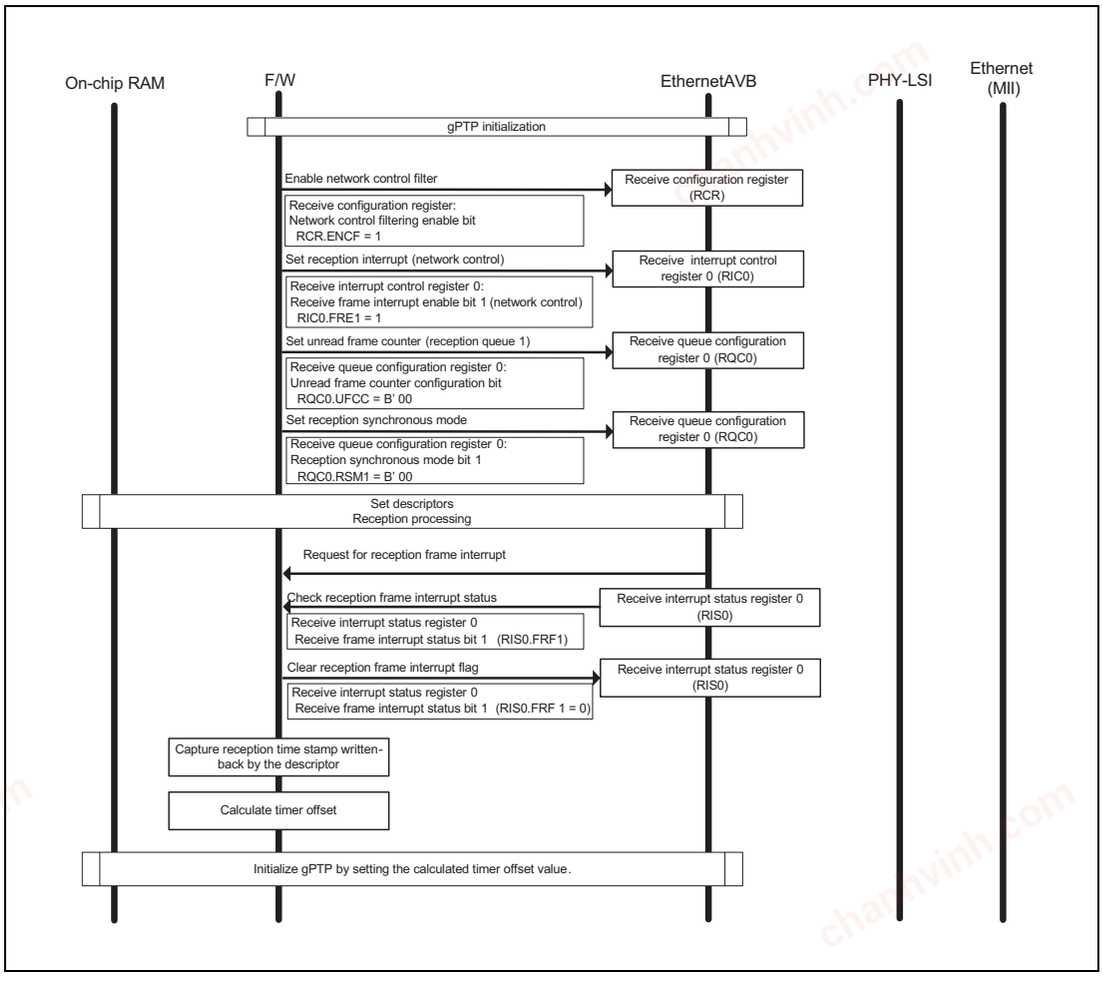


Figure 45.56 Flow of gPTP Time Stamping and Synchronization in Reception

chanhvinh.com

45.3.11.12 Flow of Capturing AVTP Presentation Times

Figure 45.57 shows the flow of capturing AVTP presentation times.

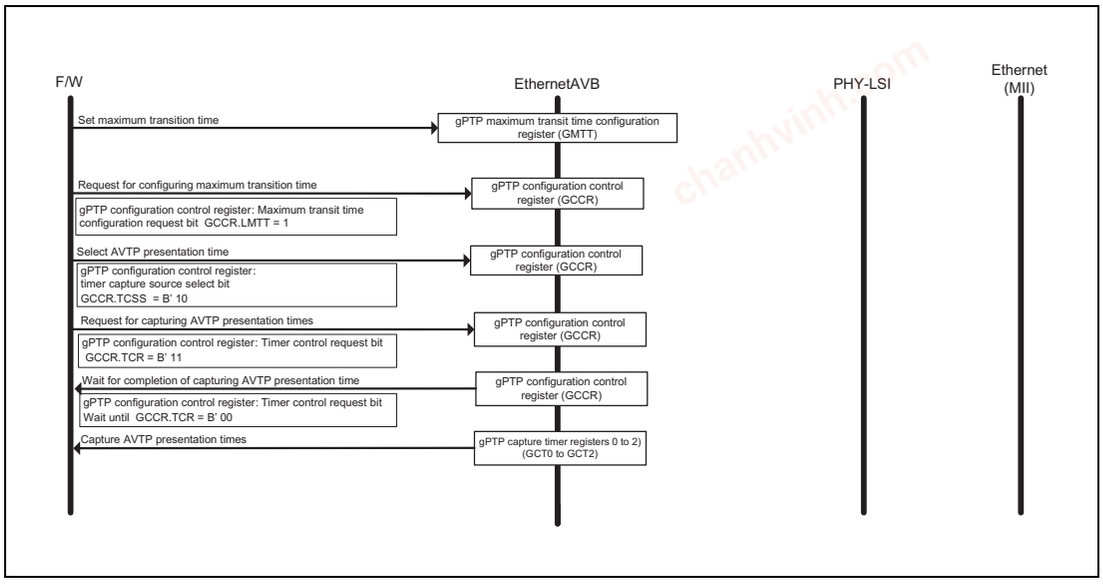


Figure 45.57 Flow of Capturing AVTP Presentation Times

45.3.11.13 Flow of AVTP Presentation Time Comparison

Figure 45.58 shows the flow of AVTP presentation time comparison.

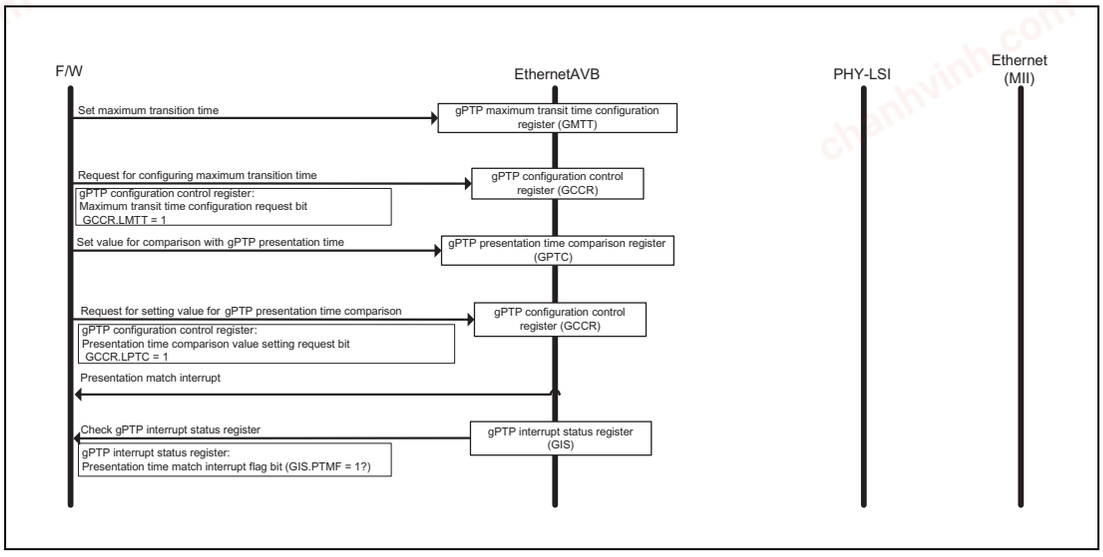


Figure 45.58 Flow of AVTP Presentation Time Comparison

chanhvinh.com

45.3.11.14 Flow of Loopback Mode Operation

Figure 45.59 shows the flow of loopback mode operation.

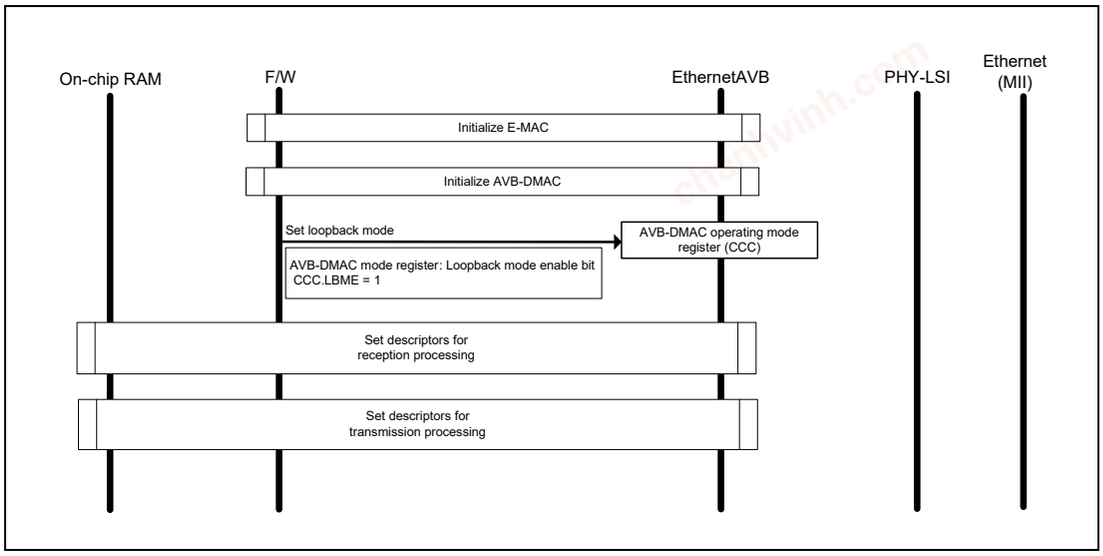


Figure 45.59 Flow of Loopback Mode Operation

chanhvinh.com

45.3.12 Connection to PHY-LSI

45.3.12.1 MII Frame Transmission/Reception Timing

Each MII frame transmission/reception timing is shown in Figure 45.60 to Figure 45.63.

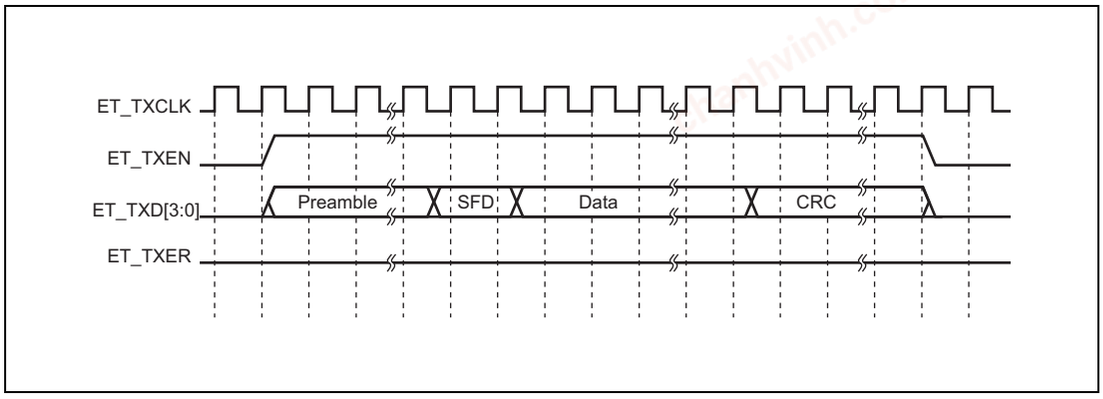


Figure 45.60 MII Frame Transmit Timing (Normal Transmission)

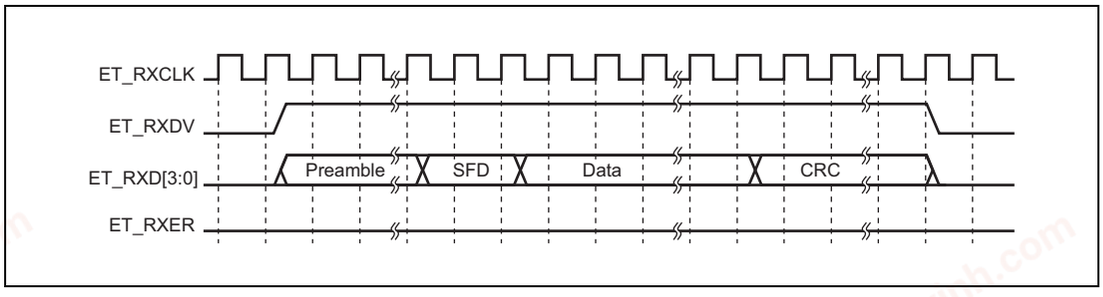


Figure 45.61 MII Frame Receive Timing (Normal Reception)

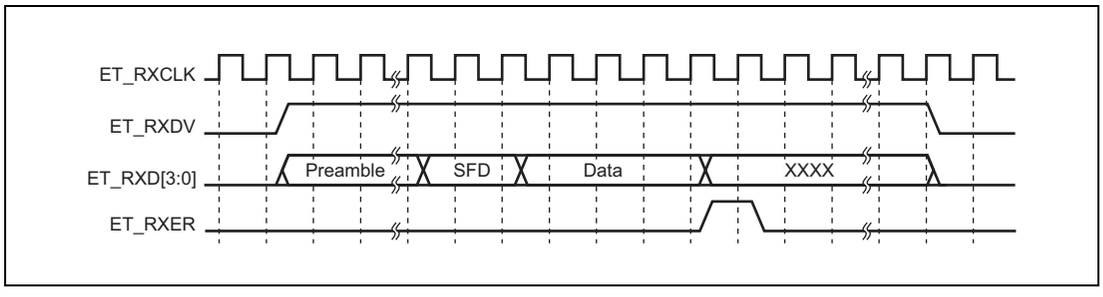


Figure 45.62 MII Frame Receive Timing (Reception Error (1))

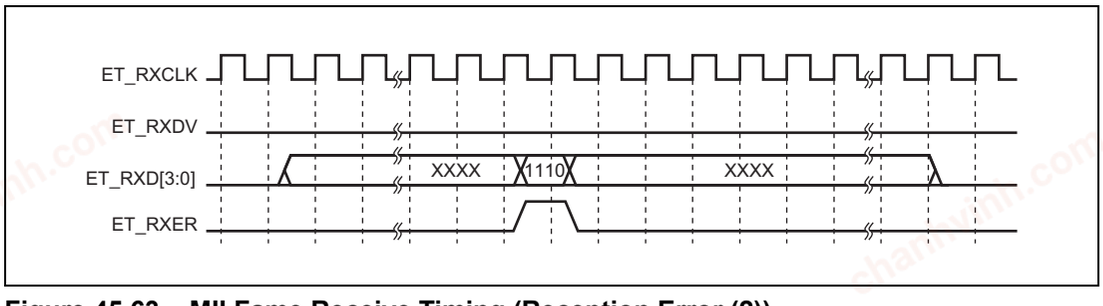


Figure 45.63 MII Fame Receive Timing (Reception Error (2))

chanhvinh.com

45.3.12.2 Accessing MII Registers

MII registers in the PHY-LSI are accessed via PIR in this LSI. PIR is used as a serial interface conforming to the MII frame format specified in IEEE802.3u.

(1) MII Management Frame Format

Figure 45.64 shows the format of an MII management frame. To access an MII register, a management frame is implemented by the program in accordance with the procedures shown in MII Register Access Procedure.

Access Type	MII Management Frame							
Item	PRE	ST	OP	PHYAD	REGAD	TA	DATA	IDLE
Number of bits	32	2	2	5	5	2	16	
Read	1..1	01	10	00001	RRRRR	Z0	D..D	
Write	1..1	01	01	00001	RRRRR	10	D..D	X

[Legend]

PRE: 32 consecutive 1s
ST: Write of 01 indicating start of frame
OP: Write of code indicating access type
PHYAD: Write of 0001 if the PHY-LSI address is 1 (sequential write starting with the MSB).
This bit changes depending on the PHY-LSI address.
REGAD: Write of 000q if the register address is 1 (sequential write starting with the MSB).
This bit changes depending on the PHY-LSI register address.
TA: Time for switching data transmission source on MII interface
(a) Write: 10 written
(b) Read: Bus release (notation: Z0) performed
DATA: 16-bit data. Sequential write or read from MSB
(a) Write: 16-bit data write
(b) Read: 16-bit data read
IDLE: Wait time until next MII management format input
(a) Write: Independent bus release (notation: X) performed
(d) Read: Bus already released in TA: control unnecessary

Figure 45.64 MII Management Frame Format

chanhvinh.com

(2) MII Register Access Procedure

The program accesses MII registers via PIR. Access is implemented by a combination of 1-bit-unit data write, 1-bit-unit data read, bus release, and independent bus release. Figure 45.65 to Figure 45.68 show the MII register access timing. The timing will differ depending on the PHY-LSI type.

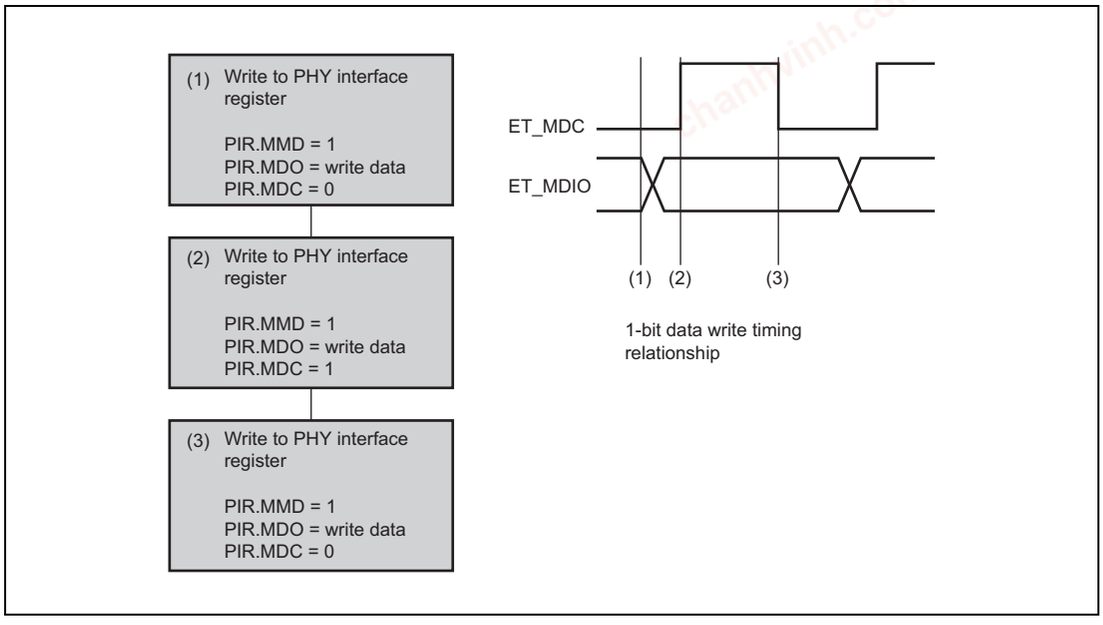


Figure 45.65 1-Bit Data Write Flowchart

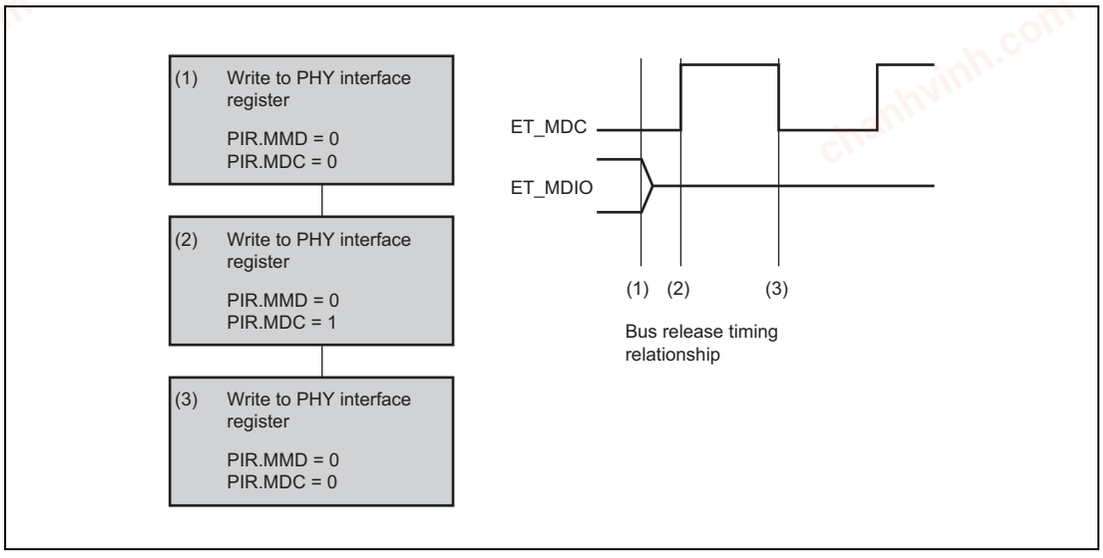


Figure 45.66 Bus Release Flowchart (TA in Read in Figure 45.64)

chanhvinh.com

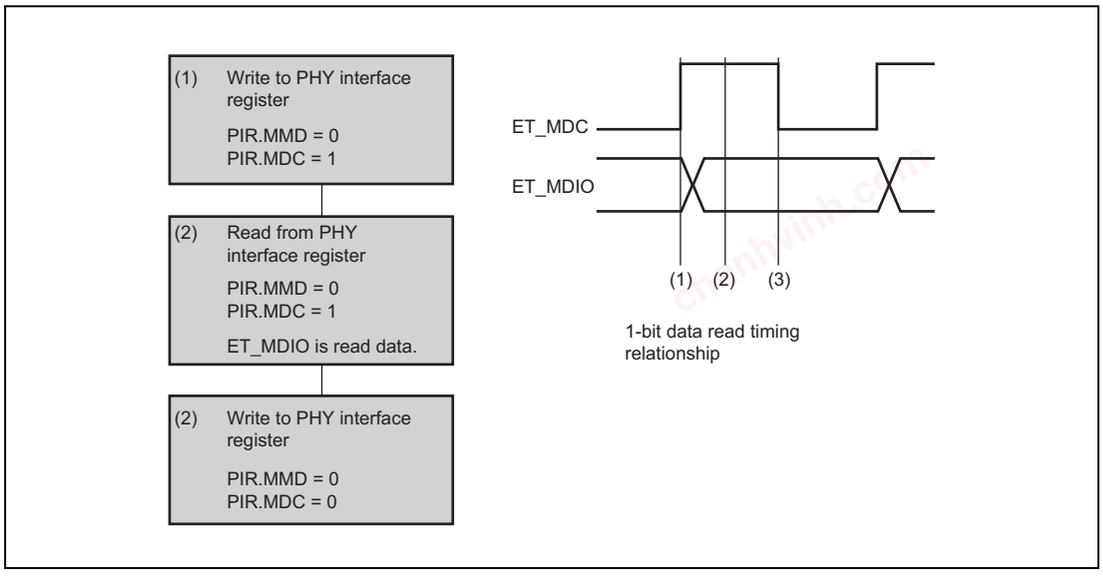


Figure 45.67 1-Bit Data Read Flowchart

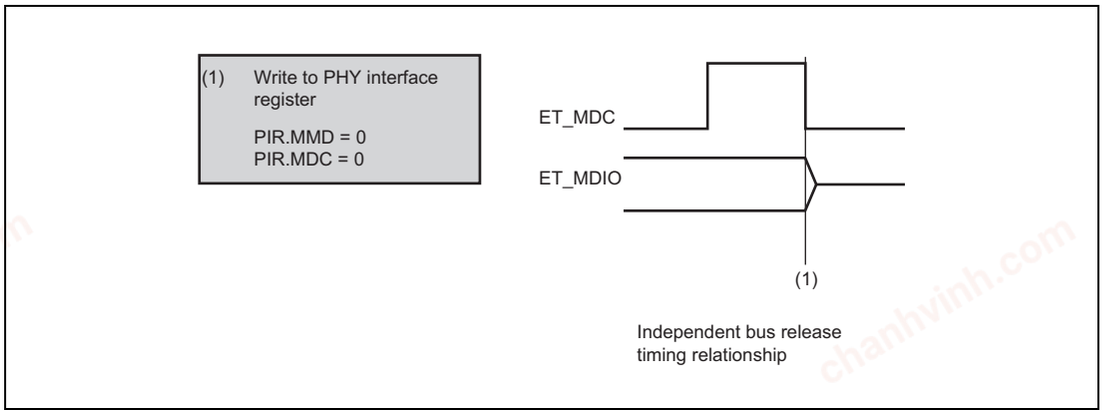


Figure 45.68 Independent Bus Release Flowchart (IDLE in Write in Figure 45.64)

chanhvinh.com

45.3.13 Usage Notes

45.3.13.1 Checksum Calculation of Ethernet Frames

This LSI is capable of calculating the checksum data of the received frames. Only the data fields of the Ethernet frames are subject to checksum calculation. Specifically, a data field follows the length/type field and is followed by the CRC field. Figure 45.69 shows schematics indicating which parts of the Ethernet frames are calculated. Calculation involves 16-bit addition only; it does not involve bit inversion. Note that when the checksum data is valid, the CRC data (4 bytes) is not transferred as a receive frame, and the checksum data (sum data) is added automatically. Figure 45.70 shows schematics of Ethernet frames to which the checksum data has been added.

CAUTION

Also for the frames with VLANtag inserted, the 15th byte from the top and the following bytes before the CRC field are subject to calculation.

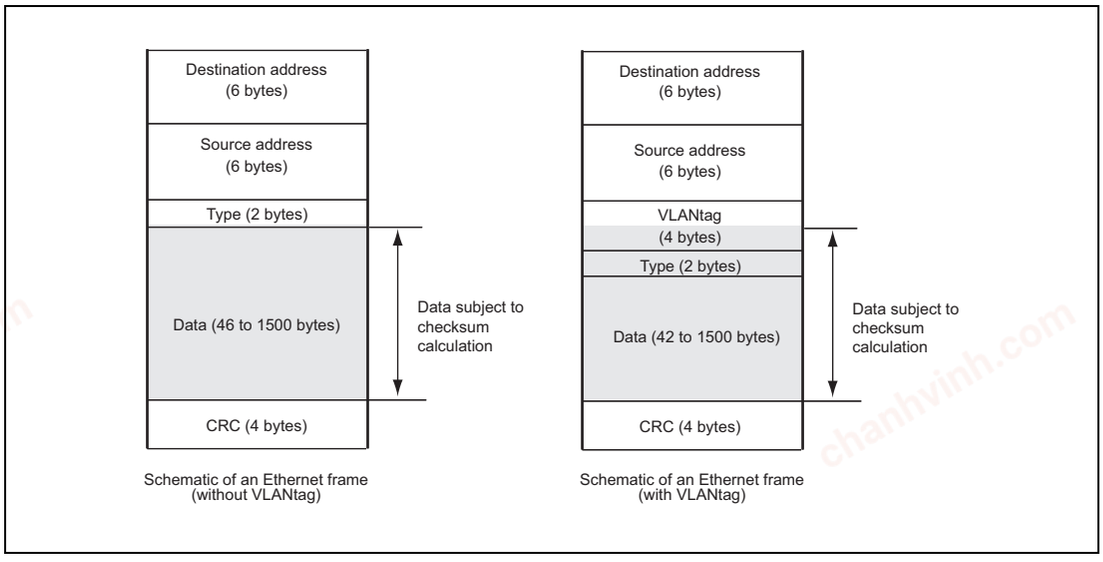


Figure 45.69 Data Subject to Checksum Calculation

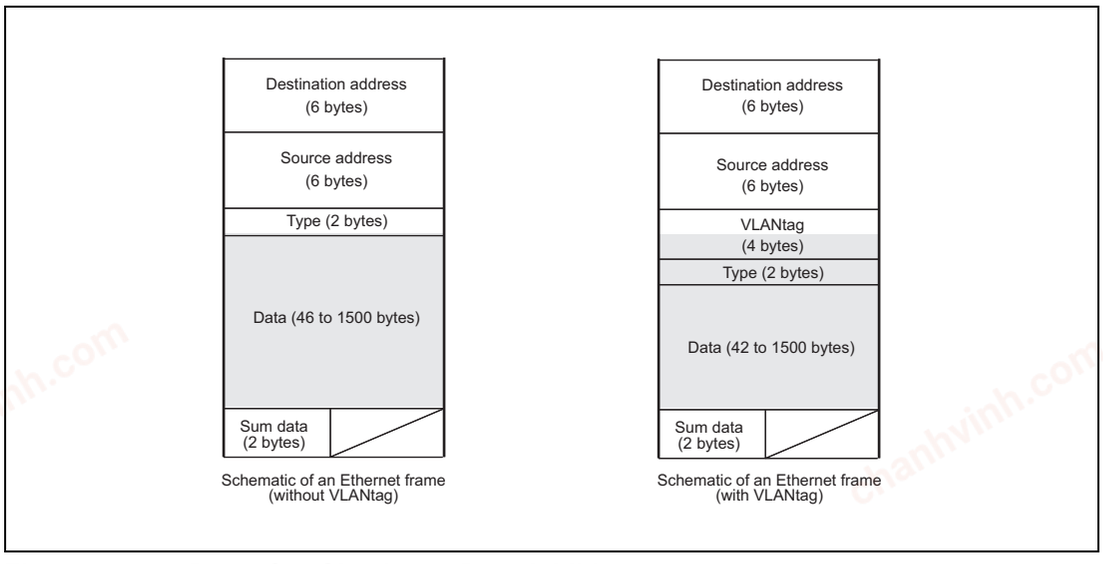


Figure 45.70 Data after Checksum Data Addition

chanhvinh.com

46. List of Registers

46.1 Register Addresses

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Secondary cache	Cache ID Register	reg0_cache_id	32	H'3FFFF000	32
	Cache Type Register	reg0_cache_type	32	H'3FFFF004	32
	Control Register	reg1_control	32	H'3FFFF100	32
	Auxiliary Control Register	reg1_aux_control	32	H'3FFFF104	32
	Tag RAM Latency Control Register	reg1_tag_ram_control	32	H'3FFFF108	32
	Data RAM Latency Control Register	reg1_data_ram_control	32	H'3FFFF10C	32
	Event Counter Control Register	reg2_ev_counter_ctrl	32	H'3FFFF200	32
	Event Counter Configuration Register 1	reg2_ev_counter1_cfg	32	H'3FFFF204	32
	Event Counter Configuration Register 0	reg2_ev_counter0_cfg	32	H'3FFFF208	32
	Event counter value register 1	reg2_ev_counter1	32	H'3FFFF20C	32
	Event counter value register 0	reg2_ev_counter0	32	H'3FFFF210	32
	Interrupt Mask Register	reg2_int_mask	32	H'3FFFF214	32
	Masked Interrupt Status Register	reg2_int_mask_status	32	H'3FFFF218	32
	Raw Interrupt Status Register	reg2_int_raw_status	32	H'3FFFF21C	32
	Interrupt Clear Register	reg2_int_clear	32	H'3FFFF220	32
	Cache Sync Register	reg7_cache_sync	32	H'3FFFF730	32
	Invalidate Line by PA Register	reg7_inv_pa	32	H'3FFFF770	32
	Invalidate by Way Register	reg7_inv_way	32	H'3FFFF77C	32
	Clean Line by PA Register	reg7_clean_pa	32	H'3FFFF7B0	32
	Clean Line by Set/Way Register	reg7_clean_index	32	H'3FFFF7B8	32
	Clean by Way Register	reg7_clean_way	32	H'3FFFF7BC	32
	Clean and Invalidate Line by PA Register	reg7_clean_inv_pa	32	H'3FFFF7F0	32
	Clean and Invalidate Line by Set/Way Register	reg7_clean_inv_index	32	H'3FFFF7F8	32
	Clean and Invalidate by Way Register	reg7_clean_inv_way	32	H'3FFFF7FC	32
	Data Lockdown 0 Register	reg9_d_lockdown0	32	H'3FFFF900	32
	Instruction Lockdown 0 Register	reg9_i_lockdown0	32	H'3FFFF904	32
	Data Lockdown 1 Register	reg9_d_lockdown1	32	H'3FFFF908	32
	Instruction Lockdown 1 Register	reg9_i_lockdown1	32	H'3FFFF90C	32
	Data Lockdown 2 Register	reg9_d_lockdown2	32	H'3FFFF910	32
	Instruction Lockdown 2 Register	reg9_i_lockdown2	32	H'3FFFF914	32
	Data Lockdown 3 Register	reg9_d_lockdown3	32	H'3FFFF918	32
	Instruction Lockdown 3 Register	reg9_i_lockdown3	32	H'3FFFF91C	32
	Data Lockdown 4 Register	reg9_d_lockdown4	32	H'3FFFF920	32
	Instruction Lockdown 4 Register	reg9_i_lockdown4	32	H'3FFFF924	32
	Data Lockdown 5 Register	reg9_d_lockdown5	32	H'3FFFF928	32
	Instruction Lockdown 5 Register	reg9_i_lockdown5	32	H'3FFFF92C	32
	Data Lockdown 6 Register	reg9_d_lockdown6	32	H'3FFFF930	32
	Instruction Lockdown 6 Register	reg9_i_lockdown6	32	H'3FFFF934	32
	Data Lockdown 7 Register	reg9_d_lockdown7	32	H'3FFFF938	32
	Instruction Lockdown 7 Register	reg9_i_lockdown7	32	H'3FFFF93C	32
	Lockdown by Line Enable Register	reg9_lock_line_en	32	H'3FFFF950	32
	Unlock All Lines Register	reg9_unlock_way	32	H'3FFFF954	32
	Address Filtering Start Register	reg12_addr_filtering_start	32	H'3FFFFC00	32
	Address Filtering End Register	reg12_addr_filtering_end	32	H'3FFFFC04	32
	Debug Control Register	reg15_debug_ctrl	32	H'3FFFFF40	32
	Prefetch Control Register	reg15_prefetch_ctrl	32	H'3FFFFF60	32
	Power Control Register	reg15_power_ctrl	32	H'3FFFFF80	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
LSI internal bus	Remap register	RMPR	32	H'FCFE1A00	32
	AXI bus control register 0	AXIBUSCTL0	32	H'FCFE1A04	32
	AXI bus control register 2	AXIBUSCTL2	32	H'FCFE1A0C	32
	AXI bus control register 5*1	AXIBUSCTL5	32	H'FCFE1A18	32
	AXI bus control register 6	AXIBUSCTL6	32	H'FCFE1A1C	32
	AXI bus control register 7	AXIBUSCTL7	32	H'FCFE1A20	32
	AXI bus response error interrupt control register 0	AXIRERRCTL0	32	H'FCFE1A30	32
	AXI bus response error interrupt control register 2	AXIRERRCTL2	32	H'FCFE1A38	32
	AXI bus response error status register 0	AXIRERRST0	32	H'FCFE1A40	32
	AXI bus response error status register 2	AXIRERRST2	32	H'FCFE1A48	32
	AXI bus response error clear register 0	AXIRERRCLR0	32	H'FCFE1A50	32
	AXI bus response error clear register 2	AXIRERRCLR2	32	H'FCFE1A58	32
Clock pulse generator	Frequency control register	FRQCR	16	H'FCFE0010	16
Interrupt controller	Interrupt control register 0	ICR0	16	H'FCFEF800	16
	Interrupt control register 1	ICR1	16	H'FCFEF802	16
	IRQ interrupt request register	IRQRR	16	H'FCFEF804	16
	Distributor control register	ICDDCR	32	H'E8201000	32
	Interrupt controller type register	ICDICTR	32	H'E8201004	32
	Distributor implementer identification register	ICDIIDR	32	H'E8201008	32
	Interrupt security register 0	ICDISR0	32	H'E8201080	32
	Interrupt security register 1	ICDISR1	32	H'E8201084	32
	Interrupt security register 2	ICDISR2	32	H'E8201088	32
	Interrupt security register 3	ICDISR3	32	H'E820108C	32
	Interrupt security register 4	ICDISR4	32	H'E8201090	32
	Interrupt security register 5	ICDISR5	32	H'E8201094	32
	Interrupt security register 6	ICDISR6	32	H'E8201098	32
	Interrupt security register 7	ICDISR7	32	H'E820109C	32
	Interrupt security register 8	ICDISR8	32	H'E82010A0	32
	Interrupt security register 9	ICDISR9	32	H'E82010A4	32
	Interrupt security register 10	ICDISR10	32	H'E82010A8	32
	Interrupt security register 11	ICDISR11	32	H'E82010AC	32
	Interrupt security register 12	ICDISR12	32	H'E82010B0	32
	Interrupt security register 13	ICDISR13	32	H'E82010B4	32
	Interrupt security register 14	ICDISR14	32	H'E82010B8	32
	Interrupt security register 15	ICDISR15	32	H'E82010BC	32
	Interrupt security register 16	ICDISR16	32	H'E82010C0	32
	Interrupt set-enable register 0	ICDISER0	32	H'E8201100	32
	Interrupt set-enable register 1	ICDISER1	32	H'E8201104	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Interrupt controller	Interrupt set-enable register 2	ICDISER2	32	H'E8201108	32
	Interrupt set-enable register 3	ICDISER3	32	H'E820110C	32
	Interrupt set-enable register 4	ICDISER4	32	H'E8201110	32
	Interrupt set-enable register 5	ICDISER5	32	H'E8201114	32
	Interrupt set-enable register 6	ICDISER6	32	H'E8201118	32
	Interrupt set-enable register 7	ICDISER7	32	H'E820111C	32
	Interrupt set-enable register 8	ICDISER8	32	H'E8201120	32
	Interrupt set-enable register 9	ICDISER9	32	H'E8201124	32
	Interrupt set-enable register 10	ICDISER10	32	H'E8201128	32
	Interrupt set-enable register 11	ICDISER11	32	H'E820112C	32
	Interrupt set-enable register 12	ICDISER12	32	H'E8201130	32
	Interrupt set-enable register 13	ICDISER13	32	H'E8201134	32
	Interrupt set-enable register 14	ICDISER14	32	H'E8201138	32
	Interrupt set-enable register 15	ICDISER15	32	H'E820113C	32
	Interrupt set-enable register 16	ICDISER16	32	H'E8201140	32
	Interrupt clear-enable register 0	ICDICER0	32	H'E8201180	32
	Interrupt clear-enable register 1	ICDICER1	32	H'E8201184	32
	Interrupt clear-enable register 2	ICDICER2	32	H'E8201188	32
	Interrupt clear-enable register 3	ICDICER3	32	H'E820118C	32
	Interrupt clear-enable register 4	ICDICER4	32	H'E8201190	32
	Interrupt clear-enable register 5	ICDICER5	32	H'E8201194	32
	Interrupt clear-enable register 6	ICDICER6	32	H'E8201198	32
	Interrupt clear-enable register 7	ICDICER7	32	H'E820119C	32
	Interrupt clear-enable register 8	ICDICER8	32	H'E82011A0	32
	Interrupt clear-enable register 9	ICDICER9	32	H'E82011A4	32
	Interrupt clear-enable register 10	ICDICER10	32	H'E82011A8	32
	Interrupt clear-enable register 11	ICDICER11	32	H'E82011AC	32
	Interrupt clear-enable register 12	ICDICER12	32	H'E82011B0	32
	Interrupt clear-enable register 13	ICDICER13	32	H'E82011B4	32
	Interrupt clear-enable register 14	ICDICER14	32	H'E82011B8	32
	Interrupt clear-enable register 15	ICDICER15	32	H'E82011BC	32
	Interrupt clear-enable register 16	ICDICER16	32	H'E82011C0	32
	Interrupt set-pending register 0	ICDISPR0	32	H'E8201200	32
	Interrupt set-pending register 1	ICDISPR1	32	H'E8201204	32
	Interrupt set-pending register 2	ICDISPR2	32	H'E8201208	32
	Interrupt set-pending register 3	ICDISPR3	32	H'E820120C	32
	Interrupt set-pending register 4	ICDISPR4	32	H'E8201210	32
	Interrupt set-pending register 5	ICDISPR5	32	H'E8201214	32
	Interrupt set-pending register 6	ICDISPR6	32	H'E8201218	32
	Interrupt set-pending register 7	ICDISPR7	32	H'E820121C	32
	Interrupt set-pending register 8	ICDISPR8	32	H'E8201220	32
	Interrupt set-pending register 9	ICDISPR9	32	H'E8201224	32
	Interrupt set-pending register 10	ICDISPR10	32	H'E8201228	32
	Interrupt set-pending register 11	ICDISPR11	32	H'E820122C	32
	Interrupt set-pending register 12	ICDISPR12	32	H'E8201230	32
	Interrupt set-pending register 13	ICDISPR13	32	H'E8201234	32
	Interrupt set-pending register 14	ICDISPR14	32	H'E8201238	32
	Interrupt set-pending register 15	ICDISPR15	32	H'E820123C	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Interrupt controller	Interrupt set-pending register 16	ICDISPR16	32	H'E8201240	32
	Interrupt clear-pending register 0	ICDICPR0	32	H'E8201280	32
	Interrupt clear-pending register 1	ICDICPR1	32	H'E8201284	32
	Interrupt clear-pending register 2	ICDICPR2	32	H'E8201288	32
	Interrupt clear-pending register 3	ICDICPR3	32	H'E820128C	32
	Interrupt clear-pending register 4	ICDICPR4	32	H'E8201290	32
	Interrupt clear-pending register 5	ICDICPR5	32	H'E8201294	32
	Interrupt clear-pending register 6	ICDICPR6	32	H'E8201298	32
	Interrupt clear-pending register 7	ICDICPR7	32	H'E820129C	32
	Interrupt clear-pending register 8	ICDICPR8	32	H'E82012A0	32
	Interrupt clear-pending register 9	ICDICPR9	32	H'E82012A4	32
	Interrupt clear-pending register 10	ICDICPR10	32	H'E82012A8	32
	Interrupt clear-pending register 11	ICDICPR11	32	H'E82012AC	32
	Interrupt clear-pending register 12	ICDICPR12	32	H'E82012B0	32
	Interrupt clear-pending register 13	ICDICPR13	32	H'E82012B4	32
	Interrupt clear-pending register 14	ICDICPR14	32	H'E82012B8	32
	Interrupt clear-pending register 15	ICDICPR15	32	H'E82012BC	32
	Interrupt clear-pending register 16	ICDICPR16	32	H'E82012C0	32
	Active bit register 0	ICDABR0	32	H'E8201300	32
	Active bit register 1	ICDABR1	32	H'E8201304	32
	Active bit register 2	ICDABR2	32	H'E8201308	32
	Active bit register 3	ICDABR3	32	H'E820130C	32
	Active bit register 4	ICDABR4	32	H'E8201310	32
	Active bit register 5	ICDABR5	32	H'E8201314	32
	Active bit register 6	ICDABR6	32	H'E8201318	32
	Active bit register 7	ICDABR7	32	H'E820131C	32
	Active bit register 8	ICDABR8	32	H'E8201320	32
	Active bit register 9	ICDABR9	32	H'E8201324	32
	Active bit register 10	ICDABR10	32	H'E8201328	32
	Active bit register 11	ICDABR11	32	H'E820132C	32
	Active bit register 12	ICDABR12	32	H'E8201330	32
	Active bit register 13	ICDABR13	32	H'E8201334	32
	Active bit register 14	ICDABR14	32	H'E8201338	32
	Active bit register 15	ICDABR15	32	H'E820133C	32
	Active bit register 16	ICDABR16	32	H'E8201340	32
	Interrupt priority register 0	ICDIPR0	32	H'E8201400	32
	Interrupt priority register 1	ICDIPR1	32	H'E8201404	32
	Interrupt priority register 2	ICDIPR2	32	H'E8201408	32
	Interrupt priority register 3	ICDIPR3	32	H'E820140C	32
	Interrupt priority register 4	ICDIPR4	32	H'E8201410	32
	Interrupt priority register 5	ICDIPR5	32	H'E8201414	32
	Interrupt priority register 6	ICDIPR6	32	H'E8201418	32
	Interrupt priority register 7	ICDIPR7	32	H'E820141C	32
	Interrupt priority register 8	ICDIPR8	32	H'E8201420	32
	Interrupt priority register 9	ICDIPR9	32	H'E8201424	32
	Interrupt priority register 10	ICDIPR10	32	H'E8201428	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Interrupt controller	Interrupt priority register 11	ICDIPR11	32	H'E820142C	32
	Interrupt priority register 12	ICDIPR12	32	H'E8201430	32
	Interrupt priority register 13	ICDIPR13	32	H'E8201434	32
	Interrupt priority register 14	ICDIPR14	32	H'E8201438	32
	Interrupt priority register 15	ICDIPR15	32	H'E820143C	32
	Interrupt priority register 16	ICDIPR16	32	H'E8201440	32
	Interrupt priority register 17	ICDIPR17	32	H'E8201444	32
	Interrupt priority register 18	ICDIPR18	32	H'E8201448	32
	Interrupt priority register 19	ICDIPR19	32	H'E820144C	32
	Interrupt priority register 20	ICDIPR20	32	H'E8201450	32
	Interrupt priority register 21	ICDIPR21	32	H'E8201454	32
	Interrupt priority register 22	ICDIPR22	32	H'E8201458	32
	Interrupt priority register 23	ICDIPR23	32	H'E820145C	32
	Interrupt priority register 24	ICDIPR24	32	H'E8201460	32
	Interrupt priority register 25	ICDIPR25	32	H'E8201464	32
	Interrupt priority register 26	ICDIPR26	32	H'E8201468	32
	Interrupt priority register 27	ICDIPR27	32	H'E820146C	32
	Interrupt priority register 28	ICDIPR28	32	H'E8201470	32
	Interrupt priority register 29	ICDIPR29	32	H'E8201474	32
	Interrupt priority register 30	ICDIPR30	32	H'E8201478	32
	Interrupt priority register 31	ICDIPR31	32	H'E820147C	32
	Interrupt priority register 32	ICDIPR32	32	H'E8201480	32
	Interrupt priority register 33	ICDIPR33	32	H'E8201484	32
	Interrupt priority register 34	ICDIPR34	32	H'E8201488	32
	Interrupt priority register 35	ICDIPR35	32	H'E820148C	32
	Interrupt priority register 36	ICDIPR36	32	H'E8201490	32
	Interrupt priority register 37	ICDIPR37	32	H'E8201494	32
	Interrupt priority register 38	ICDIPR38	32	H'E8201498	32
	Interrupt priority register 39	ICDIPR39	32	H'E820149C	32
	Interrupt priority register 40	ICDIPR40	32	H'E82014A0	32
	Interrupt priority register 41	ICDIPR41	32	H'E82014A4	32
	Interrupt priority register 42	ICDIPR42	32	H'E82014A8	32
	Interrupt priority register 43	ICDIPR43	32	H'E82014AC	32
	Interrupt priority register 44	ICDIPR44	32	H'E82014B0	32
	Interrupt priority register 45	ICDIPR45	32	H'E82014B4	32
	Interrupt priority register 46	ICDIPR46	32	H'E82014B8	32
	Interrupt priority register 47	ICDIPR47	32	H'E82014BC	32
	Interrupt priority register 48	ICDIPR48	32	H'E82014C0	32
	Interrupt priority register 49	ICDIPR49	32	H'E82014C4	32
	Interrupt priority register 50	ICDIPR50	32	H'E82014C8	32
	Interrupt priority register 51	ICDIPR51	32	H'E82014CC	32
	Interrupt priority register 52	ICDIPR52	32	H'E82014D0	32
	Interrupt priority register 53	ICDIPR53	32	H'E82014D4	32
	Interrupt priority register 54	ICDIPR54	32	H'E82014D8	32
	Interrupt priority register 55	ICDIPR55	32	H'E82014DC	32
	Interrupt priority register 56	ICDIPR56	32	H'E82014E0	32
	Interrupt priority register 57	ICDIPR57	32	H'E82014E4	32
	Interrupt priority register 58	ICDIPR58	32	H'E82014E8	32
	Interrupt priority register 59	ICDIPR59	32	H'E82014EC	32
	Interrupt priority register 60	ICDIPR60	32	H'E82014F0	32
	Interrupt priority register 61	ICDIPR61	32	H'E82014F4	32
	Interrupt priority register 62	ICDIPR62	32	H'E82014F8	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Interrupt controller	Interrupt priority register 63	ICDIPR63	32	H'E82014FC	32
	Interrupt priority register 64	ICDIPR64	32	H'E8201500	32
	Interrupt priority register 65	ICDIPR65	32	H'E8201504	32
	Interrupt priority register 66	ICDIPR66	32	H'E8201508	32
	Interrupt priority register 67	ICDIPR67	32	H'E820150C	32
	Interrupt priority register 68	ICDIPR68	32	H'E8201510	32
	Interrupt priority register 69	ICDIPR69	32	H'E8201514	32
	Interrupt priority register 70	ICDIPR70	32	H'E8201518	32
	Interrupt priority register 71	ICDIPR71	32	H'E820151C	32
	Interrupt priority register 72	ICDIPR72	32	H'E8201520	32
	Interrupt priority register 73	ICDIPR73	32	H'E8201524	32
	Interrupt priority register 74	ICDIPR74	32	H'E8201528	32
	Interrupt priority register 75	ICDIPR75	32	H'E820152C	32
	Interrupt priority register 76	ICDIPR76	32	H'E8201530	32
	Interrupt priority register 77	ICDIPR77	32	H'E8201534	32
	Interrupt priority register 78	ICDIPR78	32	H'E8201538	32
	Interrupt priority register 79	ICDIPR79	32	H'E820153C	32
	Interrupt priority register 80	ICDIPR80	32	H'E8201540	32
	Interrupt priority register 81	ICDIPR81	32	H'E8201544	32
	Interrupt priority register 82	ICDIPR82	32	H'E8201548	32
	Interrupt priority register 83	ICDIPR83	32	H'E820154C	32
	Interrupt priority register 84	ICDIPR84	32	H'E8201550	32
	Interrupt priority register 85	ICDIPR85	32	H'E8201554	32
	Interrupt priority register 86	ICDIPR86	32	H'E8201558	32
	Interrupt priority register 87	ICDIPR87	32	H'E820155C	32
	Interrupt priority register 88	ICDIPR88	32	H'E8201560	32
	Interrupt priority register 89	ICDIPR89	32	H'E8201564	32
	Interrupt priority register 90	ICDIPR90	32	H'E8201568	32
	Interrupt priority register 91	ICDIPR91	32	H'E820156C	32
	Interrupt priority register 92	ICDIPR92	32	H'E8201570	32
	Interrupt priority register 93	ICDIPR93	32	H'E8201574	32
	Interrupt priority register 94	ICDIPR94	32	H'E8201578	32
	Interrupt priority register 95	ICDIPR95	32	H'E820157C	32
	Interrupt priority register 96	ICDIPR96	32	H'E8201580	32
	Interrupt priority register 97	ICDIPR97	32	H'E8201584	32
	Interrupt priority register 98	ICDIPR98	32	H'E8201588	32
	Interrupt priority register 99	ICDIPR99	32	H'E820158C	32
	Interrupt priority register 100	ICDIPR100	32	H'E8201590	32
	Interrupt priority register 101	ICDIPR101	32	H'E8201594	32
	Interrupt priority register 102	ICDIPR102	32	H'E8201598	32
	Interrupt priority register 103	ICDIPR103	32	H'E820159C	32
	Interrupt priority register 104	ICDIPR104	32	H'E82015A0	32
	Interrupt priority register 105	ICDIPR105	32	H'E82015A4	32
	Interrupt priority register 106	ICDIPR106	32	H'E82015A8	32
	Interrupt priority register 107	ICDIPR107	32	H'E82015AC	32
	Interrupt priority register 108	ICDIPR108	32	H'E82015B0	32
	Interrupt priority register 109	ICDIPR109	32	H'E82015B4	32
	Interrupt priority register 110	ICDIPR110	32	H'E82015B8	32
	Interrupt priority register 111	ICDIPR111	32	H'E82015BC	32
	Interrupt priority register 112	ICDIPR112	32	H'E82015C0	32
	Interrupt priority register 113	ICDIPR113	32	H'E82015C4	32
	Interrupt priority register 114	ICDIPR114	32	H'E82015C8	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Interrupt controller	Interrupt priority register 115	ICDIPR115	32	H'E82015CC	32
	Interrupt priority register 116	ICDIPR116	32	H'E82015D0	32
	Interrupt priority register 117	ICDIPR117	32	H'E82015D4	32
	Interrupt priority register 118	ICDIPR118	32	H'E82015D8	32
	Interrupt priority register 119	ICDIPR119	32	H'E82015DC	32
	Interrupt priority register 120	ICDIPR120	32	H'E82015E0	32
	Interrupt priority register 121	ICDIPR121	32	H'E82015E4	32
	Interrupt priority register 122	ICDIPR122	32	H'E82015E8	32
	Interrupt priority register 123	ICDIPR123	32	H'E82015EC	32
	Interrupt priority register 124	ICDIPR124	32	H'E82015F0	32
	Interrupt priority register 125	ICDIPR125	32	H'E82015F4	32
	Interrupt priority register 126	ICDIPR126	32	H'E82015F8	32
	Interrupt priority register 127	ICDIPR127	32	H'E82015FC	32
	Interrupt priority register 128	ICDIPR128	32	H'E8201600	32
	Interrupt priority register 129	ICDIPR129	32	H'E8201604	32
	Interrupt priority register 130	ICDIPR130	32	H'E8201608	32
	Interrupt priority register 131	ICDIPR131	32	H'E820160C	32
	Interrupt priority register 132	ICDIPR132	32	H'E8201610	32
	Interrupt priority register 133	ICDIPR133	32	H'E8201614	32
	Interrupt priority register 134	ICDIPR134	32	H'E8201618	32
	Interrupt processor target register 0	ICDIPTR0	32	H'E8201800	32
	Interrupt processor target register 1	ICDIPTR1	32	H'E8201804	32
	Interrupt processor target register 2	ICDIPTR2	32	H'E8201808	32
	Interrupt processor target register 3	ICDIPTR3	32	H'E820180C	32
	Interrupt processor target register 4	ICDIPTR4	32	H'E8201810	32
	Interrupt processor target register 5	ICDIPTR5	32	H'E8201814	32
	Interrupt processor target register 6	ICDIPTR6	32	H'E8201818	32
	Interrupt processor target register 7	ICDIPTR7	32	H'E820181C	32
	Interrupt processor target register 8	ICDIPTR8	32	H'E8201820	32
	Interrupt processor target register 9	ICDIPTR9	32	H'E8201824	32
	Interrupt processor target register 10	ICDIPTR10	32	H'E8201828	32
	Interrupt processor target register 11	ICDIPTR11	32	H'E820182C	32
	Interrupt processor target register 12	ICDIPTR12	32	H'E8201830	32
	Interrupt processor target register 13	ICDIPTR13	32	H'E8201834	32
	Interrupt processor target register 14	ICDIPTR14	32	H'E8201838	32
	Interrupt processor target register 15	ICDIPTR15	32	H'E820183C	32
	Interrupt processor target register 16	ICDIPTR16	32	H'E8201840	32
	Interrupt processor target register 17	ICDIPTR17	32	H'E8201844	32
	Interrupt processor target register 18	ICDIPTR18	32	H'E8201848	32
	Interrupt processor target register 19	ICDIPTR19	32	H'E820184C	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Interrupt controller	Interrupt processor target register 20	ICDIPTR20	32	H'E8201850	32
	Interrupt processor target register 21	ICDIPTR21	32	H'E8201854	32
	Interrupt processor target register 22	ICDIPTR22	32	H'E8201858	32
	Interrupt processor target register 23	ICDIPTR23	32	H'E820185C	32
	Interrupt processor target register 24	ICDIPTR24	32	H'E8201860	32
	Interrupt processor target register 25	ICDIPTR25	32	H'E8201864	32
	Interrupt processor target register 26	ICDIPTR26	32	H'E8201868	32
	Interrupt processor target register 27	ICDIPTR27	32	H'E820186C	32
	Interrupt processor target register 28	ICDIPTR28	32	H'E8201870	32
	Interrupt processor target register 29	ICDIPTR29	32	H'E8201874	32
	Interrupt processor target register 30	ICDIPTR30	32	H'E8201878	32
	Interrupt processor target register 31	ICDIPTR31	32	H'E820187C	32
	Interrupt processor target register 32	ICDIPTR32	32	H'E8201880	32
	Interrupt processor target register 33	ICDIPTR33	32	H'E8201884	32
	Interrupt processor target register 34	ICDIPTR34	32	H'E8201888	32
	Interrupt processor target register 35	ICDIPTR35	32	H'E820188C	32
	Interrupt processor target register 36	ICDIPTR36	32	H'E8201890	32
	Interrupt processor target register 37	ICDIPTR37	32	H'E8201894	32
	Interrupt processor target register 38	ICDIPTR38	32	H'E8201898	32
	Interrupt processor target register 39	ICDIPTR39	32	H'E820189C	32
	Interrupt processor target register 40	ICDIPTR40	32	H'E82018A0	32
	Interrupt processor target register 41	ICDIPTR41	32	H'E82018A4	32
	Interrupt processor target register 42	ICDIPTR42	32	H'E82018A8	32
	Interrupt processor target register 43	ICDIPTR43	32	H'E82018AC	32
	Interrupt processor target register 44	ICDIPTR44	32	H'E82018B0	32
	Interrupt processor target register 45	ICDIPTR45	32	H'E82018B4	32
	Interrupt processor target register 46	ICDIPTR46	32	H'E82018B8	32
	Interrupt processor target register 47	ICDIPTR47	32	H'E82018BC	32
	Interrupt processor target register 48	ICDIPTR48	32	H'E82018C0	32
	Interrupt processor target register 49	ICDIPTR49	32	H'E82018C4	32
	Interrupt processor target register 50	ICDIPTR50	32	H'E82018C8	32
	Interrupt processor target register 51	ICDIPTR51	32	H'E82018CC	32
	Interrupt processor target register 52	ICDIPTR52	32	H'E82018D0	32
	Interrupt processor target register 53	ICDIPTR53	32	H'E82018D4	32
	Interrupt processor target register 54	ICDIPTR54	32	H'E82018D8	32
	Interrupt processor target register 55	ICDIPTR55	32	H'E82018DC	32
	Interrupt processor target register 56	ICDIPTR56	32	H'E82018E0	32
	Interrupt processor target register 57	ICDIPTR57	32	H'E82018E4	32
	Interrupt processor target register 58	ICDIPTR58	32	H'E82018E8	32
	Interrupt processor target register 59	ICDIPTR59	32	H'E82018EC	32
	Interrupt processor target register 60	ICDIPTR60	32	H'E82018F0	32
	Interrupt processor target register 61	ICDIPTR61	32	H'E82018F4	32
	Interrupt processor target register 62	ICDIPTR62	32	H'E82018F8	32
	Interrupt processor target register 63	ICDIPTR63	32	H'E82018FC	32
	Interrupt processor target register 64	ICDIPTR64	32	H'E8201900	32
	Interrupt processor target register 65	ICDIPTR65	32	H'E8201904	32
	Interrupt processor target register 66	ICDIPTR66	32	H'E8201908	32
	Interrupt processor target register 67	ICDIPTR67	32	H'E820190C	32
	Interrupt processor target register 68	ICDIPTR68	32	H'E8201910	32
	Interrupt processor target register 69	ICDIPTR69	32	H'E8201914	32
	Interrupt processor target register 70	ICDIPTR70	32	H'E8201918	32
	Interrupt processor target register 71	ICDIPTR71	32	H'E820191C	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Interrupt controller	Interrupt processor target register 72	ICDIPTR72	32	H'E8201920	32
	Interrupt processor target register 73	ICDIPTR73	32	H'E8201924	32
	Interrupt processor target register 74	ICDIPTR74	32	H'E8201928	32
	Interrupt processor target register 75	ICDIPTR75	32	H'E820192C	32
	Interrupt processor target register 76	ICDIPTR76	32	H'E8201930	32
	Interrupt processor target register 77	ICDIPTR77	32	H'E8201934	32
	Interrupt processor target register 78	ICDIPTR78	32	H'E8201938	32
	Interrupt processor target register 79	ICDIPTR79	32	H'E820193C	32
	Interrupt processor target register 80	ICDIPTR80	32	H'E8201940	32
	Interrupt processor target register 81	ICDIPTR81	32	H'E8201944	32
	Interrupt processor target register 82	ICDIPTR82	32	H'E8201948	32
	Interrupt processor target register 83	ICDIPTR83	32	H'E820194C	32
	Interrupt processor target register 84	ICDIPTR84	32	H'E8201950	32
	Interrupt processor target register 85	ICDIPTR85	32	H'E8201954	32
	Interrupt processor target register 86	ICDIPTR86	32	H'E8201958	32
	Interrupt processor target register 87	ICDIPTR87	32	H'E820195C	32
	Interrupt processor target register 88	ICDIPTR88	32	H'E8201960	32
	Interrupt processor target register 89	ICDIPTR89	32	H'E8201964	32
	Interrupt processor target register 90	ICDIPTR90	32	H'E8201968	32
	Interrupt processor target register 91	ICDIPTR91	32	H'E820196C	32
	Interrupt processor target register 92	ICDIPTR92	32	H'E8201970	32
	Interrupt processor target register 93	ICDIPTR93	32	H'E8201974	32
	Interrupt processor target register 94	ICDIPTR94	32	H'E8201978	32
	Interrupt processor target register 95	ICDIPTR95	32	H'E820197C	32
	Interrupt processor target register 96	ICDIPTR96	32	H'E8201980	32
	Interrupt processor target register 97	ICDIPTR97	32	H'E8201984	32
	Interrupt processor target register 98	ICDIPTR98	32	H'E8201988	32
	Interrupt processor target register 99	ICDIPTR99	32	H'E820198C	32
	Interrupt processor target register 100	ICDIPTR100	32	H'E8201990	32
	Interrupt processor target register 101	ICDIPTR101	32	H'E8201994	32
	Interrupt processor target register 102	ICDIPTR102	32	H'E8201998	32
	Interrupt processor target register 103	ICDIPTR103	32	H'E820199C	32
	Interrupt processor target register 104	ICDIPTR104	32	H'E82019A0	32
	Interrupt processor target register 105	ICDIPTR105	32	H'E82019A4	32
	Interrupt processor target register 106	ICDIPTR106	32	H'E82019A8	32
	Interrupt processor target register 107	ICDIPTR107	32	H'E82019AC	32
	Interrupt processor target register 108	ICDIPTR108	32	H'E82019B0	32
	Interrupt processor target register 109	ICDIPTR109	32	H'E82019B4	32
	Interrupt processor target register 110	ICDIPTR110	32	H'E82019B8	32
	Interrupt processor target register 111	ICDIPTR111	32	H'E82019BC	32
	Interrupt processor target register 112	ICDIPTR112	32	H'E82019C0	32
	Interrupt processor target register 113	ICDIPTR113	32	H'E82019C4	32
	Interrupt processor target register 114	ICDIPTR114	32	H'E82019C8	32
	Interrupt processor target register 115	ICDIPTR115	32	H'E82019CC	32
	Interrupt processor target register 116	ICDIPTR116	32	H'E82019D0	32
	Interrupt processor target register 117	ICDIPTR117	32	H'E82019D4	32
	Interrupt processor target register 118	ICDIPTR118	32	H'E82019D8	32
	Interrupt processor target register 119	ICDIPTR119	32	H'E82019DC	32
	Interrupt processor target register 120	ICDIPTR120	32	H'E82019E0	32
	Interrupt processor target register 121	ICDIPTR121	32	H'E82019E4	32
	Interrupt processor target register 122	ICDIPTR122	32	H'E82019E8	32
	Interrupt processor target register 123	ICDIPTR123	32	H'E82019EC	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Interrupt controller	Interrupt processor target register 124	ICDIPTR124	32	H'E82019F0	32
	Interrupt processor target register 125	ICDIPTR125	32	H'E82019F4	32
	Interrupt processor target register 126	ICDIPTR126	32	H'E82019F8	32
	Interrupt processor target register 127	ICDIPTR127	32	H'E82019FC	32
	Interrupt processor target register 128	ICDIPTR128	32	H'E8201A00	32
	Interrupt processor target register 129	ICDIPTR129	32	H'E8201A04	32
	Interrupt processor target register 130	ICDIPTR130	32	H'E8201A08	32
	Interrupt processor target register 131	ICDIPTR131	32	H'E8201A0C	32
	Interrupt processor target register 132	ICDIPTR132	32	H'E8201A10	32
	Interrupt processor target register 133	ICDIPTR133	32	H'E8201A14	32
	Interrupt processor target register 134	ICDIPTR134	32	H'E8201A18	32
	Interrupt configuration register 0	ICDICFR0	32	H'E8201C00	32
	Interrupt configuration register 1	ICDICFR1	32	H'E8201C04	32
	Interrupt configuration register 2	ICDICFR2	32	H'E8201C08	32
	Interrupt configuration register 3	ICDICFR3	32	H'E8201C0C	32
	Interrupt configuration register 4	ICDICFR4	32	H'E8201C10	32
	Interrupt configuration register 5	ICDICFR5	32	H'E8201C14	32
	Interrupt configuration register 6	ICDICFR6	32	H'E8201C18	32
	Interrupt configuration register 7	ICDICFR7	32	H'E8201C1C	32
	Interrupt configuration register 8	ICDICFR8	32	H'E8201C20	32
	Interrupt configuration register 9	ICDICFR9	32	H'E8201C24	32
	Interrupt configuration register 10	ICDICFR10	32	H'E8201C28	32
	Interrupt configuration register 11	ICDICFR11	32	H'E8201C2C	32
	Interrupt configuration register 12	ICDICFR12	32	H'E8201C30	32
	Interrupt configuration register 13	ICDICFR13	32	H'E8201C34	32
	Interrupt configuration register 14	ICDICFR14	32	H'E8201C38	32
	Interrupt configuration register 15	ICDICFR15	32	H'E8201C3C	32
	Interrupt configuration register 16	ICDICFR16	32	H'E8201C40	32
	Interrupt configuration register 17	ICDICFR17	32	H'E8201C44	32
	Interrupt configuration register 18	ICDICFR18	32	H'E8201C48	32
	Interrupt configuration register 19	ICDICFR19	32	H'E8201C4C	32
	Interrupt configuration register 20	ICDICFR20	32	H'E8201C50	32
	Interrupt configuration register 21	ICDICFR21	32	H'E8201C54	32
	Interrupt configuration register 22	ICDICFR22	32	H'E8201C58	32
	Interrupt configuration register 23	ICDICFR23	32	H'E8201C5C	32
	Interrupt configuration register 24	ICDICFR24	32	H'E8201C60	32
	Interrupt configuration register 25	ICDICFR25	32	H'E8201C64	32
	Interrupt configuration register 26	ICDICFR26	32	H'E8201C68	32
	Interrupt configuration register 27	ICDICFR27	32	H'E8201C6C	32
	Interrupt configuration register 28	ICDICFR28	32	H'E8201C70	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Interrupt controller	Interrupt configuration register 29	ICDICFR29	32	H'E8201C74	32
	Interrupt configuration register 30	ICDICFR30	32	H'E8201C78	32
	Interrupt configuration register 31	ICDICFR31	32	H'E8201C7C	32
	Interrupt configuration register 32	ICDICFR32	32	H'E8201C80	32
	Interrupt configuration register 33	ICDICFR33	32	H'E8201C84	32
	PPI status register	ppi_status	32	H'E8201D00	32
	SPI status register 0	spi_status0	32	H'E8201D04	32
	SPI status register 1	spi_status1	32	H'E8201D08	32
	SPI status register 2	spi_status2	32	H'E8201D0C	32
	SPI status register 3	spi_status3	32	H'E8201D10	32
	SPI status register 4	spi_status4	32	H'E8201D14	32
	SPI status register 5	spi_status5	32	H'E8201D18	32
	SPI status register 6	spi_status6	32	H'E8201D1C	32
	SPI status register 7	spi_status7	32	H'E8201D20	32
	SPI status register 8	spi_status8	32	H'E8201D24	32
	SPI status register 9	spi_status9	32	H'E8201D28	32
	SPI status register 10	spi_status10	32	H'E8201D2C	32
	SPI status register 11	spi_status11	32	H'E8201D30	32
	SPI status register 12	spi_status12	32	H'E8201D34	32
	SPI status register 13	spi_status13	32	H'E8201D38	32
	SPI status register 14	spi_status14	32	H'E8201D3C	32
	Software generation interrupt register	ICDSGIR	32	H'E8201F00	32
	CPU interface control register	ICCICR	32	H'E8202000	32
	Interrupt priority mask register	ICCPMR	32	H'E8202004	32
	Binary point register	ICCBPR	32	H'E8202008	32
	Interrupt acknowledge register	ICCIAR	32	H'E820200C	32
	End-of-interrupt register	ICCEOIR	32	H'E8202010	32
	Running priority register	ICCRPR	32	H'E8202014	32
	Highest pending interrupt register	ICCHPIR	32	H'E8202018	32
	Aliased binary point register	ICCABPR	32	H'E820201C	32
	CPU interface implementer identification register	ICCIIDR	32	H'E82020FC	32
Bus state controller	Common control register	CMNCR	32	H'3FFFC000	32
	CS0 space bus control register	CS0BCR	32	H'3FFFC004	32
	CS1 space bus control register	CS1BCR	32	H'3FFFC008	32
	CS2 space bus control register	CS2BCR	32	H'3FFFC00C	32
	CS3 space bus control register	CS3BCR	32	H'3FFFC010	32
	CS4 space bus control register	CS4BCR	32	H'3FFFC014	32
	CS5 space bus control register	CS5BCR	32	H'3FFFC018	32
	CS0 space wait control register	CS0WCR	32	H'3FFFC028	32
	CS1 space wait control register	CS1WCR	32	H'3FFFC02C	32
	CS2 space wait control register	CS2WCR	32	H'3FFFC030	32
	CS3 space wait control register	CS3WCR	32	H'3FFFC034	32
	CS4 space wait control register	CS4WCR	32	H'3FFFC038	32
	CS5 space wait control register	CS5WCR	32	H'3FFFC03C	32
	SDRAM control register	SDCR	32	H'3FFFC04C	32
	Refresh timer control/status register	RTCSR	16	H'3FFFC050	32
	Refresh timer counter	RTCNT	16	H'3FFFC054	32
	Refresh time constant register	RTCOR	16	H'3FFFC058	32
	Timeout cycle constant register 0	TOSCOR0	32	H'3FFFC060	32
	Timeout cycle constant register 1	TOSCOR1	32	H'3FFFC064	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Bus state controller	Timeout cycle constant register 2	TOSCOR2	32	H'3FFFC068	32
	Timeout cycle constant register 3	TOSCOR3	32	H'3FFFC06C	32
	Timeout cycle constant register 4	TOSCOR4	32	H'3FFFC070	32
	Timeout cycle constant register 5	TOSCOR5	32	H'3FFFC074	32
	Timeout status register	TOSTR	32	H'3FFFC080	32
	Timeout enable register	TOENR	32	H'3FFFC084	32
Direct memory access controller	Next0 Source Address Register 0	N0SA_0	32	H'E8200000	32
	Next0 Destination Address Register 0	N0DA_0	32	H'E8200004	32
	Next0 Transaction Byte Register 0	N0TB_0	32	H'E8200008	32
	Next1 Source Address Register 0	N1SA_0	32	H'E820000C	32
	Next1 Destination Address Register 0	N1DA_0	32	H'E8200010	32
	Next1 Transaction Byte Register 0	N1TB_0	32	H'E8200014	32
	Current Source Address Register 0	CRSA_0	32	H'E8200018	32
	Current Destination Address Register 0	CRDA_0	32	H'E820001C	32
	Current Transaction Byte Register 0	CRTB_0	32	H'E8200020	32
	Channel Status Register 0	CHSTAT_0	32	H'E8200024	32
	Channel Control Register 0	CHCTRL_0	32	H'E8200028	32
	Channel Configuration Register 0	CHCFG_0	32	H'E820002C	32
	Channel Interval Register 0	CHITVL_0	32	H'E8200030	32
	Channel Extension Register 0	CHEXT_0	32	H'E8200034	32
	Next Link Address Register 0	NXLA_0	32	H'E8200038	32
	Current Link Address Register 0	CRLA_0	32	H'E820003C	32
	Next0 Source Address Register 1	N0SA_1	32	H'E8200040	32
	Next0 Destination Address Register 1	N0DA_1	32	H'E8200044	32
	Next0 Transaction Byte Register 1	N0TB_1	32	H'E8200048	32
	Next1 Source Address Register 1	N1SA_1	32	H'E820004C	32
	Next1 Destination Address Register 1	N1DA_1	32	H'E8200050	32
	Next1 Transaction Byte Register 1	N1TB_1	32	H'E8200054	32
	Current Source Address Register 1	CRSA_1	32	H'E8200058	32
	Current Destination Address Register 1	CRDA_1	32	H'E820005C	32
	Current Transaction Byte Register 1	CRTB_1	32	H'E8200060	32
	Channel Status Register 1	CHSTAT_1	32	H'E8200064	32
	Channel Control Register 1	CHCTRL_1	32	H'E8200068	32
	Channel Configuration Register 1	CHCFG_1	32	H'E820006C	32
	Channel Interval Register 1	CHITVL_1	32	H'E8200070	32
	Channel Extension Register 1	CHEXT_1	32	H'E8200074	32
	Next Link Address Register 1	NXLA_1	32	H'E8200078	32
	Current Link Address Register 1	CRLA_1	32	H'E820007C	32
	Next0 Source Address Register 2	N0SA_2	32	H'E8200080	32
	Next0 Destination Address Register 2	N0DA_2	32	H'E8200084	32
	Next0 Transaction Byte Register 2	N0TB_2	32	H'E8200088	32
	Next1 Source Address Register 2	N1SA_2	32	H'E820008C	32
	Next1 Destination Address Register 2	N1DA_2	32	H'E8200090	32
	Next1 Transaction Byte Register 2	N1TB_2	32	H'E8200094	32
	Current Source Address Register 2	CRSA_2	32	H'E8200098	32
	Current Destination Address Register 2	CRDA_2	32	H'E820009C	32
	Current Transaction Byte Register 2	CRTB_2	32	H'E82000A0	32
	Channel Status Register 2	CHSTAT_2	32	H'E82000A4	32
	Channel Control Register 2	CHCTRL_2	32	H'E82000A8	32
	Channel Configuration Register 2	CHCFG_2	32	H'E82000AC	32
	Channel Interval Register 2	CHITVL_2	32	H'E82000B0	32
	Channel Extension Register 2	CHEXT_2	32	H'E82000B4	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Direct memory access controller	Next Link Address Register 2	NXLA_2	32	H'E82000B8	32
	Current Link Address Register 2	CRLA_2	32	H'E82000BC	32
	Next0 Source Address Register 3	N0SA_3	32	H'E82000C0	32
	Next0 Destination Address Register 3	N0DA_3	32	H'E82000C4	32
	Next0 Transaction Byte Register 3	N0TB_3	32	H'E82000C8	32
	Next1 Source Address Register 3	N1SA_3	32	H'E82000CC	32
	Next1 Destination Address Register 3	N1DA_3	32	H'E82000D0	32
	Next1 Transaction Byte Register 3	N1TB_3	32	H'E82000D4	32
	Current Source Address Register 3	CRSA_3	32	H'E82000D8	32
	Current Destination Address Register 3	CRDA_3	32	H'E82000DC	32
	Current Transaction Byte Register 3	CRTB_3	32	H'E82000E0	32
	Channel Status Register 3	CHSTAT_3	32	H'E82000E4	32
	Channel Control Register 3	CHCTRL_3	32	H'E82000E8	32
	Channel Configuration Register 3	CHCFG_3	32	H'E82000EC	32
	Channel Interval Register 3	CHITVL_3	32	H'E82000F0	32
	Channel Extension Register 3	CHEXT_3	32	H'E82000F4	32
	Next Link Address Register 3	NXLA_3	32	H'E82000F8	32
	Current Link Address Register 3	CRLA_3	32	H'E82000FC	32
	Next0 Source Address Register 4	N0SA_4	32	H'E8200100	32
	Next0 Destination Address Register 4	N0DA_4	32	H'E8200104	32
	Next0 Transaction Byte Register 4	N0TB_4	32	H'E8200108	32
	Next1 Source Address Register 4	N1SA_4	32	H'E820010C	32
	Next1 Destination Address Register 4	N1DA_4	32	H'E8200110	32
	Next1 Transaction Byte Register 4	N1TB_4	32	H'E8200114	32
	Current Source Address Register 4	CRSA_4	32	H'E8200118	32
	Current Destination Address Register 4	CRDA_4	32	H'E820011C	32
	Current Transaction Byte Register 4	CRTB_4	32	H'E8200120	32
	Channel Status Register 4	CHSTAT_4	32	H'E8200124	32
	Channel Control Register 4	CHCTRL_4	32	H'E8200128	32
	Channel Configuration Register 4	CHCFG_4	32	H'E820012C	32
	Channel Interval Register 4	CHITVL_4	32	H'E8200130	32
	Channel Extension Register 4	CHEXT_4	32	H'E8200134	32
	Next Link Address Register 4	NXLA_4	32	H'E8200138	32
	Current Link Address Register 4	CRLA_4	32	H'E820013C	32
	Next0 Source Address Register 5	N0SA_5	32	H'E8200140	32
	Next0 Destination Address Register 5	N0DA_5	32	H'E8200144	32
	Next0 Transaction Byte Register 5	N0TB_5	32	H'E8200148	32
	Next1 Source Address Register 5	N1SA_5	32	H'E820014C	32
	Next1 Destination Address Register 5	N1DA_5	32	H'E8200150	32
	Next1 Transaction Byte Register 5	N1TB_5	32	H'E8200154	32
	Current Source Address Register 5	CRSA_5	32	H'E8200158	32
	Current Destination Address Register 5	CRDA_5	32	H'E820015C	32
	Current Transaction Byte Register 5	CRTB_5	32	H'E8200160	32
	Channel Status Register 5	CHSTAT_5	32	H'E8200164	32
	Channel Control Register 5	CHCTRL_5	32	H'E8200168	32
	Channel Configuration Register 5	CHCFG_5	32	H'E820016C	32
	Channel Interval Register 5	CHITVL_5	32	H'E8200170	32
	Channel Extension Register 5	CHEXT_5	32	H'E8200174	32
	Next Link Address Register 5	NXLA_5	32	H'E8200178	32
	Current Link Address Register 5	CRLA_5	32	H'E820017C	32
	Next0 Source Address Register 6	N0SA_6	32	H'E8200180	32
	Next0 Destination Address Register 6	N0DA_6	32	H'E8200184	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Direct memory access controller	Next0 Transaction Byte Register 6	N0TB_6	32	H'E8200188	32
	Next1 Source Address Register 6	N1SA_6	32	H'E820018C	32
	Next1 Destination Address Register 6	N1DA_6	32	H'E8200190	32
	Next1 Transaction Byte Register 6	N1TB_6	32	H'E8200194	32
	Current Source Address Register 6	CRSA_6	32	H'E8200198	32
	Current Destination Address Register 6	CRDA_6	32	H'E820019C	32
	Current Transaction Byte Register 6	CRTB_6	32	H'E82001A0	32
	Channel Status Register 6	CHSTAT_6	32	H'E82001A4	32
	Channel Control Register 6	CHCTRL_6	32	H'E82001A8	32
	Channel Configuration Register 6	CHCFG_6	32	H'E82001AC	32
	Channel Interval Register 6	CHITVL_6	32	H'E82001B0	32
	Channel Extension Register 6	CHEXT_6	32	H'E82001B4	32
	Next Link Address Register 6	NXLA_6	32	H'E82001B8	32
	Current Link Address Register 6	CRLA_6	32	H'E82001BC	32
	Next0 Source Address Register 7	N0SA_7	32	H'E82001C0	32
	Next0 Destination Address Register 7	N0DA_7	32	H'E82001C4	32
	Next0 Transaction Byte Register 7	N0TB_7	32	H'E82001C8	32
	Next1 Source Address Register 7	N1SA_7	32	H'E82001CC	32
	Next1 Destination Address Register 7	N1DA_7	32	H'E82001D0	32
	Next1 Transaction Byte Register 7	N1TB_7	32	H'E82001D4	32
	Current Source Address Register 7	CRSA_7	32	H'E82001D8	32
	Current Destination Address Register 7	CRDA_7	32	H'E82001DC	32
	Current Transaction Byte Register 7	CRTB_7	32	H'E82001E0	32
	Channel Status Register 7	CHSTAT_7	32	H'E82001E4	32
	Channel Control Register 7	CHCTRL_7	32	H'E82001E8	32
	Channel Configuration Register 7	CHCFG_7	32	H'E82001EC	32
	Channel Interval Register 7	CHITVL_7	32	H'E82001F0	32
	Channel Extension Register 7	CHEXT_7	32	H'E82001F4	32
	Next Link Address Register 7	NXLA_7	32	H'E82001F8	32
	Current Link Address Register 7	CRLA_7	32	H'E82001FC	32
	DMA Control Registers 0-7	DCTRL_0_7	32	H'E8200300	32
	DMA Status EN Registers 0-7	DSTAT_EN_0_7	32	H'E8200310	32
	DMA Status ER Registers 0-7	DSTAT_ER_0_7	32	H'E8200314	32
	DMA Status END Registers 0-7	DSTAT_END_0_7	32	H'E8200318	32
	DMA Status TC Registers 0-7	DSTAT_TC_0_7	32	H'E820031C	32
	DMA Status SUS Registers 0-7	DSTAT_SUS_0_7	32	H'E8200320	32
	Next0 Source Address Register 8	N0SA_8	32	H'E8200400	32
	Next0 Destination Address Register 8	N0DA_8	32	H'E8200404	32
	Next0 Transaction Byte Register 8	N0TB_8	32	H'E8200408	32
	Next1 Source Address Register 8	N1SA_8	32	H'E820040C	32
	Next1 Destination Address Register 8	N1DA_8	32	H'E8200410	32
	Next1 Transaction Byte Register 8	N1TB_8	32	H'E8200414	32
	Current Source Address Register 8	CRSA_8	32	H'E8200418	32
	Current Destination Address Register 8	CRDA_8	32	H'E820041C	32
	Current Transaction Byte Register 8	CRTB_8	32	H'E8200420	32
	Channel Status Register 8	CHSTAT_8	32	H'E8200424	32
	Channel Control Register 8	CHCTRL_8	32	H'E8200428	32
	Channel Configuration Register 8	CHCFG_8	32	H'E820042C	32
	Channel Interval Register 8	CHITVL_8	32	H'E8200430	32
	Channel Extension Register 8	CHEXT_8	32	H'E8200434	32
	Next Link Address Register 8	NXLA_8	32	H'E8200438	32
	Current Link Address Register 8	CRLA_8	32	H'E820043C	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Direct memory access controller	Next0 Source Address Register 9	N0SA_9	32	H'E8200440	32
	Next0 Destination Address Register 9	N0DA_9	32	H'E8200444	32
	Next0 Transaction Byte Register 9	N0TB_9	32	H'E8200448	32
	Next1 Source Address Register 9	N1SA_9	32	H'E820044C	32
	Next1 Destination Address Register 9	N1DA_9	32	H'E8200450	32
	Next1 Transaction Byte Register 9	N1TB_9	32	H'E8200454	32
	Current Source Address Register 9	CRSA_9	32	H'E8200458	32
	Current Destination Address Register 9	CRDA_9	32	H'E820045C	32
	Current Transaction Byte Register 9	CRTB_9	32	H'E8200460	32
	Channel Status Register 9	CHSTAT_9	32	H'E8200464	32
	Channel Control Register 9	CHCTRL_9	32	H'E8200468	32
	Channel Configuration Register 9	CHCFG_9	32	H'E820046C	32
	Channel Interval Register 9	CHITVL_9	32	H'E8200470	32
	Channel Extension Register 9	CHEXT_9	32	H'E8200474	32
	Next Link Address Register 9	NXLA_9	32	H'E8200478	32
	Current Link Address Register 9	CRLA_9	32	H'E820047C	32
	Next0 Source Address Register 10	N0SA_10	32	H'E8200480	32
	Next0 Destination Address Register 10	N0DA_10	32	H'E8200484	32
	Next0 Transaction Byte Register 10	N0TB_10	32	H'E8200488	32
	Next1 Source Address Register 10	N1SA_10	32	H'E820048C	32
	Next1 Destination Address Register 10	N1DA_10	32	H'E8200490	32
	Next1 Transaction Byte Register 10	N1TB_10	32	H'E8200494	32
	Current Source Address Register 10	CRSA_10	32	H'E8200498	32
	Current Destination Address Register 10	CRDA_10	32	H'E820049C	32
	Current Transaction Byte Register 10	CRTB_10	32	H'E82004A0	32
	Channel Status Register 10	CHSTAT_10	32	H'E82004A4	32
	Channel Control Register 10	CHCTRL_10	32	H'E82004A8	32
	Channel Configuration Register 10	CHCFG_10	32	H'E82004AC	32
	Channel Interval Register 10	CHITVL_10	32	H'E82004B0	32
	Channel Extension Register 10	CHEXT_10	32	H'E82004B4	32
	Next Link Address Register 10	NXLA_10	32	H'E82004B8	32
	Current Link Address Register 10	CRLA_10	32	H'E82004BC	32
	Next0 Source Address Register 11	N0SA_11	32	H'E82004C0	32
	Next0 Destination Address Register 11	N0DA_11	32	H'E82004C4	32
	Next0 Transaction Byte Register 11	N0TB_11	32	H'E82004C8	32
	Next1 Source Address Register 11	N1SA_11	32	H'E82004CC	32
	Next1 Destination Address Register 11	N1DA_11	32	H'E82004D0	32
	Next1 Transaction Byte Register 11	N1TB_11	32	H'E82004D4	32
	Current Source Address Register 11	CRSA_11	32	H'E82004D8	32
	Current Destination Address Register 11	CRDA_11	32	H'E82004DC	32
	Current Transaction Byte Register 11	CRTB_11	32	H'E82004E0	32
	Channel Status Register 11	CHSTAT_11	32	H'E82004E4	32
	Channel Control Register 11	CHCTRL_11	32	H'E82004E8	32
	Channel Configuration Register 11	CHCFG_11	32	H'E82004EC	32
	Channel Interval Register 11	CHITVL_11	32	H'E82004F0	32
	Channel Extension Register 11	CHEXT_11	32	H'E82004F4	32
	Next Link Address Register 11	NXLA_11	32	H'E82004F8	32
	Current Link Address Register 11	CRLA_11	32	H'E82004FC	32
	Next0 Source Address Register 12	N0SA_12	32	H'E8200500	32
	Next0 Destination Address Register 12	N0DA_12	32	H'E8200504	32
	Next0 Transaction Byte Register 12	N0TB_12	32	H'E8200508	32
	Next1 Source Address Register 12	N1SA_12	32	H'E820050C	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Direct memory access controller	Next1 Destination Address Register 12	N1DA_12	32	H'E8200510	32
	Next1 Transaction Byte Register 12	N1TB_12	32	H'E8200514	32
	Current Source Address Register 12	CRSA_12	32	H'E8200518	32
	Current Destination Address Register 12	CRDA_12	32	H'E820051C	32
	Current Transaction Byte Register 12	CRTB_12	32	H'E8200520	32
	Channel Status Register 12	CHSTAT_12	32	H'E8200524	32
	Channel Control Register 12	CHCTRL_12	32	H'E8200528	32
	Channel Configuration Register 12	CHCFG_12	32	H'E820052C	32
	Channel Interval Register 12	CHITVL_12	32	H'E8200530	32
	Channel Extension Register 12	CHEXT_12	32	H'E8200534	32
	Next Link Address Register 12	NXLA_12	32	H'E8200538	32
	Current Link Address Register 12	CRLA_12	32	H'E820053C	32
	Next0 Source Address Register 13	N0SA_13	32	H'E8200540	32
	Next0 Destination Address Register 13	N0DA_13	32	H'E8200544	32
	Next0 Transaction Byte Register 13	N0TB_13	32	H'E8200548	32
	Next1 Source Address Register 13	N1SA_13	32	H'E820054C	32
	Next1 Destination Address Register 13	N1DA_13	32	H'E8200550	32
	Next1 Transaction Byte Register 13	N1TB_13	32	H'E8200554	32
	Current Source Address Register 13	CRSA_13	32	H'E8200558	32
	Current Destination Address Register 13	CRDA_13	32	H'E820055C	32
	Current Transaction Byte Register 13	CRTB_13	32	H'E8200560	32
	Channel Status Register 13	CHSTAT_13	32	H'E8200564	32
	Channel Control Register 13	CHCTRL_13	32	H'E8200568	32
	Channel Configuration Register 13	CHCFG_13	32	H'E820056C	32
	Channel Interval Register 13	CHITVL_13	32	H'E8200570	32
	Channel Extension Register 13	CHEXT_13	32	H'E8200574	32
	Next Link Address Register 13	NXLA_13	32	H'E8200578	32
	Current Link Address Register 13	CRLA_13	32	H'E820057C	32
	Next0 Source Address Register 14	N0SA_14	32	H'E8200580	32
	Next0 Destination Address Register 14	N0DA_14	32	H'E8200584	32
	Next0 Transaction Byte Register 14	N0TB_14	32	H'E8200588	32
	Next1 Source Address Register 14	N1SA_14	32	H'E820058C	32
	Next1 Destination Address Register 14	N1DA_14	32	H'E8200590	32
	Next1 Transaction Byte Register 14	N1TB_14	32	H'E8200594	32
	Current Source Address Register 14	CRSA_14	32	H'E8200598	32
	Current Destination Address Register 14	CRDA_14	32	H'E820059C	32
	Current Transaction Byte Register 14	CRTB_14	32	H'E82005A0	32
	Channel Status Register 14	CHSTAT_14	32	H'E82005A4	32
	Channel Control Register 14	CHCTRL_14	32	H'E82005A8	32
	Channel Configuration Register 14	CHCFG_14	32	H'E82005AC	32
	Channel Interval Register 14	CHITVL_14	32	H'E82005B0	32
	Channel Extension Register 14	CHEXT_14	32	H'E82005B4	32
	Next Link Address Register 14	NXLA_14	32	H'E82005B8	32
	Current Link Address Register 14	CRLA_14	32	H'E82005BC	32
	Next0 Source Address Register 15	N0SA_15	32	H'E82005C0	32
	Next0 Destination Address Register 15	N0DA_15	32	H'E82005C4	32
	Next0 Transaction Byte Register 15	N0TB_15	32	H'E82005C8	32
	Next1 Source Address Register 15	N1SA_15	32	H'E82005CC	32
	Next1 Destination Address Register 15	N1DA_15	32	H'E82005D0	32
	Next1 Transaction Byte Register 15	N1TB_15	32	H'E82005D4	32
	Current Source Address Register 15	CRSA_15	32	H'E82005D8	32
	Current Destination Address Register 15	CRDA_15	32	H'E82005DC	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Direct memory access controller	Current Transaction Byte Register 15	CRTB_15	32	H'E82005E0	32
	Channel Status Register 15	CHSTAT_15	32	H'E82005E4	32
	Channel Control Register 15	CHCTRL_15	32	H'E82005E8	32
	Channel Configuration Register 15	CHCFG_15	32	H'E82005EC	32
	Channel Interval Register 15	CHITVL_15	32	H'E82005F0	32
	Channel Extension Register 15	CHEXT_15	32	H'E82005F4	32
	Next Link Address Register 15	NXLA_15	32	H'E82005F8	32
	Current Link Address Register 15	CRLA_15	32	H'E82005FC	32
	DMA Control Registers 8-15	DCTRL_8_15	32	H'E8200700	32
	DMA Status EN Registers 8-15	DSTAT_EN_8_15	32	H'E8200710	32
	DMA Status ER Registers 8-15	DSTAT_ER_8_15	32	H'E8200714	32
	DMA Status END Registers 8-15	DSTAT_END_8_15	32	H'E8200718	32
	DMA Status TC Registers 8-15	DSTAT_TC_8_15	32	H'E820071C	32
	DMA Status SUS Registers 8-15	DSTAT_SUS_8_15	32	H'E8200720	32
	DMA extended resource selector 0	DMARS0	32	H'FCFE1000	32
	DMA extended resource selector 1	DMARS1	32	H'FCFE1004	32
	DMA extended resource selector 2	DMARS2	32	H'FCFE1008	32
	DMA extended resource selector 3	DMARS3	32	H'FCFE100C	32
	DMA extended resource selector 4	DMARS4	32	H'FCFE1010	32
	DMA extended resource selector 5	DMARS5	32	H'FCFE1014	32
	DMA extended resource selector 6	DMARS6	32	H'FCFE1018	32
	DMA extended resource selector 7	DMARS7	32	H'FCFE101C	32
Multi-function timer pulse unit 2	Timer control register_0	TCR_0	8	H'FCFF0300	8
	Timer mode register_0	TMDR_0	8	H'FCFF0301	8
	Timer I/O control register H_0	TIORH_0	8	H'FCFF0302	8
	Timer I/O control register L_0	TIORL_0	8	H'FCFF0303	8
	Timer interrupt enable register_0	TIER_0	8	H'FCFF0304	8
	Timer status register_0	TSR_0	8	H'FCFF0305	8
	Timer counter_0	TCNT_0	16	H'FCFF0306	16
	Timer general register A_0	TGRA_0	16	H'FCFF0308	16
	Timer general register B_0	TGRB_0	16	H'FCFF030A	16
	Timer general register C_0	TGRC_0	16	H'FCFF030C	16
	Timer general register D_0	TGRD_0	16	H'FCFF030E	16
	Timer general register E_0	TGRE_0	16	H'FCFF0320	16
	Timer general register F_0	TGRF_0	16	H'FCFF0322	16
	Timer interrupt enable register 2_0	TIER2_0	8	H'FCFF0324	8
	Timer status register 2_0	TSR2_0	8	H'FCFF0325	8
	Timer buffer operation transfer mode register_0	TBTM_0	8	H'FCFF0326	8
	Timer control register_1	TCR_1	8	H'FCFF0380	8
	Timer mode register_1	TMDR_1	8	H'FCFF0381	8
	Timer I/O control register _1	TIOR_1	8	H'FCFF0382	8
	Timer interrupt enable register_1	TIER_1	8	H'FCFF0384	8
	Timer status register_1	TSR_1	8	H'FCFF0385	8
	Timer counter_1	TCNT_1	16	H'FCFF0386	16
	Timer general register A_1	TGRA_1	16	H'FCFF0388	16
	Timer general register B_1	TGRB_1	16	H'FCFF038A	16
	Timer input capture control register	TICCR	8	H'FCFF0390	8
	Timer control register_2	TCR_2	8	H'FCFF0000	8
	Timer mode register_2	TMDR_2	8	H'FCFF0001	8
	Timer I/O control register _2	TIOR_2	8	H'FCFF0002	8
	Timer interrupt enable register_2	TIER_2	8	H'FCFF0004	8
	Timer status register_2	TSR_2	8	H'FCFF0005	8

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Multi-function timer pulse unit 2	Timer counter_2	TCNT_2	16	H'FCFF0006	16
	Timer general register A_2	TGRA_2	16	H'FCFF0008	16
	Timer general register B_2	TGRB_2	16	H'FCFF000A	16
	Timer control register_3	TCR_3	8	H'FCFF0200	8
	Timer mode register_3	TMDR_3	8	H'FCFF0202	8
	Timer I/O control register H_3	TIORH_3	8	H'FCFF0204	8
	Timer I/O control register L_3	TIORL_3	8	H'FCFF0205	8
	Timer interrupt enable register_3	TIER_3	8	H'FCFF0208	8
	Timer status register_3	TSR_3	8	H'FCFF022C	8
	Timer counter_3	TCNT_3	16	H'FCFF0210	16
	Timer general register A_3	TGRA_3	16	H'FCFF0218	16
	Timer general register B_3	TGRB_3	16	H'FCFF021A	16
	Timer general register C_3	TGRC_3	16	H'FCFF0224	16
	Timer general register D_3	TGRD_3	16	H'FCFF0226	16
	Timer buffer operation transfer mode register_3	TBTM_3	8	H'FCFF0238	8
	Timer control register_4	TCR_4	8	H'FCFF0201	8
	Timer mode register_4	TMDR_4	8	H'FCFF0203	8
	Timer I/O control register H_4	TIORH_4	8	H'FCFF0206	8
	Timer I/O control register L_4	TIORL_4	8	H'FCFF0207	8
	Timer interrupt enable register_4	TIER_4	8	H'FCFF0209	8
	Timer status register_4	TSR_4	8	H'FCFF022D	8
	Timer counter_4	TCNT_4	16	H'FCFF0212	16
	Timer general register A_4	TGRA_4	16	H'FCFF021C	16
	Timer general register B_4	TGRB_4	16	H'FCFF021E	16
	Timer general register C_4	TGRC_4	16	H'FCFF0228	16
	Timer general register D_4	TGRD_4	16	H'FCFF022A	16
	Timer buffer operation transfer mode register_4	TBTM_4	8	H'FCFF0239	8
	Timer A/D converter start request control register	TADCR	16	H'FCFF0240	16
	Timer A/D converter start request cycle set register A_4	TADCORA_4	16	H'FCFF0244	16
	Timer A/D converter start request cycle set register B_4	TADCORB_4	16	H'FCFF0246	16
	Timer A/D converter start request cycle set buffer register A_4	TADCOBRA_4	16	H'FCFF0248	16
	Timer A/D converter start request cycle set buffer register B_4	TADCOBRB_4	16	H'FCFF024A	16
	Timer start register	TSTR	8	H'FCFF0280	8
	Timer synchronous register	TSYR	8	H'FCFF0281	8
	Timer read/write enable register	TRWER	8	H'FCFF0284	8
	Timer output master enable register	TOER	8	H'FCFF020A	8
	Timer output control register 1	TOCR1	8	H'FCFF020E	8
	Timer output control register 2	TOCR2	8	H'FCFF020F	8
	Timer gate control register	TGCR	8	H'FCFF020D	8
	Timer cycle data register	TCDR	16	H'FCFF0214	16
	Timer dead time data register	TDDR	16	H'FCFF0216	16
	Timer subcounter	TCNTS	16	H'FCFF0220	16
	Timer cycle buffer register	TCBR	16	H'FCFF0222	16
	Timer interrupt skipping set register	TITCR	8	H'FCFF0230	8
	Timer interrupt skipping counter	TITCNT	8	H'FCFF0231	8
	Timer buffer transfer set register	TBTER	8	H'FCFF0232	8
	Timer dead time enable register	TDER	8	H'FCFF0234	8
	Timer waveform control register	TWCR	8	H'FCFF0260	8
	Timer output level buffer register	TOLBR	8	H'FCFF0236	8
OS timer	OSTM0 compare register	OSTM0CMP	32	H'FCFEC000	32
	OSTM0 counter register	OSTM0CNT	32	H'FCFEC004	32
	OSTM0 count enable status register	OSTM0TE	8	H'FCFEC010	8

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
OS timer	OSTM0 count start trigger register	OSTM0TS	8	H'FCFEC014	8
	OSTM0 count stop trigger register	OSTM0TT	8	H'FCFEC018	8
	OSTM0 control register	OSTM0CTL	8	H'FCFEC020	8
	OSTM1 compare register	OSTM1CMP	32	H'FCFEC400	32
	OSTM1 counter register	OSTM1CNT	32	H'FCFEC404	32
	OSTM1 count enable status register	OSTM1TE	8	H'FCFEC410	8
	OSTM1 count start trigger register	OSTM1TS	8	H'FCFEC414	8
	OSTM1 count stop trigger register	OSTM1TT	8	H'FCFEC418	8
	OSTM1 control register	OSTM1CTL	8	H'FCFEC420	8
Watchdog timer	Watchdog timer counter	WTCNT	8	H'FCFE0002	16
	Watchdog timer control/status register	WTCSR	8	H'FCFE0000	16
	Watchdog reset control/status register	WRCSR	8	H'FCFE0004	16
Realtime clock	64-Hz counter	R64CNT	8	H'FCFF1000	8
	Second counter	RSECCNT	8	H'FCFF1002	8
	Minute counter	RMINCNT	8	H'FCFF1004	8
	Hour counter	RHRCNT	8	H'FCFF1006	8
	Day of week counter	RWKCNT	8	H'FCFF1008	8
	Day counter	RDAYCNT	8	H'FCFF100A	8
	Month counter	RMONCNT	8	H'FCFF100C	8
	Year counter	RYRCNT	16	H'FCFF100E	16
	Second alarm register	RSECAR	8	H'FCFF1010	8
	Minute alarm register	RMINAR	8	H'FCFF1012	8
	Hour alarm register	RHRAR	8	H'FCFF1014	8
	Day of week alarm register	RWKAR	8	H'FCFF1016	8
	Day alarm register	RDAYAR	8	H'FCFF1018	8
	Month alarm register	RMONAR	8	H'FCFF101A	8
	Year alarm register	RYRAR	16	H'FCFF1020	16
	Control register 1	RCR1	8	H'FCFF101C	8
	Control register 2	RCR2	8	H'FCFF101E	8
	Control register 3	RCR3	8	H'FCFF1024	8
	Control register 5	RCR5	8	H'FCFF1026	8
	Frequency register H	RFRH	16	H'FCFF102A	16
	Frequency register L	RFRL	16	H'FCFF102C	16
Serial communication interface with FIFO	Serial mode register_0	SCSMR_0	16	H'E8007000	16
	Bit rate register_0	SCBRR_0	8	H'E8007004	8
	Serial control register_0	SCSCR_0	16	H'E8007008	16
	Transmit FIFO data register_0	SCFTDR_0	8	H'E800700C	8
	Serial status register_0	SCFSR_0	16	H'E8007010	16
	Receive FIFO data register_0	SCFRDR_0	8	H'E8007014	8
	FIFO control register_0	SCFCR_0	16	H'E8007018	16
	FIFO data count set register_0	SCFDR_0	16	H'E800701C	16
	Serial port register_0	SCSPTR_0	16	H'E8007020	16
	Line status register_0	SCLSR_0	16	H'E8007024	16
	Serial extension mode register_0	SCEMR_0	16	H'E8007028	16
	Serial mode register_1	SCSMR_1	16	H'E8007800	16
	Bit rate register_1	SCBRR_1	8	H'E8007804	8
	Serial control register_1	SCSCR_1	16	H'E8007808	16
	Transmit FIFO data register_1	SCFTDR_1	8	H'E800780C	8
	Serial status register_1	SCFSR_1	16	H'E8007810	16
	Receive FIFO data register_1	SCFRDR_1	8	H'E8007814	8
	FIFO control register_1	SCFCR_1	16	H'E8007818	16
	FIFO data count set register_1	SCFDR_1	16	H'E800781C	16

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Serial communication interface with FIFO	Serial port register_1	SCSPTR_1	16	H'E8007820	16
	Line status register_1	SCLSR_1	16	H'E8007824	16
	Serial extension mode register_1	SCEMR_1	16	H'E8007828	16
	Serial mode register_2	SCSMR_2	16	H'E8008000	16
	Bit rate register_2	SCBRR_2	8	H'E8008004	8
	Serial control register_2	SCSCR_2	16	H'E8008008	16
	Transmit FIFO data register_2	SCFTDR_2	8	H'E800800C	8
	Serial status register_2	SCFSR_2	16	H'E8008010	16
	Receive FIFO data register_2	SCFRDR_2	8	H'E8008014	8
	FIFO control register_2	SCFCR_2	16	H'E8008018	16
	FIFO data count set register_2	SCFDR_2	16	H'E800801C	16
	Serial port register_2	SCSPTR_2	16	H'E8008020	16
	Line status register_2	SCLSR_2	16	H'E8008024	16
	Serial extension mode register_2	SCEMR_2	16	H'E8008028	16
	Serial mode register_3	SCSMR_3	16	H'E8008800	16
	Bit rate register_3	SCBRR_3	8	H'E8008804	8
	Serial control register_3	SCSCR_3	16	H'E8008808	16
	Transmit FIFO data register_3	SCFTDR_3	8	H'E800880C	8
	Serial status register_3	SCFSR_3	16	H'E8008810	16
	Receive FIFO data register_3	SCFRDR_3	8	H'E8008814	8
	FIFO control register_3	SCFCR_3	16	H'E8008818	16
	FIFO data count set register_3	SCFDR_3	16	H'E800881C	16
	Serial port register_3	SCSPTR_3	16	H'E8008820	16
	Line status register_3	SCLSR_3	16	H'E8008824	16
	Serial extension mode register_3	SCEMR_3	16	H'E8008828	16
	Serial mode register_4	SCSMR_4	16	H'E8009000	16
	Bit rate register_4	SCBRR_4	8	H'E8009004	8
	Serial control register_4	SCSCR_4	16	H'E8009008	16
	Transmit FIFO data register_4	SCFTDR_4	8	H'E800900C	8
	Serial status register_4	SCFSR_4	16	H'E8009010	16
	Receive FIFO data register_4	SCFRDR_4	8	H'E8009014	8
	FIFO control register_4	SCFCR_4	16	H'E8009018	16
	FIFO data count set register_4	SCFDR_4	16	H'E800901C	16
	Serial port register_4	SCSPTR_4	16	H'E8009020	16
	Line status register_4	SCLSR_4	16	H'E8009024	16
	Serial extension mode register_4	SCEMR_4	16	H'E8009028	16
Serial communications interface	Serial mode register 0	SMR0	8	H'E800B000	8
	Bit rate register 0	BRR0	8	H'E800B001	8
	Serial control register 0	SCR0	8	H'E800B002	8
	Transmit data register 0	TDR0	8	H'E800B003	8
	Serial status register 0	SSR0	8	H'E800B004	8
	Receive data register 0	RDR0	8	H'E800B005	8
	Smart card mode register 0	SCMR0	8	H'E800B006	8
	Serial extended mode register 0	SEMR0	8	H'E800B007	8
	Noise filter setting register 0	SNFR0	8	H'E800B008	8
	Extended function control register 0	SECR0	8	H'E800B00D	8
	Serial mode register 1	SMR1	8	H'E800B800	8
	Bit rate register 1	BRR1	8	H'E800B801	8
	Serial control register 1	SCR1	8	H'E800B802	8
	Transmit data register 1	TDR1	8	H'E800B803	8
	Serial status register 1	SSR1	8	H'E800B804	8
	Receive data register 1	RDR1	8	H'E800B805	8

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Serial communications interface	Smart card mode register 1	SCMR1	8	H'E800B806	8
	Serial extended mode register 1	SEMR1	8	H'E800B807	8
	Noise filter setting register 1	SNFR1	8	H'E800B808	8
	Extended function control register 1	SECR1	8	H'E800B80D	8
	IrDA control register	IRCR	8	H'E8014000	8
Renesas serial peripheral interface	Control register_0	SPCR_0	8	H'E800C800	8
	Slave select polarity register_0	SSLP_0	8	H'E800C801	8
	Pin control register_0	SPPCR_0	8	H'E800C802	8
	Status register_0	SPSR_0	8	H'E800C803	8
	Data register_0	SPDR_0	32	H'E800C804	8, 16, 32
	Sequence control register_0	SPSCR_0	8	H'E800C808	8
	Sequence status register_0	SPSSR_0	8	H'E800C809	8
	Bit rate register_0	SPBR_0	8	H'E800C80A	8
	Data control register_0	SPDCR_0	8	H'E800C80B	8
	Clock delay register_0	SPCKD_0	8	H'E800C80C	8
	Slave select negation delay register_0	SSLND_0	8	H'E800C80D	8
	Next-access delay register_0	SPND_0	8	H'E800C80E	8
	Command register 0_0	SPCMD0_0	16	H'E800C810	16
	Command register 1_0	SPCMD1_0	16	H'E800C812	16
	Command register 2_0	SPCMD2_0	16	H'E800C814	16
	Command register 3_0	SPCMD3_0	16	H'E800C816	16
	Buffer control register_0	SPBFCR_0	8	H'E800C820	8
	Buffer data count setting register_0	SPBFDR_0	16	H'E800C822	16
	Control register_1	SPCR_1	8	H'E800D000	8
	Slave select polarity register_1	SSLP_1	8	H'E800D001	8
	Pin control register_1	SPPCR_1	8	H'E800D002	8
	Status register_1	SPSR_1	8	H'E800D003	8
	Data register_1	SPDR_1	32	H'E800D004	8, 16, 32
	Sequence control register_1	SPSCR_1	8	H'E800D008	8
	Sequence status register_1	SPSSR_1	8	H'E800D009	8
	Bit rate register_1	SPBR_1	8	H'E800D00A	8
	Data control register_1	SPDCR_1	8	H'E800D00B	8
	Clock delay register_1	SPCKD_1	8	H'E800D00C	8
	Slave select negation delay register_1	SSLND_1	8	H'E800D00D	8
	Next-access delay register_1	SPND_1	8	H'E800D00E	8
	Command register 0_1	SPCMD0_1	16	H'E800D010	16
	Command register 1_1	SPCMD1_1	16	H'E800D012	16
	Command register 2_1	SPCMD2_1	16	H'E800D014	16
	Command register 3_1	SPCMD3_1	16	H'E800D016	16
	Buffer control register_1	SPBFCR_1	8	H'E800D020	8
	Buffer data count setting register_1	SPBFDR_1	16	H'E800D022	16
	Control register_2	SPCR_2	8	H'E800D800	8
	Slave select polarity register_2	SSLP_2	8	H'E800D801	8
	Pin control register_2	SPPCR_2	8	H'E800D802	8
	Status register_2	SPSR_2	8	H'E800D803	8
	Data register_2	SPDR_2	32	H'E800D804	8, 16, 32
	Sequence control register_2	SPSCR_2	8	H'E800D808	8
	Sequence status register_2	SPSSR_2	8	H'E800D809	8
	Bit rate register_2	SPBR_2	8	H'E800D80A	8
	Data control register_2	SPDCR_2	8	H'E800D80B	8
	Clock delay register_2	SPCKD_2	8	H'E800D80C	8
	Slave select negation delay register_2	SSLND_2	8	H'E800D80D	8

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Renesas serial peripheral interface	Next-access delay register_2	SPND_2	8	H'E800D80E	8
	Command register 0_2	SPCMD0_2	16	H'E800D810	16
	Command register 1_2	SPCMD1_2	16	H'E800D812	16
	Command register 2_2	SPCMD2_2	16	H'E800D814	16
	Command register 3_2	SPCMD3_2	16	H'E800D816	16
	Buffer control register_2	SPBFCR_2	8	H'E800D820	8
	Buffer data count setting register_2	SPBFDR_2	16	H'E800D822	16
SPI multi I/O bus controller	Common control register_0	CMNCR_0	32	H'3FEFA000	32
	SSL delay register_0	SSLDR_0	32	H'3FEFA004	32
	Bit rate register_0	SPBCR_0	32	H'3FEFA008	32
	Data read control register_0	DRCR_0	32	H'3FEFA00C	32
	Data read command setting register_0	DRCMR_0	32	H'3FEFA010	32
	Data read extended address setting register_0	DREAR_0	32	H'3FEFA014	32
	Data read option setting register_0	DROPR_0	32	H'3FEFA018	32
	Data read enable setting register_0	DRENR_0	32	H'3FEFA01C	32
	SPI mode control register_0	SMCR_0	32	H'3FEFA020	32
	SPI mode command setting register_0	SMCMR_0	32	H'3FEFA024	32
	SPI mode address setting register_0	SMADR_0	32	H'3FEFA028	32
	SPI mode option setting register_0	SMOPR_0	32	H'3FEFA02C	32
	SPI mode enable setting register_0	SMENR_0	32	H'3FEFA030	32
	SPI mode read data register 0_0	SMRDR0_0	32	H'3FEFA038	8, 16, 32
	SPI mode read data register 1_0	SMRDR1_0	32	H'3FEFA03C	8, 16, 32
	SPI mode write data register 0_0	SMWDR0_0	32	H'3FEFA040	8, 16, 32
	SPI mode write data register 1_0	SMWDR1_0	32	H'3FEFA044	8, 16, 32
	Common status register_0	CMNSR_0	32	H'3FEFA048	32
	SPI AC input characteristics adjustment register_0*2	CKDLY_0	32	H'3FEFA050	32
	Data read dummy cycle setting register_0	DRDMCR_0	32	H'3FEFA058	32
	Data read DDR enable register_0*2	DRDRENR_0	32	H'3FEFA05C	32
	SPI mode dummy cycle setting register_0	SMDMCR_0	32	H'3FEFA060	32
	SPI mode DDR enable register_0*2	SMDRENR_0	32	H'3FEFA064	32
	SPI AC output characteristics adjustment register_0*2	SPODLY_0	32	H'3FEFA068	32
I2C bus interface	I2C bus control register 1_0	RIIC0CR1	32	H'FCFEE000	8, 16, 32
	I2C bus control register 2_0	RIIC0CR2	32	H'FCFEE004	8, 16, 32
	I2C bus mode register 1_0	RIIC0MR1	32	H'FCFEE008	8, 16, 32
	I2C bus mode register 2_0	RIIC0MR2	32	H'FCFEE00C	8, 16, 32
	I2C bus mode register 3_0	RIIC0MR3	32	H'FCFEE010	8, 16, 32
	I2C bus function enable register_0	RIIC0FER	32	H'FCFEE014	8, 16, 32
	I2C bus status enable register_0	RIIC0SER	32	H'FCFEE018	8, 16, 32
	I2C bus interrupt enable register_0	RIIC0IER	32	H'FCFEE01C	8, 16, 32
	I2C bus status register 1_0	RIIC0SR1	32	H'FCFEE020	8, 16, 32
	I2C bus status register 2_0	RIIC0SR2	32	H'FCFEE024	8, 16, 32
	I2C slave address register 0_0	RIIC0SAR0	32	H'FCFEE028	8, 16, 32
	I2C slave address register 1_0	RIIC0SAR1	32	H'FCFEE02C	8, 16, 32
	I2C slave address register 2_0	RIIC0SAR2	32	H'FCFEE030	8, 16, 32
	I2C bus bit rate low-level register_0	RIIC0BRL	32	H'FCFEE034	8, 16, 32
	I2C bus bit rate high-level register_0	RIIC0BRH	32	H'FCFEE038	8, 16, 32
	I2C bus transmit data register_0	RIIC0DRT	32	H'FCFEE03C	8, 16, 32
	I2C bus transmit data register_0	RIIC0DRR	32	H'FCFEE040	8, 16, 32
	I2C bus control register 1_1	RIIC1CR1	32	H'FCFEE400	8, 16, 32
	I2C bus control register 2_1	RIIC1CR2	32	H'FCFEE404	8, 16, 32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
I2C bus interface	I2C bus mode register 1_1	RIIC1MR1	32	H'FCFEE408	8, 16, 32
	I2C bus mode register 2_1	RIIC1MR2	32	H'FCFEE40C	8, 16, 32
	I2C bus mode register 3_1	RIIC1MR3	32	H'FCFEE410	8, 16, 32
	I2C bus function enable register_1	RIIC1FER	32	H'FCFEE414	8, 16, 32
	I2C bus status enable register_1	RIIC1SER	32	H'FCFEE418	8, 16, 32
	I2C bus interrupt enable register_1	RIIC1IER	32	H'FCFEE41C	8, 16, 32
	I2C bus status register 1_1	RIIC1SR1	32	H'FCFEE420	8, 16, 32
	I2C bus status register 2_1	RIIC1SR2	32	H'FCFEE424	8, 16, 32
	I2C slave address register 0_1	RIIC1SAR0	32	H'FCFEE428	8, 16, 32
	I2C slave address register 1_1	RIIC1SAR1	32	H'FCFEE42C	8, 16, 32
	I2C slave address register 2_1	RIIC1SAR2	32	H'FCFEE430	8, 16, 32
	I2C bus bit rate low-level register_1	RIIC1BRL	32	H'FCFEE434	8, 16, 32
	I2C bus bit rate high-level register_1	RIIC1BRH	32	H'FCFEE438	8, 16, 32
	I2C bus transmit data register_1	RIIC1DRT	32	H'FCFEE43C	8, 16, 32
	I2C bus receive data register_1	RIIC1DRR	32	H'FCFEE440	8, 16, 32
	I2C bus control register 1_2	RIIC2CR1	32	H'FCFEE800	8, 16, 32
	I2C bus control register 2_2	RIIC2CR2	32	H'FCFEE804	8, 16, 32
	I2C bus mode register 1_2	RIIC2MR1	32	H'FCFEE808	8, 16, 32
	I2C bus mode register 2_2	RIIC2MR2	32	H'FCFEE80C	8, 16, 32
	I2C bus mode register 3_2	RIIC2MR3	32	H'FCFEE810	8, 16, 32
	I2C bus function enable register_2	RIIC2FER	32	H'FCFEE814	8, 16, 32
	I2C bus status enable register_2	RIIC2SER	32	H'FCFEE818	8, 16, 32
	I2C bus interrupt enable register_2	RIIC2IER	32	H'FCFEE81C	8, 16, 32
	I2C bus status register 1_2	RIIC2SR1	32	H'FCFEE820	8, 16, 32
	I2C bus status register 2_2	RIIC2SR2	32	H'FCFEE824	8, 16, 32
	I2C slave address register 0_2	RIIC2SAR0	32	H'FCFEE828	8, 16, 32
	I2C slave address register 1_2	RIIC2SAR1	32	H'FCFEE82C	8, 16, 32
	I2C slave address register 2_2	RIIC2SAR2	32	H'FCFEE830	8, 16, 32
	I2C bus bit rate low-level register_2	RIIC2BRL	32	H'FCFEE834	8, 16, 32
	I2C bus bit rate high-level register_2	RIIC2BRH	32	H'FCFEE838	8, 16, 32
	I2C bus transmit data register_2	RIIC2DRT	32	H'FCFEE83C	8, 16, 32
	I2C bus receive data register_2	RIIC2DRR	32	H'FCFEE840	8, 16, 32
	I2C bus control register 1_3	RIIC3CR1	32	H'FCFEEC00	8, 16, 32
	I2C bus control register 2_3	RIIC3CR2	32	H'FCFEEC04	8, 16, 32
	I2C bus mode register 1_3	RIIC3MR1	32	H'FCFEEC08	8, 16, 32
	I2C bus mode register 2_3	RIIC3MR2	32	H'FCFEEC0C	8, 16, 32
	I2C bus mode register 3_3	RIIC3MR3	32	H'FCFEEC10	8, 16, 32
	I2C bus function enable register_3	RIIC3FER	32	H'FCFEEC14	8, 16, 32
	I2C bus status enable register_3	RIIC3SER	32	H'FCFEEC18	8, 16, 32
	I2C bus interrupt enable register_3	RIIC3IER	32	H'FCFEEC1C	8, 16, 32
	I2C bus status register 1_3	RIIC3SR1	32	H'FCFEEC20	8, 16, 32
	I2C bus status register 2_3	RIIC3SR2	32	H'FCFEEC24	8, 16, 32
	I2C slave address register 0_3	RIIC3SAR0	32	H'FCFEEC28	8, 16, 32
	I2C slave address register 1_3	RIIC3SAR1	32	H'FCFEEC2C	8, 16, 32
	I2C slave address register 2_3	RIIC3SAR2	32	H'FCFEEC30	8, 16, 32
	I2C bus bit rate low-level register_3	RIIC3BRL	32	H'FCFEEC34	8, 16, 32
	I2C bus bit rate high-level register_3	RIIC3BRH	32	H'FCFEEC38	8, 16, 32
	I2C bus transmit data register_3	RIIC3DRT	32	H'FCFEEC3C	8, 16, 32
	I2C bus receive data register_3	RIIC3DRR	32	H'FCFEEC40	8, 16, 32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Serial sound interface	Control register_0	SSICR_0	32	H'E820B000	32
	Status register_0	SSISR_0	32	H'E820B004	32
	FIFO control register_0	SSIFCR_0	32	H'E820B010	32
	FIFO status register_0	SSIFSR_0	32	H'E820B014	32
	Transmit FIFO data register_0	SSIFTDR_0	32	H'E820B018	32
	Receive FIFO data register_0	SSIFRDR_0	32	H'E820B01C	32
	TDM mode register_0	SSITDMR_0	32	H'E820B020	32
	FC control register_0	SSIFCCR_0	32	H'E820B024	32
	FC mode register_0	SSIFCMR_0	32	H'E820B028	32
	FC status register_0	SSIFCSR_0	32	H'E820B02C	32
	Control register_1	SSICR_1	32	H'E820B800	32
	Status register_1	SSISR_1	32	H'E820B804	32
	FIFO control register_1	SSIFCR_1	32	H'E820B810	32
	FIFO status register_1	SSIFSR_1	32	H'E820B814	32
	Transmit FIFO data register_1	SSIFTDR_1	32	H'E820B818	32
	Receive FIFO data register_1	SSIFRDR_1	32	H'E820B81C	32
	TDM mode register_1	SSITDMR_1	32	H'E820B820	32
	FC control register_1	SSIFCCR_1	32	H'E820B824	32
	FC mode register_1	SSIFCMR_1	32	H'E820B828	32
	FC status register_1	SSIFCSR_1	32	H'E820B82C	32
	Control register_2	SSICR_2	32	H'E820C000	32
	Status register_2	SSISR_2	32	H'E820C004	32
	FIFO control register_2	SSIFCR_2	32	H'E820C010	32
	FIFO status register_2	SSIFSR_2	32	H'E820C014	32
	Transmit FIFO data register_2	SSIFTDR_2	32	H'E820C018	32
	Receive FIFO data register_2	SSIFRDR_2	32	H'E820C01C	32
	TDM mode register_2	SSITDMR_2	32	H'E820C020	32
	FC control register_2	SSIFCCR_2	32	H'E820C024	32
	FC mode register_2	SSIFCMR_2	32	H'E820C028	32
	FC status register_2	SSIFCSR_2	32	H'E820C02C	32
	Control register_3	SSICR_3	32	H'E820C800	32
	Status register_3	SSISR_3	32	H'E820C804	32
	FIFO control register_3	SSIFCR_3	32	H'E820C810	32
	FIFO status register_3	SSIFSR_3	32	H'E820C814	32
	Transmit FIFO data register_3	SSIFTDR_3	32	H'E820C818	32
	Receive FIFO data register_3	SSIFRDR_3	32	H'E820C81C	32
	TDM mode register_3	SSITDMR_3	32	H'E820C820	32
	FC control register_3	SSIFCCR_3	32	H'E820C824	32
	FC mode register_3	SSIFCMR_3	32	H'E820C828	32
	FC status register_3	SSIFCSR_3	32	H'E820C82C	32
Media local bus*1	Device Control Cfg Register	DCCR	32	H'E8034000	32
	System Status Cfg Register	SSCR	32	H'E8034004	32
	System Data Cfg Register	SDCR	32	H'E8034008	32
	System Mask Cfg Register	SMCR	32	H'E803400C	32
	Version Control Cfg Register	VCCR	32	H'E803401C	32
	Synchronous Base Address Cfg Register	SBCR	32	H'E8034020	32
	Asynchronous Base Address Cfg Register	ABCR	32	H'E8034024	32
	Control Base Address Cfg Register	CBCR	32	H'E8034028	32
	Isochronous Base Address Cfg Register	IBCR	32	H'E803402C	32
	Channel Interrupt Cfg Register	CICR	32	H'E8034030	32
	Channel 0 Entry Cfg Register	CECR0	32	H'E8034040	32
	Channel 0 Status Cfg Register	CSCR0	32	H'E8034044	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Media local bus*1	Channel 0 Current Buffer Cfg Register	CCBCR0	32	H'E8034048	32
	Channel 0 Next Buffer Cfg Register	CNBCR0	32	H'E803404C	32
	Channel 1 Entry Cfg Register	CECR1	32	H'E8034050	32
	Channel 1 Status Cfg Register	CSCR1	32	H'E8034054	32
	Channel 1 Current Buffer Cfg Register	CCBCR1	32	H'E8034058	32
	Channel 1 Next Buffer Cfg Register	CNBCR1	32	H'E803405C	32
	Channel 2 Entry Cfg Register	CECR2	32	H'E8034060	32
	Channel 2 Status Cfg Register	CSCR2	32	H'E8034064	32
	Channel 2 Current Buffer Cfg Register	CCBCR2	32	H'E8034068	32
	Channel 2 Next Buffer Cfg Register	CNBCR2	32	H'E803406C	32
	Channel 3 Entry Cfg Register	CECR3	32	H'E8034070	32
	Channel 3 Status Cfg Register	CSCR3	32	H'E8034074	32
	Channel 3 Current Buffer Cfg Register	CCBCR3	32	H'E8034078	32
	Channel 3 Next Buffer Cfg Register	CNBCR3	32	H'E803407C	32
	Channel 4 Entry Cfg Register	CECR4	32	H'E8034080	32
	Channel 4 Status Cfg Register	CSCR4	32	H'E8034084	32
	Channel 4 Current Buffer Cfg Register	CCBCR4	32	H'E8034088	32
	Channel 4 Next Buffer Cfg Register	CNBCR4	32	H'E803408C	32
	Channel 5 Entry Cfg Register	CECR5	32	H'E8034090	32
	Channel 5 Status Cfg Register	CSCR5	32	H'E8034094	32
	Channel 5 Current Buffer Cfg Register	CCBCR5	32	H'E8034098	32
	Channel 5 Next Buffer Cfg Register	CNBCR5	32	H'E803409C	32
	Channel 6 Entry Cfg Register	CECR6	32	H'E80340A0	32
	Channel 6 Status Cfg Register	CSCR6	32	H'E80340A4	32
	Channel 6 Current Buffer Cfg Register	CCBCR6	32	H'E80340A8	32
	Channel 6 Next Buffer Cfg Register	CNBCR6	32	H'E80340AC	32
	Channel 7 Entry Cfg Register	CECR7	32	H'E80340B0	32
	Channel 7 Status Cfg Register	CSCR7	32	H'E80340B4	32
	Channel 7 Current Buffer Cfg Register	CCBCR7	32	H'E80340B8	32
	Channel 7 Next Buffer Cfg Register	CNBCR7	32	H'E80340BC	32
	Channel 8 Entry Cfg Register	CECR8	32	H'E80340C0	32
	Channel 8 Status Cfg Register	CSCR8	32	H'E80340C4	32
	Channel 8 Current Buffer Cfg Register	CCBCR8	32	H'E80340C8	32
	Channel 8 Next Buffer Cfg Register	CNBCR8	32	H'E80340CC	32
	Channel 9 Entry Cfg Register	CECR9	32	H'E80340D0	32
	Channel 9 Status Cfg Register	CSCR9	32	H'E80340D4	32
	Channel 9 Current Buffer Cfg Register	CCBCR9	32	H'E80340D8	32
	Channel 9 Next Buffer Cfg Register	CNBCR9	32	H'E80340DC	32
	Channel 10 Entry Cfg Register	CECR10	32	H'E80340E0	32
	Channel 10 Status Cfg Register	CSCR10	32	H'E80340E4	32
	Channel 10 Current Buffer Cfg Register	CCBCR10	32	H'E80340E8	32
	Channel 10 Next Buffer Cfg Register	CNBCR10	32	H'E80340EC	32
	Channel 11 Entry Cfg Register	CECR11	32	H'E80340F0	32
	Channel 11 Status Cfg Register	CSCR11	32	H'E80340F4	32
	Channel 11 Current Buffer Cfg Register	CCBCR11	32	H'E80340F8	32
	Channel 11 Next Buffer Cfg Register	CNBCR11	32	H'E80340FC	32
	Channel 12 Entry Cfg Register	CECR12	32	H'E8034100	32
	Channel 12 Status Cfg Register	CSCR12	32	H'E8034104	32
	Channel 12 Current Buffer Cfg Register	CCBCR12	32	H'E8034108	32
	Channel 12 Next Buffer Cfg Register	CNBCR12	32	H'E803410C	32
	Channel 13 Entry Cfg Register	CECR13	32	H'E8034110	32
	Channel 13 Status Cfg Register	CSCR13	32	H'E8034114	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Media local bus*1	Channel 13 Current Buffer Cfg Register	CCBCR13	32	H'E8034118	32
	Channel 13 Next Buffer Cfg Register	CNBCR13	32	H'E803411C	32
	Channel 14 Entry Cfg Register	CECR14	32	H'E8034120	32
	Channel 14 Status Cfg Register	CSCR14	32	H'E8034124	32
	Channel 14 Current Buffer Cfg Register	CCBCR14	32	H'E8034128	32
	Channel 14 Next Buffer Cfg Register	CNBCR14	32	H'E803412C	32
	Channel 15 Entry Cfg Register	CECR15	32	H'E8034130	32
	Channel 15 Status Cfg Register	CSCR15	32	H'E8034134	32
	Channel 15 Current Buffer Cfg Register	CCBCR15	32	H'E8034138	32
	Channel 15 Next Buffer Cfg Register	CNBCR15	32	H'E803413C	32
	Channel 16 Entry Cfg Register	CECR16	32	H'E8034140	32
	Channel 16 Status Cfg Register	CSCR16	32	H'E8034144	32
	Channel 16 Current Buffer Cfg Register	CCBCR16	32	H'E8034148	32
	Channel 16 Next Buffer Cfg Register	CNBCR16	32	H'E803414C	32
	Channel 17 Entry Cfg Register	CECR17	32	H'E8034150	32
	Channel 17 Status Cfg Register	CSCR17	32	H'E8034154	32
	Channel 17 Current Buffer Cfg Register	CCBCR17	32	H'E8034158	32
	Channel 17 Next Buffer Cfg Register	CNBCR17	32	H'E803415C	32
	Channel 18 Entry Cfg Register	CECR18	32	H'E8034160	32
	Channel 18 Status Cfg Register	CSCR18	32	H'E8034164	32
	Channel 18 Current Buffer Cfg Register	CCBCR18	32	H'E8034168	32
	Channel 18 Next Buffer Cfg Register	CNBCR18	32	H'E803416C	32
	Channel 19 Entry Cfg Register	CECR19	32	H'E8034170	32
	Channel 19 Status Cfg Register	CSCR19	32	H'E8034174	32
	Channel 19 Current Buffer Cfg Register	CCBCR19	32	H'E8034178	32
	Channel 19 Next Buffer Cfg Register	CNBCR19	32	H'E803417C	32
	Channel 20 Entry Cfg Register	CECR20	32	H'E8034180	32
	Channel 20 Status Cfg Register	CSCR20	32	H'E8034184	32
	Channel 20 Current Buffer Cfg Register	CCBCR20	32	H'E8034188	32
	Channel 20 Next Buffer Cfg Register	CNBCR20	32	H'E803418C	32
	Channel 21 Entry Cfg Register	CECR21	32	H'E8034190	32
	Channel 21 Status Cfg Register	CSCR21	32	H'E8034194	32
	Channel 21 Current Buffer Cfg Register	CCBCR21	32	H'E8034198	32
	Channel 21 Next Buffer Cfg Register	CNBCR21	32	H'E803419C	32
	Channel 22 Entry Cfg Register	CECR22	32	H'E80341A0	32
	Channel 22 Status Cfg Register	CSCR22	32	H'E80341A4	32
	Channel 22 Current Buffer Cfg Register	CCBCR22	32	H'E80341A8	32
	Channel 22 Next Buffer Cfg Register	CNBCR22	32	H'E80341AC	32
	Channel 23 Entry Cfg Register	CECR23	32	H'E80341B0	32
	Channel 23 Status Cfg Register	CSCR23	32	H'E80341B4	32
	Channel 23 Current Buffer Cfg Register	CCBCR23	32	H'E80341B8	32
	Channel 23 Next Buffer Cfg Register	CNBCR23	32	H'E80341BC	32
	Channel 24 Entry Cfg Register	CECR24	32	H'E80341C0	32
	Channel 24 Status Cfg Register	CSCR24	32	H'E80341C4	32
	Channel 24 Current Buffer Cfg Register	CCBCR24	32	H'E80341C8	32
	Channel 24 Next Buffer Cfg Register	CNBCR24	32	H'E80341CC	32
	Channel 25 Entry Cfg Register	CECR25	32	H'E80341D0	32
	Channel 25 Status Cfg Register	CSCR25	32	H'E80341D4	32
	Channel 25 Current Buffer Cfg Register	CCBCR25	32	H'E80341D8	32
	Channel 25 Next Buffer Cfg Register	CNBCR25	32	H'E80341DC	32
	Channel 26 Entry Cfg Register	CECR26	32	H'E80341E0	32
	Channel 26 Status Cfg Register	CSCR26	32	H'E80341E4	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Media local bus*1	Channel 26 Current Buffer Cfg Register	CCBCR26	32	H'E80341E8	32
	Channel 26 Next Buffer Cfg Register	CNBCR26	32	H'E80341EC	32
	Channel 27 Entry Cfg Register	CECR27	32	H'E80341F0	32
	Channel 27 Status Cfg Register	CSCR27	32	H'E80341F4	32
	Channel 27 Current Buffer Cfg Register	CCBCR27	32	H'E80341F8	32
	Channel 27 Next Buffer Cfg Register	CNBCR27	32	H'E80341FC	32
	Channel 28 Entry Cfg Register	CECR28	32	H'E8034200	32
	Channel 28 Status Cfg Register	CSCR28	32	H'E8034204	32
	Channel 28 Current Buffer Cfg Register	CCBCR28	32	H'E8034208	32
	Channel 28 Next Buffer Cfg Register	CNBCR28	32	H'E803420C	32
	Channel 29 Entry Cfg Register	CECR29	32	H'E8034210	32
	Channel 29 Status Cfg Register	CSCR29	32	H'E8034214	32
	Channel 29 Current Buffer Cfg Register	CCBCR29	32	H'E8034218	32
	Channel 29 Next Buffer Cfg Register	CNBCR29	32	H'E803421C	32
	Channel 30 Entry Cfg Register	CECR30	32	H'E8034220	32
	Channel 30 Status Cfg Register	CSCR30	32	H'E8034224	32
	Channel 30 Current Buffer Cfg Register	CCBCR30	32	H'E8034228	32
	Channel 30 Next Buffer Cfg Register	CNBCR30	32	H'E803422C	32
	Local Channel 0 Buffer Cfg Register	LCBCR0	32	H'E8034280	32
	Local Channel 1 Buffer Cfg Register	LCBCR1	32	H'E8034284	32
	Local Channel 2 Buffer Cfg Register	LCBCR2	32	H'E8034288	32
	Local Channel 3 Buffer Cfg Register	LCBCR3	32	H'E803428C	32
	Local Channel 4 Buffer Cfg Register	LCBCR4	32	H'E8034290	32
	Local Channel 5 Buffer Cfg Register	LCBCR5	32	H'E8034294	32
	Local Channel 6 Buffer Cfg Register	LCBCR6	32	H'E8034298	32
	Local Channel 7 Buffer Cfg Register	LCBCR7	32	H'E803429C	32
	Local Channel 8 Buffer Cfg Register	LCBCR8	32	H'E80342A0	32
	Local Channel 9 Buffer Cfg Register	LCBCR9	32	H'E80342A4	32
	Local Channel 10 Buffer Cfg Register	LCBCR10	32	H'E80342A8	32
	Local Channel 11 Buffer Cfg Register	LCBCR11	32	H'E80342AC	32
	Local Channel 12 Buffer Cfg Register	LCBCR12	32	H'E80342B0	32
	Local Channel 13 Buffer Cfg Register	LCBCR13	32	H'E80342B4	32
	Local Channel 14 Buffer Cfg Register	LCBCR14	32	H'E80342B8	32
	Local Channel 15 Buffer Cfg Register	LCBCR15	32	H'E80342BC	32
	Local Channel 16 Buffer Cfg Register	LCBCR16	32	H'E80342C0	32
	Local Channel 17 Buffer Cfg Register	LCBCR17	32	H'E80342C4	32
	Local Channel 18 Buffer Cfg Register	LCBCR18	32	H'E80342C8	32
	Local Channel 19 Buffer Cfg Register	LCBCR19	32	H'E80342CC	32
	Local Channel 20 Buffer Cfg Register	LCBCR20	32	H'E80342D0	32
	Local Channel 21 Buffer Cfg Register	LCBCR21	32	H'E80342D4	32
	Local Channel 22 Buffer Cfg Register	LCBCR22	32	H'E80342D8	32
	Local Channel 23 Buffer Cfg Register	LCBCR23	32	H'E80342DC	32
	Local Channel 24 Buffer Cfg Register	LCBCR24	32	H'E80342E0	32
	Local Channel 25 Buffer Cfg Register	LCBCR25	32	H'E80342E4	32
	Local Channel 26 Buffer Cfg Register	LCBCR26	32	H'E80342E8	32
	Local Channel 27 Buffer Cfg Register	LCBCR27	32	H'E80342EC	32
	Local Channel 28 Buffer Cfg Register	LCBCR28	32	H'E80342F0	32
	Local Channel 29 Buffer Cfg Register	LCBCR29	32	H'E80342F4	32
	Local Channel 30 Buffer Cfg Register	LCBCR30	32	H'E80342F8	32
CAN interface	Channel m configuration register (m = 0, 1)	RSCAN0CmCFG (m = 0, 1)	32	H'E803A000 + m * H'0010	8, 16, 32
	Channel m control register (m = 0, 1)	RSCAN0CmCTR (m = 0, 1)	32	H'E803A004 + m * H'0010	8, 16, 32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
CAN interface	Channel m status register (m = 0, 1)	RSCAN0CmSTS (m = 0, 1)	32	H'E803A008 + m * H'0010	8, 16, 32
	Channel m error flag register (m = 0, 1)	RSCAN0CmERFL (m = 0, 1)	32	H'E803A00C + m * H'0010	8, 16, 32
	Global configuration register	RSCAN0GCFG	32	H'E803A084	8, 16, 32
	Global control register	RSCAN0GCTR	32	H'E803A088	8, 16, 32
	Global status register	RSCAN0GSTS	32	H'E803A08C	8, 16, 32
	Global error flag register	RSCAN0GERFL	32	H'E803A090	8, 16, 32
	Global timestamp counter register	RSCAN0GTSC	32	H'E803A094	16, 32
	Receive rule entry control register	RSCAN0GAFLECTR	32	H'E803A098	8, 16, 32
	Receive rule configuration register 0	RSCAN0GAFLCFG0	32	H'E803A09C	8, 16, 32
	Receive buffer number register	RSCAN0RMNB	32	H'E803A0A4	8, 16, 32
	Receive buffer new data register y (y = 0)	RSCAN0RMNDy (y = 0)	32	H'E803A0A8 + y *H'0004	8, 16, 32
	Receive FIFO buffer configuration and control register x (x = 0 to 7)	RSCAN0RFCCx (x = 0 to 7)	32	H'E803A0B8 + x * H'0004	8, 16, 32
	Receive FIFO buffer status register x (x = 0 to 7)	RSCAN0RFSTSx (x = 0 to 7)	32	H'E803A0D8 + x * H'0004	8, 16, 32
	Receive FIFO buffer pointer control register x (x = 0 to 7)	RSCAN0RFPCTRx (x = 0 to 7)	32	H'E803A0F8 + x * H'0004	8, 16, 32
	Transmit/receive FIFO buffer configuration and control register k (k = 0 to 5)	RSCAN0CFCCk (k = 0 to 5)	32	H'E803A118 + k * H'0004	8, 16, 32
	Transmit/receive FIFO buffer status register k (k = 0 to 5)	RSCAN0CFSTSk (k = 0 to 5)	32	H'E803A178 + k * H'0004	8, 16, 32
	Transmit/receive FIFO buffer pointer control register k (k = 0 to 5)	RSCAN0CFPCTRk (k = 0 to 5)	32	H'E803A1D8 + k * H'0004	8, 16, 32
	FIFO empty status register	RSCAN0FESTS	32	H'E803A238	8, 16, 32
	FIFO full status register	RSCAN0FFSTS	32	H'E803A23C	8, 16, 32
	FIFO Msg lost status register	RSCAN0FMSTS	32	H'E803A240	8, 16, 32
	Receive FIFO buffer interrupt flag status register	RSCAN0RFISTS	32	H'E803A244	8, 16, 32
	Transmit/receive FIFO buffer receive interrupt flag status register	RSCAN0CFRISTS	32	H'E803A248	8, 16, 32
	Transmit/receive FIFO buffer transmit interrupt flag status register	RSCAN0CFTISTS	32	H'E803A24C	8, 16, 32
	Transmit buffer control register p (p = 0 to 31)	RSCAN0TMCp (p = 0 to 31)	8	H'E803A250 + p * H'0001	8
	Transmit buffer status register p (p = 0 to 31)	RSCAN0TMSTSp (p = 0 to 31)	8	H'E803A2D0 + p * H'0001	8
	Transmit buffer transmit request status register y (y = 0)	RSCAN0TMTRSTSy (y = 0)	32	H'E803A350 + y * H'0004	8, 16, 32
	Transmit buffer transmit abort request status register y (y = 0)	RSCAN0TMTARSTSy (y = 0)	32	H'E803A360 + y * H'0004	8, 16, 32
	Transmit buffer transmit complete status register y (y = 0)	RSCAN0TMTCSTSy (y = 0)	32	H'E803A370 + y * H'0004	8, 16, 32
	Transmit buffer transmit abort status register y (y = 0)	RSCAN0TMTASTSy (y = 0)	32	H'E803A380 + y * H'0004	8, 16, 32
	Transmit buffer interrupt enable configuration register y (y = 0)	RSCAN0TMIECy (y = 0)	32	H'E803A390 + y * H'0004	8, 16, 32
	Transmit queue configuration and control register m (m = 0, 1)	RSCAN0TXQCCm (m = 0, 1)	32	H'E803A3A0 + m * H'0010	8, 16, 32
	Transmit queue status register m (m = 0, 1)	RSCAN0TXQSTSm (m = 0, 1)	32	H'E803A3C0 + m * H'0004	8, 16, 32
	Transmit queue pointer control register m (m = 0, 1)	RSCAN0TXQPCTRm (m = 0, 1)	32	H'E803A3E0 + m * H'0004	8, 16, 32
	Transmit history configuration and control register m (m = 0, 1)	RSCAN0THLCCm (m = 0, 1)	32	H'E803A400 + m * H'0004	8, 16, 32
	Transmit history status register m (m = 0, 1)	RSCAN0THLSTSm (m = 0, 1)	32	H'E803A420 + m * H'0004	8, 16, 32
	Transmit history pointer control register m (m = 0, 1)	RSCAN0THLPCTRm (m = 0, 1)	32	H'E803A440 + m * H'0004	8, 16, 32
	Global TX interrupt status register 0	RSCAN0GTINTSTS0	32	H'E803A460	8, 16, 32
	Global test configuration register	RSCAN0GTSTCFG	32	H'E803A468	8, 16, 32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
CAN interface	Global test control register	RSCAN0GTSTCTR	32	H'E803A46C	8, 16, 32
	Global lock key register	RSCAN0GLOCKK	32	H'E803A47C	16, 32
	Receive rule ID register j (j = 0 to 15)	RSCAN0GAFLIDj (j = 0 to 15)	32	H'E803A500 + j * H'0010	8, 16, 32
	Receive rule mask register j (j = 0 to 15)	RSCAN0GAFLMj (j = 0 to 15)	32	H'E803A504 + j * H'0010	8, 16, 32
	Receive rule pointer 0 register j (j = 0 to 15)	RSCAN0GAFLP0j (j = 0 to 15)	32	H'E803A508 + j * H'0010	8, 16, 32
	Receive rule pointer 1 register j (j = 0 to 15)	RSCAN0GAFLP1j (j = 0 to 15)	32	H'E803A50C + j * H'0010	8, 16, 32
	Receive buffer ID register q (q = 0 to 31)	RSCAN0RMIDq (q = 0 to 31)	32	H'E803A600 + q * H'0010	8, 16, 32
	Receive buffer pointer register q (q = 0 to 31)	RSCAN0RMPTRq (q = 0 to 31)	32	H'E803A604 + q * H'0010	8, 16, 32
	Receive buffer data field 0 register q (q = 0 to 31)	RSCAN0RMDF0q (q = 0 to 31)	32	H'E803A608 + q * H'0010	8, 16, 32
	Receive buffer data field 1 register q (q = 0 to 31)	RSCAN0RMDF1q (q = 0 to 31)	32	H'E803A60C + q * H'0010	8, 16, 32
	Receive FIFO buffer access ID register x (x = 0 to 7)	RSCAN0RFIDx (x = 0 to 7)	32	H'E803AE00 + x * H'0010	8, 16, 32
	Receive FIFO buffer access pointer register x (x = 0 to 7)	RSCAN0RFPTRx (x = 0 to 7)	32	H'E803AE04 + x * H'0010	8, 16, 32
	Receive FIFO buffer access data field 0 register x (x = 0 to 7)	RSCAN0RFDF0x (x = 0 to 7)	32	H'E803AE08 + x * H'0010	8, 16, 32
	Receive FIFO buffer access data field 1 register x (x = 0 to 7)	RSCAN0RFDF1x (x = 0 to 7)	32	H'E803AE0C + x * H'0010	8, 16, 32
	Transmit/receive FIFO buffer access ID register k (k = 0 to 5)	RSCAN0CFIDk (k = 0 to 5)	32	H'E803AE80 + k * H'0010	8, 16, 32
	Transmit/receive FIFO buffer access pointer register k (k = 0 to 5)	RSCAN0CFPTRk (k = 0 to 5)	32	H'E803AE84 + k * H'0010	8, 16, 32
	Transmit/receive FIFO buffer access data field 0 register k (k = 0 to 5)	RSCAN0CFDF0k (k = 0 to 5)	32	H'E803AE88 + k * H'0010	8, 16, 32
	Transmit/receive FIFO buffer access data field 1 register k (k = 0 to 5)	RSCAN0CFDF1k (k = 0 to 5)	32	H'E803AE8C + k * H'0010	8, 16, 32
	Transmit buffer ID register p (p = 0 to 31)	RSCAN0TMIDp (p = 0 to 31)	32	H'E803B000 + p * H'0010	8, 16, 32
	Transmit buffer pointer register p (p = 0 to 31)	RSCAN0TMPTRp (p = 0 to 31)	32	H'E803B004 + p * H'0010	8, 16, 32
	Transmit buffer data field 0 register p (p = 0 to 31)	RSCAN0TMDF0p (p = 0 to 31)	32	H'E803B008 + p * H'0010	8, 16, 32
	Transmit buffer data field 1 register p (p = 0 to 31)	RSCAN0TMDF1p (p = 0 to 31)	32	H'E803B00C + p * H'0010	8, 16, 32
	Transmit history access register m (m = 0, 1)	RSCAN0THLACCm (m = 0, 1)	32	H'E803B800 + m * H'0004	8, 16, 32
IEBus controller*1	IEBB0 bus control register	IEBB0BCR	8	H'FCFEF000	8
	IEBB0 power save register	IEBB0PSR	8	H'FCFEF004	8
	IEBB0 unit address register	IEBB0UAR	16	H'FCFEF008	16
	IEBB0 slave address register	IEBB0SAR	16	H'FCFEF00C	16
	IEBB0 partner address register	IEBB0PAR	16	H'FCFEF010	16
	IEBB0 reception slave address register	IEBB0RSA	16	H'FCFEF014	16
	IEBB0 control data register	IEBB0CDR	8	H'FCFEF018	8
	IEBB0 transmission control data register	IEBB0TCD	8	H'FCFEF01C	8
	IEBB0 reception control data register	IEBB0RCD	8	H'FCFEF020	8
	IEBB0 message length register	IEBB0DLR	8	H'FCFEF024	8
	IEBB0 transmission message length register	IEBB0TDL	8	H'FCFEF028	8
	IEBB0 reception message length register	IEBB0RDL	8	H'FCFEF02C	8
	IEBB0 clock selection register	IEBB0CKS	8	H'FCFEF030	8
	IEBB0 transfer mode setting register	IEBB0TMS	8	H'FCFEF034	8
	IEBB0 pointer clear register	IEBB0PCR	8	H'FCFEF038	8
	IEBB0 buffer status register	IEBB0BSR	16	H'FCFEF03C	16
	IEBB0 slave status register	IEBB0SSR	8	H'FCFEF040	8
	IEBB0 unit status register	IEBB0USR	8	H'FCFEF044	8

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
IEBus controller*1	IEBB0 interrupt status register	IEBB0ISR	8	H'FCFEF048	8
	IEBB0 error status register	IEBB0ESR	8	H'FCFEF04C	8
	IEBB0 field status register	IEBB0FSR	8	H'FCFEF050	8
	IEBB0 success count register	IEBB0SCR	8	H'FCFEF054	8
	IEBB0 communication count register	IEBB0CCR	8	H'FCFEF058	8
	IEBB0 status clear register 0	IEBB0STC0	8	H'FCFEF05C	8
	IEBB0 status clear register 1	IEBB0STC1	8	H'FCFEF060	8
	IEBB0 data register	IEBB0DR	8	H'FCFEF064	8
Renesas SPDIF interface	Transmitter channel 1 audio register	TLCA	32	H'E8012000	32
	Transmitter channel 2 audio register	TRCA	32	H'E8012004	32
	Transmitter channel 1 status register	TLCS	32	H'E8012008	32
	Transmitter channel 2 status register	TRCS	32	H'E801200C	32
	Transmitter user data register	TUI	32	H'E8012010	32
	Receiver channel 1 audio register	RLCA	32	H'E8012014	32
	Receiver channel 2 audio register	RRCA	32	H'E8012018	32
	Receiver channel 1 status register	RLCS	32	H'E801201C	32
	Receiver channel 2 status register	RRCS	32	H'E8012020	32
	Receiver user data register	RUI	32	H'E8012024	32
	Control register	CTRL	32	H'E8012028	32
	Status register	STAT	32	H'E801202C	32
	Transmitter DMA audio data register	TDAD	32	H'E8012030	32
	Receiver DMA audio data register	RDAD	32	H'E8012034	32
CD-ROM decoder*1	Enable control register	CROMEN	8	H'E8005000	8
	Sync code-based synchronization control register	CROMSY0	8	H'E8005001	8
	Decoding mode control register	CROMCTL0	8	H'E8005002	8
	EDC/ECC check control register	CROMCTL1	8	H'E8005003	8
	Automatic decoding stop control register	CROMCTL3	8	H'E8005005	8
	Decoding option setting control register	CROMCTL4	8	H'E8005006	8
	HEAD20 to HEAD22 representation control register	CROMCTL5	8	H'E8005007	8
	Sync code status register	CROMST0	8	H'E8005008	8
	Post-ECC header error status register	CROMST1	8	H'E8005009	8
	Post-ECC subheader error status register	CROMST3	8	H'E800500B	8
	Header/subheader validity check status register	CROMST4	8	H'E800500C	8
	Mode determination and link sector detection status register	CROMST5	8	H'E800500D	8
	ECC/EDC error status register	CROMST6	8	H'E800500E	8
	Buffer status register	CBUFST0	8	H'E8005014	8
	Decoding stoppage source status register	CBUFST1	8	H'E8005015	8
	Buffer overflow status register	CBUFST2	8	H'E8005016	8
	Pre-ECC correction header: minutes data register	HEAD00	8	H'E8005018	8
	Pre-ECC correction header: seconds data register	HEAD01	8	H'E8005019	8
	Pre-ECC correction header: frames (1/75 second) data register	HEAD02	8	H'E800501A	8
	Pre-ECC correction header: mode data register	HEAD03	8	H'E800501B	8
	Pre-ECC correction subheader: file number (byte 16) data register	SHEAD00	8	H'E800501C	8
	Pre-ECC correction subheader: channel number (byte 17) data register	SHEAD01	8	H'E800501D	8
	Pre-ECC correction subheader: sub-mode (byte 18) data register	SHEAD02	8	H'E800501E	8
	Pre-ECC correction subheader: data type (byte 19) data register	SHEAD03	8	H'E800501F	8
	Pre-ECC correction subheader: file number (byte 20) data register	SHEAD04	8	H'E8005020	8
	Pre-ECC correction subheader: channel number (byte 21) data register	SHEAD05	8	H'E8005021	8

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
CD-ROM decoder*1	Pre-ECC correction subheader: sub-mode (byte 22) data register	SHEAD06	8	H'E8005022	8
	Pre-ECC correction subheader: data type (byte 23) data register	SHEAD07	8	H'E8005023	8
	Post-ECC correction header: minutes data register	HEAD20	8	H'E8005024	8
	Post-ECC correction header: seconds data register	HEAD21	8	H'E8005025	8
	Post-ECC correction header: frames (1/75 second) data register	HEAD22	8	H'E8005026	8
	Post-ECC correction header: mode data register	HEAD23	8	H'E8005027	8
	Post-ECC correction subheader: file number (byte 16) data register	SHEAD20	8	H'E8005028	8
	Post-ECC correction subheader: channel number (byte 17) data register	SHEAD21	8	H'E8005029	8
	Post-ECC correction subheader: sub-mode (byte 18) data register	SHEAD22	8	H'E800502A	8
	Post-ECC correction subheader: data type (byte 19) data register	SHEAD23	8	H'E800502B	8
	Post-ECC correction subheader: file number (byte 20) data register	SHEAD24	8	H'E800502C	8
	Post-ECC correction subheader: channel number (byte 21) data register	SHEAD25	8	H'E800502D	8
	Post-ECC correction subheader: sub-mode (byte 22) data register	SHEAD26	8	H'E800502E	8
	Post-ECC correction subheader: data type (byte 23) data register	SHEAD27	8	H'E800502F	8
	Automatic buffering setting control register	CBUFCTL0	8	H'E8005040	8
	Automatic buffering start sector setting: minutes control register	CBUFCTL1	8	H'E8005041	8
	Automatic buffering start sector setting: seconds control register	CBUFCTL2	8	H'E8005042	8
	Automatic buffering start sector setting: frames control register	CBUFCTL3	8	H'E8005043	8
	ISY interrupt source mask control register	CROMST0M	8	H'E8005045	8
	CD-ROM decoder reset control register	ROMDECRST	8	H'E8005100	8
	CD-ROM decoder reset status register	RSTSTAT	8	H'E8005101	8
	Serial sound interface data control register	SSI	8	H'E8005102	8
	Interrupt flag register	INTHOLD	8	H'E8005108	8
	Interrupt source mask control register	INHINT	8	H'E8005109	8
	CD-ROM decoder stream data input register	STRMDIN0	16	H'E8005200	16 (R/W), 32 (W)
	CD-ROM decoder stream data input register	STRMDIN2	16	H'E8005202	16
	CD-ROM decoder stream data output register	STRMDOUT0	16	H'E8005204	16
LIN interface*1	LIN wake-up baud rate selector register	RLN30LWBR	8	H'FCFE9001	8
	LIN baud rate prescaler 0 register	RLN30LBRP0	8	H'FCFE9002	8
	LIN baud rate prescaler 1 register	RLN30LBRP1	8	H'FCFE9003	8
	LIN self-test control register	RLN30LSTC	8	H'FCFE9004	8
	LIN mode register	RLN30LMD	8	H'FCFE9008	8
	LIN break field configuration register	RLN30LBFC	8	H'FCFE9009	8
	LIN space configuration register	RLN30LSC	8	H'FCFE900A	8
	LIN wake-up configuration register	RLN30LWUP	8	H'FCFE900B	8
	LIN interrupt enable register	RLN30LIE	8	H'FCFE900C	8
	LIN error detection enable register	RLN30LEDE	8	H'FCFE900D	8
	LIN control register	RLN30LCUC	8	H'FCFE900E	8
	LIN transmission control register	RLN30LTRC	8	H'FCFE9010	8
	LIN mode status register	RLN30LMST	8	H'FCFE9011	8
	LIN status register	RLN30LST	8	H'FCFE9012	8
	LIN error status register	RLN30LEST	8	H'FCFE9013	8
	LIN data field configuration register	RLN30LDFC	8	H'FCFE9014	8
	LIN ID buffer register	RLN30LIDB	8	H'FCFE9015	8

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
LIN interface*1	LIN checksum buffer register	RLN30LCBR	8	H'FCFE9016	8
	LIN data buffer 1 register	RLN30LDB1	8	H'FCFE9018	8
	LIN data buffer 2 register	RLN30LDB2	8	H'FCFE9019	8
	LIN data buffer 3 register	RLN30LDB3	8	H'FCFE901A	8
	LIN data buffer 4 register	RLN30LDB4	8	H'FCFE901B	8
	LIN data buffer 5 register	RLN30LDB5	8	H'FCFE901C	8
	LIN data buffer 6 register	RLN30LDB6	8	H'FCFE901D	8
	LIN data buffer 7 register	RLN30LDB7	8	H'FCFE901E	8
	LIN data buffer 8 register	RLN30LDB8	8	H'FCFE901F	8

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Ethernet controller	Software reset register	ARSTR	32	H'E8204800	32
	E-MAC mode register	ECMR0	32	H'E8203500	32
	E-MAC status register	ECSR0	32	H'E8203510	32
	E-MAC interrupt permission register	ECSIPR0	32	H'E8203518	32
	PHY interface register	PIR0	32	H'E8203520	32
	MAC address high register	MAHR0	32	H'E82035C0	32
	MAC address low register	MALR0	32	H'E82035C8	32
	Receive frame length register	RFLR0	32	H'E8203508	32
	CRC error frame receive counter register	CEFCR0	32	H'E8203740	32
	Frame receive error counter register	FRECR0	32	H'E8203748	32
	Too-short frame receive counter register	TSFRCR0	32	H'E8203750	32
	Too-long frame receive counter register	TLFRCR0	32	H'E8203758	32
	Residual-bit frame receive counter register	RFCR0	32	H'E8203760	32
	Multicast address frame receive counter register	MAFCR0	32	H'E8203778	32
	Automatic PAUSE frame register	APR0	32	H'E8203554	32
	Manual PAUSE frame register	MPR0	32	H'E8203558	32
	Automatic PAUSE frame retransmit count register	TPAUSER0	32	H'E8203564	32
	PAUSE frame transmit counter register	PFTCR0	32	H'E820355C	32
	PAUSE frame receive counter register	PFRCR0	32	H'E8203560	32
	TSU counter reset register	TSU_CTRST	32	H'E8204804	32
	CAM entry table specification enable register (common)	TSU_FWSLC	32	H'E8204838	32
	VLANtag set register	TSU_VTAG0	32	H'E8204858	32
	CAM entry table busy register	TSU_ADSBSY	32	H'E8204860	32
	CAM entry table enable register	TSU_TEN	32	H'E8204864	32
	CAM entry table POST 1 register	TSU_POST1	32	H'E8204870	32
	CAM entry table POST 2 register	TSU_POST2	32	H'E8204874	32
	CAM entry table POST 3 register	TSU_POST3	32	H'E8204878	32
	CAM entry table POST 4 register	TSU_POST4	32	H'E820487C	32
	CAM entry table 0H register	TSU_ADRH0	32	H'E8204900	32
	CAM entry table 1H register	TSU_ADRH1	32	H'E8204908	32
	CAM entry table 2H register	TSU_ADRH2	32	H'E8204910	32
	CAM entry table 3H register	TSU_ADRH3	32	H'E8204918	32
	CAM entry table 4H register	TSU_ADRH4	32	H'E8204920	32
	CAM entry table 5H register	TSU_ADRH5	32	H'E8204928	32
	CAM entry table 6H register	TSU_ADRH6	32	H'E8204930	32
	CAM entry table 7H register	TSU_ADRH7	32	H'E8204938	32
	CAM entry table 8H register	TSU_ADRH8	32	H'E8204940	32
	CAM entry table 9H register	TSU_ADRH9	32	H'E8204948	32
	CAM entry table 10H register	TSU_ADRH10	32	H'E8204950	32
	CAM entry table 11H register	TSU_ADRH11	32	H'E8204958	32
	CAM entry table 12H register	TSU_ADRH12	32	H'E8204960	32
	CAM entry table 13H register	TSU_ADRH13	32	H'E8204968	32
	CAM entry table 14H register	TSU_ADRH14	32	H'E8204970	32
	CAM entry table 15H register	TSU_ADRH15	32	H'E8204978	32
	CAM entry table 16H register	TSU_ADRH16	32	H'E8204980	32
	CAM entry table 17H register	TSU_ADRH17	32	H'E8204988	32
	CAM entry table 18H register	TSU_ADRH18	32	H'E8204990	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Ethernet controller	CAM entry table 19H register	TSU_ADRH19	32	H'E8204998	32
	CAM entry table 20H register	TSU_ADRH20	32	H'E82049A0	32
	CAM entry table 21H register	TSU_ADRH21	32	H'E82049A8	32
	CAM entry table 22H register	TSU_ADRH22	32	H'E82049B0	32
	CAM entry table 23H register	TSU_ADRH23	32	H'E82049B8	32
	CAM entry table 24H register	TSU_ADRH24	32	H'E82049C0	32
	CAM entry table 25H register	TSU_ADRH25	32	H'E82049C8	32
	CAM entry table 26H register	TSU_ADRH26	32	H'E82049D0	32
	CAM entry table 27H register	TSU_ADRH27	32	H'E82049D8	32
	CAM entry table 28H register	TSU_ADRH28	32	H'E82049E0	32
	CAM entry table 29H register	TSU_ADRH29	32	H'E82049E8	32
	CAM entry table 30H register	TSU_ADRH30	32	H'E82049F0	32
	CAM entry table 31H register	TSU_ADRH31	32	H'E82049F8	32
	CAM entry table 0L register	TSU_ADRL0	32	H'E8204904	32
	CAM entry table 1L register	TSU_ADRL1	32	H'E820490C	32
	CAM entry table 2L register	TSU_ADRL2	32	H'E8204914	32
	CAM entry table 3L register	TSU_ADRL3	32	H'E820491C	32
	CAM entry table 4L register	TSU_ADRL4	32	H'E8204924	32
	CAM entry table 5L register	TSU_ADRL5	32	H'E820492C	32
	CAM entry table 6L register	TSU_ADRL6	32	H'E8204934	32
	CAM entry table 7L register	TSU_ADRL7	32	H'E820493C	32
	CAM entry table 8L register	TSU_ADRL8	32	H'E8204944	32
	CAM entry table 9L register	TSU_ADRL9	32	H'E820494C	32
	CAM entry table 10L register	TSU_ADRL10	32	H'E8204954	32
	CAM entry table 11L register	TSU_ADRL11	32	H'E820495C	32
	CAM entry table 12L register	TSU_ADRL12	32	H'E8204964	32
	CAM entry table 13L register	TSU_ADRL13	32	H'E820496C	32
	CAM entry table 14L register	TSU_ADRL14	32	H'E8204974	32
	CAM entry table 15L register	TSU_ADRL15	32	H'E820497C	32
	CAM entry table 16L register	TSU_ADRL16	32	H'E8204984	32
	CAM entry table 17L register	TSU_ADRL17	32	H'E820498C	32
	CAM entry table 18L register	TSU_ADRL18	32	H'E8204994	32
	CAM entry table 19L register	TSU_ADRL19	32	H'E820499C	32
	CAM entry table 20L register	TSU_ADRL20	32	H'E82049A4	32
	CAM entry table 21L register	TSU_ADRL21	32	H'E82049AC	32
	CAM entry table 22L register	TSU_ADRL22	32	H'E82049B4	32
	CAM entry table 23L register	TSU_ADRL23	32	H'E82049BC	32
	CAM entry table 24L register	TSU_ADRL24	32	H'E82049C4	32
	CAM entry table 25L register	TSU_ADRL25	32	H'E82049CC	32
	CAM entry table 26L register	TSU_ADRL26	32	H'E82049D4	32
	CAM entry table 27L register	TSU_ADRL27	32	H'E82049DC	32
	CAM entry table 28L register	TSU_ADRL28	32	H'E82049E4	32
	CAM entry table 29L register	TSU_ADRL29	32	H'E82049EC	32
	CAM entry table 30L register	TSU_ADRL30	32	H'E82049F4	32
	CAM entry table 31L register	TSU_ADRL31	32	H'E82049FC	32
	Transmit frame counter register	TXNLCR0	32	H'E8204880	32
	Transmit frame counter register	TXALCR0	32	H'E8204884	32
	Receive frame counter register	RXNLCR0	32	H'E8204888	32
	Receive frame counter register	RXALCR0	32	H'E820488C	32
	E-DMAC start register	EDSR0	32	H'E8203000	32
	E-DMAC mode register	EDMR0	32	H'E8203400	32
	E-DMAC transmit request register	EDTRR0	32	H'E8203408	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Ethernet controller	E-DMAC receive request register	EDRRR0	32	H'E8203410	32
	E-MAC/E-DMAC status register	EESR0	32	H'E8203428	32
	E-MAC/E-DMAC status interrupt permission register	EESIPR0	32	H'E8203430	32
	Transmit descriptor list start address register	TDLAR0	32	H'E8203010	32
	Transmit descriptor fetch address register	TDFAR0	32	H'E8203014	32
	Transmit descriptor finished address register	TDFXR0	32	H'E8203018	32
	Transmit descriptor final flag register	TDFFR0	32	H'E820301C	32
	Receive descriptor list start address register	RDLAR0	32	H'E8203030	32
	Receive descriptor fetch address register	RDFAR0	32	H'E8203034	32
	Receive descriptor finished address register	RDFXR0	32	H'E8203038	32
	Receive descriptor final flag register	RDFFR0	32	H'E820303C	32
	Transmit/receive status copy enable register	TRSCER0	32	H'E8203438	32
	Receive missed-frame counter register	RMFCR0	32	H'E8203440	32
	Transmit FIFO threshold register	TFTR0	32	H'E8203448	32
	FIFO depth register	FDR0	32	H'E8203450	32
	Receiving method control register	RMCR0	32	H'E8203458	32
	Receive data padding insert register	RPADIR0	32	H'E8203460	32
	Overflow alert FIFO threshold register	FCFTR0	32	H'E8203468	32
	Intelligent checksum mode register	CSMR	32	H'E82034E4	32
	Intelligent checksum skipped bytes monitor register	CSSBM	32	H'E82034E8	32
	Intelligent checksum monitor register	CSSMR	32	H'E82034EC	32
A/D converter	A/D data register A	ADDRA	16	H'E8005800	16
	A/D data register B	ADDRB	16	H'E8005802	16
	A/D data register C	ADDRC	16	H'E8005804	16
	A/D data register D	ADDRD	16	H'E8005806	16
	A/D data register E	ADDRE	16	H'E8005808	16
	A/D data register F	ADDRF	16	H'E800580A	16
	A/D data register G	ADDRG	16	H'E800580C	16
	A/D data register H	ADDRH	16	H'E800580E	16
	A/D comparison upper limit value register A	ADCMPHA	16	H'E8005820	16
	A/D comparison lower limit value register A	ADCMPLA	16	H'E8005822	16
	A/D comparison upper limit value register B	ADCMPHB	16	H'E8005824	16
	A/D comparison lower limit value register B	ADCMPLB	16	H'E8005826	16
	A/D comparison upper limit value register C	ADCMPHC	16	H'E8005828	16
	A/D comparison lower limit value register C	ADCMPLC	16	H'E800582A	16
	A/D comparison upper limit value register D	ADCMPHD	16	H'E800582C	16
	A/D comparison lower limit value register D	ADCMPLD	16	H'E800582E	16
	A/D comparison upper limit value register E	ADCMPHE	16	H'E8005830	16
	A/D comparison lower limit value register E	ADCMPLE	16	H'E8005832	16
	A/D comparison upper limit value register F	ADCMPHF	16	H'E8005834	16
	A/D comparison lower limit value register F	ADCMPLF	16	H'E8005836	16
	A/D comparison upper limit value register G	ADCMPHG	16	H'E8005838	16
	A/D comparison lower limit value register G	ADCMPLG	16	H'E800583A	16
	A/D comparison upper limit value register H	ADCMPHH	16	H'E800583C	16
	A/D comparison lower limit value register H	ADCMPLH	16	H'E800583E	16
	A/D control/status register	ADCSR	16	H'E8005860	16
	A/D comparison interrupt enable register	ADCMPER	16	H'E8005862	16
	A/D comparison status register	ADCMPSR	16	H'E8005864	16

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
USB2.0 host/function module	System configuration control register_0	SYSCFG0_0	16	H'E8010000	16
	CPU bus wait setting register_0	BUSWAIT_0	16	H'E8010002	16
	System configuration status register_0	SYSSTS0_0	16	H'E8010004	16
	Device state control register 0_0	DVSTCTR0_0	16	H'E8010008	16
	Test mode register_0	TESTMODE_0	16	H'E801000C	16
	DMA0-FIFO bus configuration register_0	D0FBCFG_0	16	H'E8010010	16
	DMA1-FIFO bus configuration register_0	D1FBCFG_0	16	H'E8010012	16
	CFIFO port register_0	CFIFO_0	32	H'E8010014	8, 16, 32
	D0FIFO port register_0	D0FIFO_0	32	H'E8010018	8, 16, 32
	D1FIFO port register_0	D1FIFO_0	32	H'E801001C	8, 16, 32
	CFIFO port select register_0	CFIFOSEL_0	16	H'E8010020	16
	CFIFO port control register_0	CFIFOCTR_0	16	H'E8010022	16
	D0FIFO port select register_0	D0FIFOSEL_0	16	H'E8010028	16
	D0FIFO port control register_0	D0FIFOCTR_0	16	H'E801002A	16
	D1FIFO port select register_0	D1FIFOSEL_0	16	H'E801002C	16
	D1FIFO port control register_0	D1FIFOCTR_0	16	H'E801002E	16
	Interrupt enable register 0_0	INTENB0_0	16	H'E8010030	16
	Interrupt enable register 1_0	INTENB1_0	16	H'E8010032	16
	BRDY interrupt enable register_0	BRDYENB_0	16	H'E8010036	16
	NRDY interrupt enable register_0	NRDYENB_0	16	H'E8010038	16
	BEMP interrupt enable register_0	BEMPENB_0	16	H'E801003A	16
	SOF output configuration register_0	SOFCFG_0	16	H'E801003C	16
	Interrupt status register 0_0	INTSTS0_0	16	H'E8010040	16
	Interrupt status register 1_0	INTSTS1_0	16	H'E8010042	16
	BRDY interrupt status register_0	BRDYSTS_0	16	H'E8010046	16
	NRDY interrupt status register_0	NRDYSTS_0	16	H'E8010048	16
	BEMP interrupt status register_0	BEMPSTS_0	16	H'E801004A	16
	Frame number register_0	FRMNUM_0	16	H'E801004C	16
	µFrame number register_0	UFRMNUM_0	16	H'E801004E	16
	USB address register_0	USBADDR_0	16	H'E8010050	16
	USB request type register_0	USBREQ_0	16	H'E8010054	16
	USB request value register_0	USBVAL_0	16	H'E8010056	16
	USB request index register_0	USBINDX_0	16	H'E8010058	16
	USB request length register_0	USBLENG_0	16	H'E801005A	16
	DCP configuration register_0	DCPCFG_0	16	H'E801005C	16
	DCP maximum packet size register_0	DCPMAXP_0	16	H'E801005E	16
	DCP control register_0	DCPCTR_0	16	H'E8010060	16
	Pipe window select register_0	PIPESEL_0	16	H'E8010064	16
	Pipe configuration register_0	PIPECFG_0	16	H'E8010068	16
	Pipe buffer setting register_0	PIPEBUF_0	16	H'E801006A	16
	Pipe maximum packet size register_0	PIPEMAXP_0	16	H'E801006C	16
	Pipe timing control register_0	PIPEPERI_0	16	H'E801006E	16
	Pipe 1 control register_0	PIPE1CTR_0	16	H'E8010070	16
	Pipe 2 control register_0	PIPE2CTR_0	16	H'E8010072	16
	Pipe 3 control register_0	PIPE3CTR_0	16	H'E8010074	16
	Pipe 4 control register_0	PIPE4CTR_0	16	H'E8010076	16
	Pipe 5 control register_0	PIPE5CTR_0	16	H'E8010078	16
	Pipe 6 control register_0	PIPE6CTR_0	16	H'E801007A	16
	Pipe 7 control register_0	PIPE7CTR_0	16	H'E801007C	16
	Pipe 8 control register_0	PIPE8CTR_0	16	H'E801007E	16
	Pipe 9 control register_0	PIPE9CTR_0	16	H'E8010080	16
	Pipe A control register_0	PIPEACTR_0	16	H'E8010082	16

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
USB2.0 host/function module	Pipe B control register_0	PIPEBCTR_0	16	H'E8010084	16
	Pipe C control register_0	PIPECCTR_0	16	H'E8010086	16
	Pipe D control register_0	PIPEDCTR_0	16	H'E8010088	16
	Pipe E control register_0	PIPEECTR_0	16	H'E801008A	16
	Pipe F control register_0	PIPEFCTR_0	16	H'E801008C	16
	Pipe 1 transaction counter enable register_0	PIPE1TRE_0	16	H'E8010090	16
	Pipe 1 transaction counter register_0	PIPE1TRN_0	16	H'E8010092	16
	Pipe 2 transaction counter enable register_0	PIPE2TRE_0	16	H'E8010094	16
	Pipe 2 transaction counter register_0	PIPE2TRN_0	16	H'E8010096	16
	Pipe 3 transaction counter enable register_0	PIPE3TRE_0	16	H'E8010098	16
	Pipe 3 transaction counter register_0	PIPE3TRN_0	16	H'E801009A	16
	Pipe 4 transaction counter enable register_0	PIPE4TRE_0	16	H'E801009C	16
	Pipe 4 transaction counter register_0	PIPE4TRN_0	16	H'E801009E	16
	Pipe 5 transaction counter enable register_0	PIPE5TRE_0	16	H'E80100A0	16
	Pipe 5 transaction counter register_0	PIPE5TRN_0	16	H'E80100A2	16
	Pipe B transaction counter enable register_0	PIPEBTRE_0	16	H'E80100A4	16
	Pipe B transaction counter register_0	PIPEBTRN_0	16	H'E80100A6	16
	Pipe C transaction counter enable register_0	PIPECTRE_0	16	H'E80100A8	16
	Pipe C transaction counter register_0	PIPECTRN_0	16	H'E80100AA	16
	Pipe D transaction counter enable register_0	PIPEDTRE_0	16	H'E80100AC	16
	Pipe D transaction counter register_0	PIPEDTRN_0	16	H'E80100AE	16
	Pipe E transaction counter enable register_0	PIPEETRE_0	16	H'E80100B0	16
	Pipe E transaction counter register_0	PIPEETRN_0	16	H'E80100B2	16
	Pipe F transaction counter enable register_0	PIPEFTRE_0	16	H'E80100B4	16
	Pipe F transaction counter register_0	PIPEFTRN_0	16	H'E80100B6	16
	Pipe 9 transaction counter enable register_0	PIPE9TRE_0	16	H'E80100B8	16
	Pipe 9 transaction counter register_0	PIPE9TRN_0	16	H'E80100BA	16
	Pipe A transaction counter enable register_0	PIPEATRE_0	16	H'E80100BC	16
	Pipe A transaction counter register_0	PIPEATRN_0	16	H'E80100BE	16
	Device address 0 configuration register_0	DEVADD0_0	16	H'E80100D0	16
	Device address 1 configuration register_0	DEVADD1_0	16	H'E80100D2	16
	Device address 2 configuration register_0	DEVADD2_0	16	H'E80100D4	16
	Device address 3 configuration register_0	DEVADD3_0	16	H'E80100D6	16
	Device address 4 configuration register_0	DEVADD4_0	16	H'E80100D8	16
	Device address 5 configuration register_0	DEVADD5_0	16	H'E80100DA	16
	Device address 6 configuration register_0	DEVADD6_0	16	H'E80100DC	16
	Device address 7 configuration register_0	DEVADD7_0	16	H'E80100DE	16
	Device address 8 configuration register_0	DEVADD8_0	16	H'E80100E0	16
	Device address 9 configuration register_0	DEVADD9_0	16	H'E80100E2	16
	Device address A configuration register_0	DEVADDA_0	16	H'E80100E4	16
	Suspend mode register_0	SUSPMODE_0	16	H'E8010102	16
	D0FIFO continuous transfer port register 0_0	D0FIFOB0_0	32	H'E8010160	32
	D0FIFO continuous transfer port register 1_0	D0FIFOB1_0	32	H'E8010164	32
	D0FIFO continuous transfer port register 2_0	D0FIFOB2_0	32	H'E8010168	32
	D0FIFO continuous transfer port register 3_0	D0FIFOB3_0	32	H'E801016C	32
	D0FIFO continuous transfer port register 4_0	D0FIFOB4_0	32	H'E8010170	32
	D0FIFO continuous transfer port register 5_0	D0FIFOB5_0	32	H'E8010174	32
	D0FIFO continuous transfer port register 6_0	D0FIFOB6_0	32	H'E8010178	32
	D0FIFO continuous transfer port register 7_0	D0FIFOB7_0	32	H'E801017C	32
	D1FIFO continuous transfer port register 0_0	D1FIFOB0_0	32	H'E8010180	32
	D1FIFO continuous transfer port register 1_0	D1FIFOB1_0	32	H'E8010184	32
	D1FIFO continuous transfer port register 2_0	D1FIFOB2_0	32	H'E8010188	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
USB2.0 host/function module	D1FIFO continuous transfer port register 3_0	D1FIFOB3_0	32	H'E801018C	32
	D1FIFO continuous transfer port register 4_0	D1FIFOB4_0	32	H'E8010190	32
	D1FIFO continuous transfer port register 5_0	D1FIFOB5_0	32	H'E8010194	32
	D1FIFO continuous transfer port register 6_0	D1FIFOB6_0	32	H'E8010198	32
	D1FIFO continuous transfer port register 7_0	D1FIFOB7_0	32	H'E801019C	32
	System configuration control register_1	SYSCFG0_1	16	H'E8207000	16
	CPU bus wait setting register_1	BUSWAIT_1	16	H'E8207002	16
	System configuration status register_1	SYSSTS0_1	16	H'E8207004	16
	Device state control register 0_1	DVSTCTR0_1	16	H'E8207008	16
	Test mode register_1	TESTMODE_1	16	H'E820700C	16
	DMA0-FIFO bus configuration register_1	D0FBCFG_1	16	H'E8207010	16
	DMA1-FIFO bus configuration register_1	D1FBCFG_1	16	H'E8207012	16
	CFIFO port register_1	CFIFO_1	32	H'E8207014	8, 16, 32
	D0FIFO port register_1	D0FIFO_1	32	H'E8207018	8, 16, 32
	D1FIFO port register_1	D1FIFO_1	32	H'E820701C	8, 16, 32
	CFIFO port select register_1	CFIFOSEL_1	16	H'E8207020	16
	CFIFO port control register_1	CFIFOCTR_1	16	H'E8207022	16
	D0FIFO port select register_1	D0FIFOSEL_1	16	H'E8207028	16
	D0FIFO port control register_1	D0FIFOCTR_1	16	H'E820702A	16
	D1FIFO port select register_1	D1FIFOSEL_1	16	H'E820702C	16
	D1FIFO port control register_1	D1FIFOCTR_1	16	H'E820702E	16
	Interrupt enable register 0_1	INTENB0_1	16	H'E8207030	16
	Interrupt enable register 1_1	INTENB1_1	16	H'E8207032	16
	BRDY interrupt enable register_1	BRDYENB_1	16	H'E8207036	16
	NRDY interrupt enable register_1	NRDYENB_1	16	H'E8207038	16
	BEMP interrupt enable register_1	BEMPENB_1	16	H'E820703A	16
	SOF output configuration register_1	SOFCFG_1	16	H'E820703C	16
	Interrupt status register 0_1	INTSTS0_1	16	H'E8207040	16
	Interrupt status register 1_1	INTSTS1_1	16	H'E8207042	16
	BRDY interrupt status register_1	BRDYSTS_1	16	H'E8207046	16
	NRDY interrupt status register_1	NRDYSTS_1	16	H'E8207048	16
	BEMP interrupt status register_1	BEMPSTS_1	16	H'E820704A	16
	Frame number register_1	FRMNUM_1	16	H'E820704C	16
	µFrame number register_1	UFRMNUM_1	16	H'E820704E	16
	USB address register_1	USBADDR_1	16	H'E8207050	16
	USB request type register_1	USBREQ_1	16	H'E8207054	16
	USB request value register_1	USBVAL_1	16	H'E8207056	16
	USB request index register_1	USBINDX_1	16	H'E8207058	16
	USB request length register_1	USBLENG_1	16	H'E820705A	16
	DCP configuration register_1	DCPCFG_1	16	H'E820705C	16
	DCP maximum packet size register_1	DCPMAXP_1	16	H'E820705E	16
	DCP control register_1	DCPCTR_1	16	H'E8207060	16
	Pipe window select register_1	PIPESEL_1	16	H'E8207064	16
	Pipe configuration register_1	PIPECFG_1	16	H'E8207068	16
	Pipe buffer setting register_1	PIPEBUF_1	16	H'E820706A	16
	Pipe maximum packet size register_1	PIPEMAXP_1	16	H'E820706C	16
	Pipe timing control register_1	PIPEPERI_1	16	H'E820706E	16
	Pipe 1 control register_1	PIPE1CTR_1	16	H'E8207070	16
	Pipe 2 control register_1	PIPE2CTR_1	16	H'E8207072	16
	Pipe 3 control register_1	PIPE3CTR_1	16	H'E8207074	16
	Pipe 4 control register_1	PIPE4CTR_1	16	H'E8207076	16
	Pipe 5 control register_1	PIPE5CTR_1	16	H'E8207078	16

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
USB2.0 host/function module	Pipe 6 control register_1	PIPE6CTR_1	16	H'E820707A	16
	Pipe 7 control register_1	PIPE7CTR_1	16	H'E820707C	16
	Pipe 8 control register_1	PIPE8CTR_1	16	H'E820707E	16
	Pipe 9 control register_1	PIPE9CTR_1	16	H'E8207080	16
	Pipe A control register_1	PIPEACTR_1	16	H'E8207082	16
	Pipe B control register_1	PIPEBCTR_1	16	H'E8207084	16
	Pipe C control register_1	PIPECCTR_1	16	H'E8207086	16
	Pipe D control register_1	PIPEDCTR_1	16	H'E8207088	16
	Pipe E control register_1	PIPEECTR_1	16	H'E820708A	16
	Pipe F control register_1	PIPEFCTR_1	16	H'E820708C	16
	Pipe 1 transaction counter enable register_1	PIPE1TRE_1	16	H'E8207090	16
	Pipe 1 transaction counter register_1	PIPE1TRN_1	16	H'E8207092	16
	Pipe 2 transaction counter enable register_1	PIPE2TRE_1	16	H'E8207094	16
	Pipe 2 transaction counter register_1	PIPE2TRN_1	16	H'E8207096	16
	Pipe 3 transaction counter enable register_1	PIPE3TRE_1	16	H'E8207098	16
	Pipe 3 transaction counter register_1	PIPE3TRN_1	16	H'E820709A	16
	Pipe 4 transaction counter enable register_1	PIPE4TRE_1	16	H'E820709C	16
	Pipe 4 transaction counter register_1	PIPE4TRN_1	16	H'E820709E	16
	Pipe 5 transaction counter enable register_1	PIPE5TRE_1	16	H'E82070A0	16
	Pipe 5 transaction counter register_1	PIPE5TRN_1	16	H'E82070A2	16
	Pipe B transaction counter enable register_1	PIPEBTRE_1	16	H'E82070A4	16
	Pipe B transaction counter register_1	PIPEBTRN_1	16	H'E82070A6	16
	Pipe C transaction counter enable register_1	PIPECTRE_1	16	H'E82070A8	16
	Pipe C transaction counter register_1	PIPECTRN_1	16	H'E82070AA	16
	Pipe D transaction counter enable register_1	PIPEDTRE_1	16	H'E82070AC	16
	Pipe D transaction counter register_1	PIPEDTRN_1	16	H'E82070AE	16
	Pipe E transaction counter enable register_1	PIPEETRE_1	16	H'E82070B0	16
	Pipe E transaction counter register_1	PIPEETRN_1	16	H'E82070B2	16
	Pipe F transaction counter enable register_1	PIPEFTRE_1	16	H'E82070B4	16
	Pipe F transaction counter register_1	PIPEFTRN_1	16	H'E82070B6	16
	Pipe 9 transaction counter enable register_1	PIPE9TRE_1	16	H'E82070B8	16
	Pipe 9 transaction counter register_1	PIPE9TRN_1	16	H'E82070BA	16
	Pipe A transaction counter enable register_1	PIPEATRE_1	16	H'E82070BC	16
	Pipe A transaction counter register_1	PIPEATRN_1	16	H'E82070BE	16
	Device address 0 configuration register_1	DEVADD0_1	16	H'E82070D0	16
	Device address 1 configuration register_1	DEVADD1_1	16	H'E82070D2	16
	Device address 2 configuration register_1	DEVADD2_1	16	H'E82070D4	16
	Device address 3 configuration register_1	DEVADD3_1	16	H'E82070D6	16
	Device address 4 configuration register_1	DEVADD4_1	16	H'E82070D8	16
	Device address 5 configuration register_1	DEVADD5_1	16	H'E82070DA	16
	Device address 6 configuration register_1	DEVADD6_1	16	H'E82070DC	16
	Device address 7 configuration register_1	DEVADD7_1	16	H'E82070DE	16
	Device address 8 configuration register_1	DEVADD8_1	16	H'E82070E0	16
	Device address 9 configuration register_1	DEVADD9_1	16	H'E82070E2	16
	Device address A configuration register_1	DEVADDA_1	16	H'E82070E4	16
	Suspend mode register_1	SUSPMODE_1	16	H'E8207102	16
	D0FIFO continuous transfer port register 0_1	D0FIFOB0_1	32	H'E8207160	32
	D0FIFO continuous transfer port register 1_1	D0FIFOB1_1	32	H'E8207164	32
	D0FIFO continuous transfer port register 2_1	D0FIFOB2_1	32	H'E8207168	32
	D0FIFO continuous transfer port register 3_1	D0FIFOB3_1	32	H'E820716C	32
	D0FIFO continuous transfer port register 4_1	D0FIFOB4_1	32	H'E8207170	32
	D0FIFO continuous transfer port register 5_1	D0FIFOB5_1	32	H'E8207174	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
USB2.0 host/function module	D0FIFO continuous transfer port register 6_1	D0FIFOB6_1	32	H'E8207178	32
	D0FIFO continuous transfer port register 7_1	D0FIFOB7_1	32	H'E820717C	32
	D1FIFO continuous transfer port register 0_1	D1FIFOB0_1	32	H'E8207180	32
	D1FIFO continuous transfer port register 1_1	D1FIFOB1_1	32	H'E8207184	32
	D1FIFO continuous transfer port register 2_1	D1FIFOB2_1	32	H'E8207188	32
	D1FIFO continuous transfer port register 3_1	D1FIFOB3_1	32	H'E820718C	32
	D1FIFO continuous transfer port register 4_1	D1FIFOB4_1	32	H'E8207190	32
	D1FIFO continuous transfer port register 5_1	D1FIFOB5_1	32	H'E8207194	32
	D1FIFO continuous transfer port register 6_1	D1FIFOB6_1	32	H'E8207198	32
	D1FIFO continuous transfer port register 7_1	D1FIFOB7_1	32	H'E820719C	32
Video display controller 5	External input block register update control register	INP_UPDATE	32	H'FCFF7400	32
	Input select control register	INP_SEL_CNT	32	H'FCFF7404	32
	External input sync signal control register	INP_EXT_SYNC_CNT	32	H'FCFF7408	32
	Vsync signal phase adjustment register	INP_VSYNC_PH_ADJ	32	H'FCFF740C	32
	Sync signal phase adjustment register	INP_DLY_ADJ	32	H'FCFF7410	32
	Image quality adjustment block register update control register	IMGCNT_UPDATE	32	H'FCFF7480	32
	NR control register 0	IMGCNT_NR_CNT0	32	H'FCFF7484	32
	NR control register 1	IMGCNT_NR_CNT1	32	H'FCFF7488	32
	Image quality adjustment block matrix mode register	IMGCNT_MTX_MODE	32	H'FCFF74A0	32
	Image quality adjustment block matrix YG adjustment register 0	IMGCNT_MTX_YG_ADJ0	32	H'FCFF74A4	32
	Image quality adjustment block matrix YG adjustment register 1	IMGCNT_MTX_YG_ADJ1	32	H'FCFF74A8	32
	Image quality adjustment block matrix CBB adjustment register 0	IMGCNT_MTX_CBB_ADJ0	32	H'FCFF74AC	32
	Image quality adjustment block matrix CBB adjustment register 1	IMGCNT_MTX_CBB_ADJ1	32	H'FCFF74B0	32
	Image quality adjustment block matrix CRR adjustment register 0	IMGCNT_MTX_CRR_ADJ0	32	H'FCFF74B4	32
	Image quality adjustment block matrix CRR adjustment register 1	IMGCNT_MTX_CRR_ADJ1	32	H'FCFF74B8	32
	SCL0 register update control register (SC0)	SC0_SCL0_UPDATE	32	H'FCFF7500	32
	Mask control register (SC0)	SC0_SCL0_FRC1	32	H'FCFF7504	32
	Missing Vsync compensation control register (SC0)	SC0_SCL0_FRC2	32	H'FCFF7508	32
	Output sync select register (SC0)	SC0_SCL0_FRC3	32	H'FCFF750C	32
	Free-running period control register (SC0)	SC0_SCL0_FRC4	32	H'FCFF7510	32
	Output delay control register (SC0)	SC0_SCL0_FRC5	32	H'FCFF7514	32
	Full-screen vertical size register (SC0)	SC0_SCL0_FRC6	32	H'FCFF7518	32
	Full-screen horizontal size register (SC0)	SC0_SCL0_FRC7	32	H'FCFF751C	32
	Vsync detection register (SC0)	SC0_SCL0_FRC9	32	H'FCFF7524	32
	Status monitor 0 register (SC0)	SC0_SCL0_MON0	16	H'FCFF7528	16
	Interrupt control register (SC0)	SC0_SCL0_INT	16	H'FCFF752A	16
	Scaling-down control register (SC0)	SC0_SCL0_DS1	32	H'FCFF752C	32
	Vertical capture size register (SC0)	SC0_SCL0_DS2	32	H'FCFF7530	32
	Horizontal capture size register (SC0)	SC0_SCL0_DS3	32	H'FCFF7534	32
	Horizontal scale down register (SC0)	SC0_SCL0_DS4	32	H'FCFF7538	32
	Initial vertical phase register (SC0)	SC0_SCL0_DS5	32	H'FCFF753C	32
	Vertical scaling register (SC0)	SC0_SCL0_DS6	32	H'FCFF7540	32
	Scaling-down control block output size register (SC0)	SC0_SCL0_DS7	32	H'FCFF7544	32
	Scaling-up control register (SC0)	SC0_SCL0_US1	32	H'FCFF7548	32
	Output image vertical size register (SC0)	SC0_SCL0_US2	32	H'FCFF754C	32
	Output image horizontal size register (SC0)	SC0_SCL0_US3	32	H'FCFF7550	32
	Scaling-up control block input size register (SC0)	SC0_SCL0_US4	32	H'FCFF7554	32
	Horizontal scale up register (SC0)	SC0_SCL0_US5	32	H'FCFF7558	32
	Horizontal scale up initial phase register (SC0)	SC0_SCL0_US6	32	H'FCFF755C	32
	Trimming register (SC0)	SC0_SCL0_US7	32	H'FCFF7560	32
	Frame buffer read select register (SC0)	SC0_SCL0_US8	32	H'FCFF7564	32
	Background color register (SC0)	SC0_SCL0_OVR1	32	H'FCFF756C	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Video display controller 5	SCL1 register update control register (SC0)	SC0_SCL1_UPDATE	32	H'FCFF7580	32
	Writing mode register (SC0)	SC0_SCL1_WR1	32	H'FCFF7588	32
	Write address register 1T (SC0)	SC0_SCL1_WR2	32	H'FCFF758C	32
	Write address register 2T (SC0)	SC0_SCL1_WR3	32	H'FCFF7590	32
	Write address register 3T (SC0)	SC0_SCL1_WR4	32	H'FCFF7594	32
	Frame sub-sampling register (SC0)	SC0_SCL1_WR5	32	H'FCFF759C	32
	Bit reduction register (SC0)	SC0_SCL1_WR6	32	H'FCFF75A0	32
	Write detection register (SC0)	SC0_SCL1_WR7	32	H'FCFF75A4	32
	Write address register 1B (SC0)	SC0_SCL1_WR8	32	H'FCFF75A8	32
	Write address register 2B (SC0)	SC0_SCL1_WR9	32	H'FCFF75AC	32
	Write address register 3B (SC0)	SC0_SCL1_WR10	32	H'FCFF75B0	32
	Write detection register B (SC0)	SC0_SCL1_WR11	32	H'FCFF75B4	32
	Status monitor 1 register (SC0)	SC0_SCL1_MON1	32	H'FCFF75B8	32
	Pointer buffer 0 register (SC0)	SC0_SCL1_PBUF0	32	H'FCFF75BC	32
	Pointer buffer 1 register (SC0)	SC0_SCL1_PBUF1	32	H'FCFF75C0	32
	Pointer buffer 2 register (SC0)	SC0_SCL1_PBUF2	32	H'FCFF75C4	32
	Pointer buffer 3 register (SC0)	SC0_SCL1_PBUF3	32	H'FCFF75C8	32
	Pointer buffer and field information register (SC0)	SC0_SCL1_PBUF_FLD	32	H'FCFF75CC	32
	Pointer buffer control register (SC0)	SC0_SCL1_PBUF_CNT	32	H'FCFF75D0	32
	Graphics 0 register update control register	GR0_UPDATE	32	H'FCFF7600	32
	Frame buffer read control register (graphics 0)	GR0_FLM_RD	32	H'FCFF7604	32
	Frame buffer control register 1 (graphics 0)	GR0_FLM1	32	H'FCFF7608	32
	Frame buffer control register 2 (graphics 0)	GR0_FLM2	32	H'FCFF760C	32
	Frame buffer control register 3 (graphics 0)	GR0_FLM3	32	H'FCFF7610	32
	Frame buffer control register 4 (graphics 0)	GR0_FLM4	32	H'FCFF7614	32
	Frame buffer control register 5 (graphics 0)	GR0_FLM5	32	H'FCFF7618	32
	Frame buffer control register 6 (graphics 0)	GR0_FLM6	32	H'FCFF761C	32
	Alpha blending control register 1 (graphics 0)	GR0_AB1	32	H'FCFF7620	32
	Alpha blending control register 2 (graphics 0)	GR0_AB2	32	H'FCFF7624	32
	Alpha blending control register 3 (graphics 0)	GR0_AB3	32	H'FCFF7628	32
	Alpha blending control register 7 (graphics 0)	GR0_AB7	32	H'FCFF7638	32
	Alpha blending control register 8 (graphics 0)	GR0_AB8	32	H'FCFF763C	32
	Alpha blending control register 9 (graphics 0)	GR0_AB9	32	H'FCFF7640	32
	Alpha blending control register 10 (graphics 0)	GR0_AB10	32	H'FCFF7644	32
	Alpha blending control register 11 (graphics 0)	GR0_AB11	32	H'FCFF7648	32
	Background color control register (graphics 0)	GR0_BASE	32	H'FCFF764C	32
	CLUT table control register (graphics 0)	GR0_CLUT	32	H'FCFF7650	32
	Register update control register in image quality improver (image quality improver 0)	ADJ0_UPDATE	32	H'FCFF7680	32
	Black stretch register (image quality improver 0)	ADJ0_BKSTR_SET	32	H'FCFF7684	32
	Enhancer timing adjustment register 1 (image quality improver 0)	ADJ0_ENH_TIM1	32	H'FCFF7688	32
	Enhancer timing adjustment register 2 (image quality improver 0)	ADJ0_ENH_TIM2	32	H'FCFF768C	32
	Enhancer timing adjustment register 3 (image quality improver 0)	ADJ0_ENH_TIM3	32	H'FCFF7690	32
	Enhancer sharpness register 1 (image quality improver 0)	ADJ0_ENH_SHP1	32	H'FCFF7694	32
	Enhancer sharpness register 2 (image quality improver 0)	ADJ0_ENH_SHP2	32	H'FCFF7698	32
	Enhancer sharpness register 3 (image quality improver 0)	ADJ0_ENH_SHP3	32	H'FCFF769C	32
	Enhancer sharpness register 4 (image quality improver 0)	ADJ0_ENH_SHP4	32	H'FCFF76A0	32
	Enhancer sharpness register 5 (image quality improver 0)	ADJ0_ENH_SHP5	32	H'FCFF76A4	32
	Enhancer sharpness register 6 (image quality improver 0)	ADJ0_ENH_SHP6	32	H'FCFF76A8	32
	Enhancer LTI register 1 (image quality improver 0)	ADJ0_ENH_LTI1	32	H'FCFF76AC	32
	Enhancer LTI register 2 (image quality improver 0)	ADJ0_ENH_LTI2	32	H'FCFF76B0	32
	Matrix mode register in image quality improver (image quality improver 0)	ADJ0_MTX_MODE	32	H'FCFF76B4	32
	Matrix YG control register 0 in image quality improver (image quality improver 0)	ADJ0_MTX_YG_ADJ0	32	H'FCFF76B8	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Video display controller 5	Matrix YG control register 1 in image quality improver (image quality improver 0)	ADJ0_MTX_YG_ADJ1	32	H'FCFF76BC	32
	Matrix CBB control register 0 in image quality improver (image quality improver 0)	ADJ0_MTX_CBB_ADJ0	32	H'FCFF76C0	32
	Matrix CBB control register 1 in image quality improver (image quality improver 0)	ADJ0_MTX_CBB_ADJ1	32	H'FCFF76C4	32
	Matrix CRR control register 0 in image quality improver (image quality improver 0)	ADJ0_MTX_CRR_ADJ0	32	H'FCFF76C8	32
	Matrix CRR control register 1 in image quality improver (image quality improver 0)	ADJ0_MTX_CRR_ADJ1	32	H'FCFF76CC	32
	Graphics 2 register update control register	GR2_UPDATE	32	H'FCFF7700	32
	Frame buffer read control register (Graphics 2)	GR2_FLM_RD	32	H'FCFF7704	32
	Frame buffer control register 1 (Graphics 2)	GR2_FLM1	32	H'FCFF7708	32
	Frame buffer control register 2 (Graphics 2)	GR2_FLM2	32	H'FCFF770C	32
	Frame buffer control register 3 (Graphics 2)	GR2_FLM3	32	H'FCFF7710	32
	Frame buffer control register 4 (Graphics 2)	GR2_FLM4	32	H'FCFF7714	32
	Frame buffer control register 5 (Graphics 2)	GR2_FLM5	32	H'FCFF7718	32
	Frame buffer control register 6 (Graphics 2)	GR2_FLM6	32	H'FCFF771C	32
	Alpha blending control register 1 (Graphics 2)	GR2_AB1	32	H'FCFF7720	32
	Alpha blending control register 2 (Graphics 2)	GR2_AB2	32	H'FCFF7724	32
	Alpha blending control register 3 (Graphics 2)	GR2_AB3	32	H'FCFF7728	32
	Alpha blending control register 4 (Graphics 2)	GR2_AB4	32	H'FCFF772C	32
	Alpha blending control register 5 (Graphics 2)	GR2_AB5	32	H'FCFF7730	32
	Alpha blending control register 6 (Graphics 2)	GR2_AB6	32	H'FCFF7734	32
	Alpha blending control register 7 (Graphics 2)	GR2_AB7	32	H'FCFF7738	32
	Alpha blending control register 8 (Graphics 2)	GR2_AB8	32	H'FCFF773C	32
	Alpha blending control register 9 (Graphics 2)	GR2_AB9	32	H'FCFF7740	32
	Alpha blending control register 10 (Graphics 2)	GR2_AB10	32	H'FCFF7744	32
	Alpha blending control register 11 (Graphics 2)	GR2_AB11	32	H'FCFF7748	32
	Background color control register (Graphics 2)	GR2_BASE	32	H'FCFF774C	32
	CLUT table control register (Graphics 2)	GR2_CLUT	32	H'FCFF7750	32
	Status monitor register (Graphics 2)	GR2_MON	32	H'FCFF7754	32
	Graphics 3 register update control register	GR3_UPDATE	32	H'FCFF7780	32
	Frame buffer read control register (Graphics 3)	GR3_FLM_RD	32	H'FCFF7784	32
	Frame buffer control register 1 (Graphics 3)	GR3_FLM1	32	H'FCFF7788	32
	Frame buffer control register 2 (Graphics 3)	GR3_FLM2	32	H'FCFF778C	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Video display controller 5	Frame buffer control register 3 (Graphics 3)	GR3_FLM3	32	H'FCFF7790	32
	Frame buffer control register 4 (Graphics 3)	GR3_FLM4	32	H'FCFF7794	32
	Frame buffer control register 5 (Graphics 3)	GR3_FLM5	32	H'FCFF7798	32
	Frame buffer control register 6 (Graphics 3)	GR3_FLM6	32	H'FCFF779C	32
	Alpha blending control register 1 (Graphics 3)	GR3_AB1	32	H'FCFF77A0	32
	Alpha blending control register 2 (Graphics 3)	GR3_AB2	32	H'FCFF77A4	32
	Alpha blending control register 3 (Graphics 3)	GR3_AB3	32	H'FCFF77A8	32
	Alpha blending control register 4 (Graphics 3)	GR3_AB4	32	H'FCFF77AC	32
	Alpha blending control register 5 (Graphics 3)	GR3_AB5	32	H'FCFF77B0	32
	Alpha blending control register 6 (Graphics 3)	GR3_AB6	32	H'FCFF77B4	32
	Alpha blending control register 7 (Graphics 3)	GR3_AB7	32	H'FCFF77B8	32
	Alpha blending control register 8 (Graphics 3)	GR3_AB8	32	H'FCFF77BC	32
	Alpha blending control register 9 (Graphics 3)	GR3_AB9	32	H'FCFF77C0	32
	Alpha blending control register 10 (Graphics 3)	GR3_AB10	32	H'FCFF77C4	32
	Alpha blending control register 11 (Graphics 3)	GR3_AB11	32	H'FCFF77C8	32
	Background color control register (Graphics 3)	GR3_BASE	32	H'FCFF77CC	32
	CLUT table and interrupt control register (Graphics 3)	GR3_CLUT_INT	32	H'FCFF77D0	32
	Status monitor register (Graphics 3)	GR3_MON	32	H'FCFF77D4	32
	VIN synthesizer register update control register	GR_VIN_UPDATE	32	H'FCFF7E00	32
	Alpha blending control register 1 (VIN synthesizer)	GR_VIN_AB1	32	H'FCFF7E20	32
	Register update control register G in gamma correction block	GAM_G_UPDATE	32	H'FCFF7800	32
	Function switch register in gamma correction block	GAM_SW	32	H'FCFF7804	32
	Table setting register G1 in gamma correction block	GAM_G_LUT1	32	H'FCFF7808	32
	Table setting register G2 in gamma correction block	GAM_G_LUT2	32	H'FCFF780C	32
	Table setting register G3 in gamma correction block	GAM_G_LUT3	32	H'FCFF7810	32
	Table setting register G4 in gamma correction block	GAM_G_LUT4	32	H'FCFF7814	32
	Table setting register G5 in gamma correction block	GAM_G_LUT5	32	H'FCFF7818	32
	Table setting register G6 in gamma correction block	GAM_G_LUT6	32	H'FCFF781C	32
	Table setting register G7 in gamma correction block	GAM_G_LUT7	32	H'FCFF7820	32
	Table setting register G8 in gamma correction block	GAM_G_LUT8	32	H'FCFF7824	32
	Table setting register G9 in gamma correction block	GAM_G_LUT9	32	H'FCFF7828	32
	Table setting register G10 in gamma correction block	GAM_G_LUT10	32	H'FCFF782C	32
	Table setting register G11 in gamma correction block	GAM_G_LUT11	32	H'FCFF7830	32
	Table setting register G12 in gamma correction block	GAM_G_LUT12	32	H'FCFF7834	32
	Table setting register G13 in gamma correction block	GAM_G_LUT13	32	H'FCFF7838	32
	Table setting register G14 in gamma correction block	GAM_G_LUT14	32	H'FCFF783C	32
	Table setting register G15 in gamma correction block	GAM_G_LUT15	32	H'FCFF7840	32
	Table setting register G16 in gamma correction block	GAM_G_LUT16	32	H'FCFF7844	32
	Area setting register G1 in gamma correction block	GAM_G_AREA1	32	H'FCFF7848	32
	Area setting register G2 in gamma correction block	GAM_G_AREA2	32	H'FCFF784C	32
	Area setting register G3 in gamma correction block	GAM_G_AREA3	32	H'FCFF7850	32
	Area setting register G4 in gamma correction block	GAM_G_AREA4	32	H'FCFF7854	32
	Area setting register G5 in gamma correction block	GAM_G_AREA5	32	H'FCFF7858	32
	Area setting register G6 in gamma correction block	GAM_G_AREA6	32	H'FCFF785C	32
	Area setting register G7 in gamma correction block	GAM_G_AREA7	32	H'FCFF7860	32
	Area setting register G8 in gamma correction block	GAM_G_AREA8	32	H'FCFF7864	32
	Register update control register B in gamma correction block	GAM_B_UPDATE	32	H'FCFF7880	32
	Table setting register B1 in gamma correction block	GAM_B_LUT1	32	H'FCFF7888	32
	Table setting register B2 in gamma correction block	GAM_B_LUT2	32	H'FCFF788C	32
	Table setting register B3 in gamma correction block	GAM_B_LUT3	32	H'FCFF7890	32
	Table setting register B4 in gamma correction block	GAM_B_LUT4	32	H'FCFF7894	32
	Table setting register B5 in gamma correction block	GAM_B_LUT5	32	H'FCFF7898	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Video display controller 5	Table setting register B6 in gamma correction block	GAM_B_LUT6	32	H'FCFF789C	32
	Table setting register B7 in gamma correction block	GAM_B_LUT7	32	H'FCFF78A0	32
	Table setting register B8 in gamma correction block	GAM_B_LUT8	32	H'FCFF78A4	32
	Table setting register B9 in gamma correction block	GAM_B_LUT9	32	H'FCFF78A8	32
	Table setting register B10 in gamma correction block	GAM_B_LUT10	32	H'FCFF78AC	32
	Table setting register B11 in gamma correction block	GAM_B_LUT11	32	H'FCFF78B0	32
	Table setting register B12 in gamma correction block	GAM_B_LUT12	32	H'FCFF78B4	32
	Table setting register B13 in gamma correction block	GAM_B_LUT13	32	H'FCFF78B8	32
	Table setting register B14 in gamma correction block	GAM_B_LUT14	32	H'FCFF78BC	32
	Table setting register B15 in gamma correction block	GAM_B_LUT15	32	H'FCFF78C0	32
	Table setting register B16 in gamma correction block	GAM_B_LUT16	32	H'FCFF78C4	32
	Area setting register B1 in gamma correction block	GAM_B_AREA1	32	H'FCFF78C8	32
	Area setting register B2 in gamma correction block	GAM_B_AREA2	32	H'FCFF78CC	32
	Area setting register B3 in gamma correction block	GAM_B_AREA3	32	H'FCFF78D0	32
	Area setting register B4 in gamma correction block	GAM_B_AREA4	32	H'FCFF78D4	32
	Area setting register B5 in gamma correction block	GAM_B_AREA5	32	H'FCFF78D8	32
	Area setting register B6 in gamma correction block	GAM_B_AREA6	32	H'FCFF78DC	32
	Area setting register B7 in gamma correction block	GAM_B_AREA7	32	H'FCFF78E0	32
	Area setting register B8 in gamma correction block	GAM_B_AREA8	32	H'FCFF78E4	32
	Register update control register R in gamma correction block	GAM_R_UPDATE	32	H'FCFF7900	32
	Table setting register R1 in gamma correction block	GAM_R_LUT1	32	H'FCFF7908	32
	Table setting register R2 in gamma correction block	GAM_R_LUT2	32	H'FCFF790C	32
	Table setting register R3 in gamma correction block	GAM_R_LUT3	32	H'FCFF7910	32
	Table setting register R4 in gamma correction block	GAM_R_LUT4	32	H'FCFF7914	32
	Table setting register R5 in gamma correction block	GAM_R_LUT5	32	H'FCFF7918	32
	Table setting register R6 in gamma correction block	GAM_R_LUT6	32	H'FCFF791C	32
	Table setting register R7 in gamma correction block	GAM_R_LUT7	32	H'FCFF7920	32
	Table setting register R8 in gamma correction block	GAM_R_LUT8	32	H'FCFF7924	32
	Table setting register R9 in gamma correction block	GAM_R_LUT9	32	H'FCFF7928	32
	Table setting register R10 in gamma correction block	GAM_R_LUT10	32	H'FCFF792C	32
	Table setting register R11 in gamma correction block	GAM_R_LUT11	32	H'FCFF7930	32
	Table setting register R12 in gamma correction block	GAM_R_LUT12	32	H'FCFF7934	32
	Table setting register R13 in gamma correction block	GAM_R_LUT13	32	H'FCFF7938	32
	Table setting register R14 in gamma correction block	GAM_R_LUT14	32	H'FCFF793C	32
	Table setting register R15 in gamma correction block	GAM_R_LUT15	32	H'FCFF7940	32
	Table setting register R16 in gamma correction block	GAM_R_LUT16	32	H'FCFF7944	32
	Area setting register R1 in gamma correction block	GAM_R_AREA1	32	H'FCFF7948	32
	Area setting register R2 in gamma correction block	GAM_R_AREA2	32	H'FCFF794C	32
	Area setting register R3 in gamma correction block	GAM_R_AREA3	32	H'FCFF7950	32
	Area setting register R4 in gamma correction block	GAM_R_AREA4	32	H'FCFF7954	32
	Area setting register R5 in gamma correction block	GAM_R_AREA5	32	H'FCFF7958	32
	Area setting register R6 in gamma correction block	GAM_R_AREA6	32	H'FCFF795C	32
	Area setting register R7 in gamma correction block	GAM_R_AREA7	32	H'FCFF7960	32
	Area setting register R8 in gamma correction block	GAM_R_AREA8	32	H'FCFF7964	32
	TCON register update control register	TCON_UPDATE	32	H'FCFF7980	32
	TCON reference timing setting register	TCON_TIM	32	H'FCFF7984	32
	TCON vertical timing setting register A1	TCON_TIM_STVA1	32	H'FCFF7988	32
	TCON vertical timing setting register A2	TCON_TIM_STVA2	32	H'FCFF798C	32
	TCON vertical timing setting register B1	TCON_TIM_STVB1	32	H'FCFF7990	32
	TCON vertical timing setting register B2	TCON_TIM_STVB2	32	H'FCFF7994	32
	TCON horizontal timing setting register STH1	TCON_TIM_STH1	32	H'FCFF7998	32
	TCON horizontal timing setting register STH2	TCON_TIM_STH2	32	H'FCFF799C	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Video display controller 5	TCON horizontal timing setting register STB1	TCON_TIM_STB1	32	H'FCFF79A0	32
	TCON horizontal timing setting register STB2	TCON_TIM_STB2	32	H'FCFF79A4	32
	TCON horizontal timing setting register CPV1	TCON_TIM_CPV1	32	H'FCFF79A8	32
	TCON horizontal timing setting register CPV2	TCON_TIM_CPV2	32	H'FCFF79AC	32
	TCON horizontal timing setting register POLA1	TCON_TIM_POLA1	32	H'FCFF79B0	32
	TCON horizontal timing setting register POLA2	TCON_TIM_POLA2	32	H'FCFF79B4	32
	TCON horizontal timing setting register POLB1	TCON_TIM_POLB1	32	H'FCFF79B8	32
	TCON horizontal timing setting register POLB2	TCON_TIM_POLB2	32	H'FCFF79BC	32
	TCON data enable polarity setting register	TCON_TIM_DE	32	H'FCFF79C0	32
	Register update control register in output controller	OUT_UPDATE	32	H'FCFF7A00	32
	Output interface register	OUT_SET	32	H'FCFF7A04	32
	Brightness (DC) correction register 1	OUT_BRIGHT1	32	H'FCFF7A08	32
	Brightness (DC) correction register 2	OUT_BRIGHT2	32	H'FCFF7A0C	32
	Contrast (gain) correction register	OUT_CONTRAST	32	H'FCFF7A10	32
	Panel dither register	OUT_PDTHA	32	H'FCFF7A14	32
	Output phase control register	OUT_CLK_PHASE	32	H'FCFF7A24	32
	Interrupt control register 1	SYSCNT_INT1	32	H'FCFF7A80	32
	Interrupt control register 2	SYSCNT_INT2	32	H'FCFF7A84	32
	Interrupt control register 4	SYSCNT_INT4	32	H'FCFF7A8C	32
	Interrupt control register 5	SYSCNT_INT5	32	H'FCFF7A90	32
	Panel clock control register	SYSCNT_PANEL_CLK	16	H'FCFF7A98	16
	CLUT table read select signal status register	SYSCNT_CLUT	16	H'FCFF7A9A	16
Capture engine unit	CEU Capture start register	CAPSR	32	H'E8210000	32
	CEU Capture control register	CAPCR	32	H'E8210004	32
	CEU Capture interface control register	CAMCR	32	H'E8210008	32
	CEU Capture interface cycle register	CMCYR	32	H'E821000C	32
	CEU Capture interface offset register	CAMOR	32	Plane A: H'E8210010 Plane B: H'E8211010 Mirror: H'E8212010	32
	CEU Capture interface width register	CAPWR	32	Plane A: H'E8210014 Plane B: H'E8211014 Mirror: H'E8212014	32
	CEU Capture interface input format register	CAIFR	32	H'E8210018	32
	CEU register control register	CRCNTR	32	H'E8210028	32
	CEU register forcible control register	CRCMPR	32	H'E821002C	32
	CEU Capture filter control register	CFLCR	32	Plane A: H'E8210030 Plane B: H'E8211030 Mirror: H'E8212030	32
	CEU Capture filter size clip register	CFSZR	32	Plane A: H'E8210034 Plane B: H'E8211034 Mirror: H'E8212034	32
	CEU Capture destination width register	CDWDR	32	Plane A: H'E8210038 Plane B: H'E8211038 Mirror: H'E8212038	32
	CEU Capture data address Y register	CDAYR	32	Plane A: H'E821003C Plane B: H'E821103C Mirror: H'E821203C	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Capture engine unit	CEU Capture data address C register	CDACR	32	Plane A: H'E8210040 Plane B: H'E8211040 Mirror: H'E8212040	32
	CEU Capture data bottom-field address Y register	CDBYR	32	Plane A: H'E8210044 Plane B: H'E8211044 Mirror: H'E8212044	32
	CEU Capture data bottom-field address C register	CDBCR	32	Plane A: H'E8210048 Plane B: H'E8211048 Mirror: H'E8212048	32
	CEU Capture bundle destination size register	CBDSR	32	Plane A: H'E821004C Plane B: H'E821104C Mirror: H'E821204C	32
	CEU Firewall operation control register	CFWCR	32	H'E821005C	32
	CEU Capture low-pass filter control register	CLFCR	32	Plane A: H'E8210060 Plane B: H'E8211060 Mirror: H'E8212060	32
	CEU Capture data output control register	CDOCR	32	Plane A: H'E8210064 Plane B: H'E8211064 Mirror: H'E8212064	32
	CEU Capture event interrupt enable register	CEIER	32	H'E8210070	32
	CEU Capture event flag clear register	CETCR	32	H'E8210074	32
	CEU Capture status register	CSTSR	32	H'E821007C	32
	CEU Capture data size register	CDSSR	32	H'E8210084	32
	CEU Capture data address Y register 2	CDAYR2	32	Plane A: H'E8210090 Plane B: H'E8211090 Mirror: H'E8212090	32
	CEU Capture data address C register 2	CDACR2	32	Plane A: H'E8210094 Plane B: H'E8211094 Mirror: H'E8212094	32
	CEU Capture data bottom-field address Y register 2	CDBYR2	32	Plane A: H'E8210098 Plane B: H'E8211098 Mirror: H'E8212098	32
	CEU Capture data bottom-field address C register 2	CDBCR2	32	Plane A: H'E821009C Plane B: H'E821109C Mirror: H'E821209C	32
SCUX	IPC0_0 Initialization Register	IPCIR_IPC0_0	32	H'E8208000	32
	IPC0_0 Pass Select Register	IPSLR_IPC0_0	32	H'E8208004	32
	IPC0_1 Initialization Register	IPCIR_IPC0_1	32	H'E8208100	32
	IPC0_1 Pass Select Register	IPSLR_IPC0_1	32	H'E8208104	32
	IPC0_2 Initialization Register	IPCIR_IPC0_2	32	H'E8208200	32
	IPC0_2 Pass Select Register	IPSLR_IPC0_2	32	H'E8208204	32
	IPC0_3 Initialization Register	IPCIR_IPC0_3	32	H'E8208300	32
	IPC0_3 Pass Select Register	IPSLR_IPC0_3	32	H'E8208304	32
	OPC0_0 Initialization Register	OPCIR_OPC0_0	32	H'E8208400	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
SCUX	OPC0_0 Pass Select Register	OPSLR_OPC0_0	32	H'E8208404	32
	OPC0_1 Initialization Register	OPCIR_OPC0_1	32	H'E8208500	32
	OPC0_1 Pass Select Register	OPSLR_OPC0_1	32	H'E8208504	32
	OPC0_2 Initialization Register	OPCIR_OPC0_2	32	H'E8208600	32
	OPC0_2 Pass Select Register	OPSLR_OPC0_2	32	H'E8208604	32
	OPC0_3 Initialization Register	OPCIR_OPC0_3	32	H'E8208700	32
	OPC0_3 Pass Select Register	OPSLR_OPC0_3	32	H'E8208704	32
	FFD0_0 FIFO Download Initialization Register	FFDIR_FFD0_0	32	H'E8208800	32
	FFD0_0 FIFO Download Audio Information Register	FDAIR_FFD0_0	32	H'E8208804	32
	FFD0_0 FIFO Download Request Size Register	DRQSR_FFD0_0	32	H'E8208808	32
	FFD0_0 FIFO Download Pass Register	FFDPR_FFD0_0	32	H'E820880C	32
	FFD0_0 FIFO Download Boot Register	FFDBR_FFD0_0	32	H'E8208810	32
	FFD0_0 FIFO Download Event Mask Register	DEVMR_FFD0_0	32	H'E8208814	32
	FFD0_0 FIFO Download Event Clear Register	DEVCR_FFD0_0	32	H'E820881C	32
	FFD0_1 FIFO Download Initialization Register	FFDIR_FFD0_1	32	H'E8208900	32
	FFD0_1 FIFO Download Audio Information Register	FDAIR_FFD0_1	32	H'E8208904	32
	FFD0_1 FIFO Download Request Size Register	DRQSR_FFD0_1	32	H'E8208908	32
	FFD0_1 FIFO Download Pass Register	FFDPR_FFD0_1	32	H'E820890C	32
	FFD0_1 FIFO Download Boot Register	FFDBR_FFD0_1	32	H'E8208910	32
	FFD0_1 FIFO Download Event Mask Register	DEVMR_FFD0_1	32	H'E8208914	32
	FFD0_1 FIFO Download Event Clear Register	DEVCR_FFD0_1	32	H'E820891C	32
	FFD0_2 FIFO Download Initialization Register	FFDIR_FFD0_2	32	H'E8208A00	32
	FFD0_2 FIFO Download Audio Information Register	FDAIR_FFD0_2	32	H'E8208A04	32
	FFD0_2 FIFO Download Request Size Register	DRQSR_FFD0_2	32	H'E8208A08	32
	FFD0_2 FIFO Download Pass Register	FFDPR_FFD0_2	32	H'E8208A0C	32
	FFD0_2 FIFO Download Boot Register	FFDBR_FFD0_2	32	H'E8208A10	32
	FFD0_2 FIFO Download Event Mask Register	DEVMR_FFD0_2	32	H'E8208A14	32
	FFD0_2 FIFO Download Event Clear Register	DEVCR_FFD0_2	32	H'E8208A1C	32
	FFD0_3 FIFO Download Initialization Register	FFDIR_FFD0_3	32	H'E8208B00	32
	FFD0_3 FIFO Download Audio Information Register	FDAIR_FFD0_3	32	H'E8208B04	32
	FFD0_3 FIFO Download Request Size Register	DRQSR_FFD0_3	32	H'E8208B08	32
	FFD0_3 FIFO Download Pass Register	FFDPR_FFD0_3	32	H'E8208B0C	32
	FFD0_3 FIFO Download Boot Register	FFDBR_FFD0_3	32	H'E8208B10	32
	FFD0_3 FIFO Download Event Mask Register	DEVMR_FFD0_3	32	H'E8208B14	32
	FFD0_3 FIFO Download Event Clear Register	DEVCR_FFD0_3	32	H'E8208B1C	32
	FFU0_0 FIFO Upload Initialization Register	FFUIR_FFU0_0	32	H'E8208C00	32
	FFU0_0 FIFO Upload Audio Information Register	FUAIR_FFU0_0	32	H'E8208C04	32
	FFU0_0 FIFO Upload Request Size Register	URQSR_FFU0_0	32	H'E8208C08	32
	FFU0_0 FIFO Upload Pass Register	FFUPR_FFU0_0	32	H'E8208C0C	32
	FFU0_0 FIFO Upload Event Mask Register	UEVMR_FFU0_0	32	H'E8208C10	32
	FFU0_0 FIFO Upload Event Clear Register	UEVCR_FFU0_0	32	H'E8208C18	32
	FFU0_1 FIFO Upload Initialization Register	FFUIR_FFU0_1	32	H'E8208D00	32
	FFU0_1 FIFO Upload Audio Information Register	FUAIR_FFU0_1	32	H'E8208D04	32
	FFU0_1 FIFO Upload Request Size Register	URQSR_FFU0_1	32	H'E8208D08	32
	FFU0_1 FIFO Upload Pass Register	FFUPR_FFU0_1	32	H'E8208D0C	32
	FFU0_1 FIFO Upload Event Mask Register	UEVMR_FFU0_1	32	H'E8208D10	32
	FFU0_1 FIFO Upload Event Clear Register	UEVCR_FFU0_1	32	H'E8208D18	32
	FFU0_2 FIFO Upload Initialization Register	FFUIR_FFU0_2	32	H'E8208E00	32
	FFU0_2 FIFO Upload Audio Information Register	FUAIR_FFU0_2	32	H'E8208E04	32
	FFU0_2 FIFO Upload Request Size Register	URQSR_FFU0_2	32	H'E8208E08	32
	FFU0_2 FIFO Upload Pass Register	FFUPR_FFU0_2	32	H'E8208E0C	32
	FFU0_2 FIFO Upload Event Mask Register	UEVMR_FFU0_2	32	H'E8208E10	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
SCUX	FFU0_2 FIFO Upload Event Clear Register	UEVCR_FFU0_2	32	H'E8208E18	32
	FFU0_3 FIFO Upload Initialization Register	FFUIR_FFU0_3	32	H'E8208F00	32
	FFU0_3 FIFO Upload Audio Information Register	FUAIR_FFU0_3	32	H'E8208F04	32
	FFU0_3 FIFO Upload Request Size Register	URQSR_FFU0_3	32	H'E8208F08	32
	FFU0_3 FIFO Upload Pass Register	FFUPR_FFU0_3	32	H'E8208F0C	32
	FFU0_3 FIFO Upload Event Mask Register	UEVMR_FFU0_3	32	H'E8208F10	32
	FFU0_3 FIFO Upload Event Clear Register	UEVCR_FFU0_3	32	H'E8208F18	32
	2SRC0_0 Initialization Register 0	SRCIR0_2SRC0_0	32	H'E8209000	32
	2SRC0_0 Audio Information Register 0	SADIR0_2SRC0_0	32	H'E8209004	32
	2SRC0_0 Bypass Register 0	SRCBR0_2SRC0_0	32	H'E8209008	32
	2SRC0_0 IFS Control Register 0	IFSCR0_2SRC0_0	32	H'E820900C	32
	2SRC0_0 IFS Value Setting Register 0	IFSVR0_2SRC0_0	32	H'E8209010	32
	2SRC0_0 Control Register 0	SRCCR0_2SRC0_0	32	H'E8209014	32
	2SRC0_0 Minimum FS Setting Register 0	MNFSR0_2SRC0_0	32	H'E8209018	32
	2SRC0_0 Buffer Size Setting Register 0	BFSSR0_2SRC0_0	32	H'E820901C	32
	2SRC0_0 SCU2 Status Register 0	SC2SR0_2SRC0_0	32	H'E8209020	32
	2SRC0_0 Wait Time Setting Register 0	WATSR0_2SRC0_0	32	H'E8209024	32
	2SRC0_0 Event Mask Register 0	SEVMR0_2SRC0_0	32	H'E8209028	32
	2SRC0_0 Event Clear Register 0	SEVCR0_2SRC0_0	32	H'E8209030	32
	2SRC0_0 Initialization Register 1	SRCIR1_2SRC0_0	32	H'E8209034	32
	2SRC0_0 Audio Information Register 1	SADIR1_2SRC0_0	32	H'E8209038	32
	2SRC0_0 Bypass Register 1	SRCBR1_2SRC0_0	32	H'E820903C	32
	2SRC0_0 IFS Control Register 1	IFSCR1_2SRC0_0	32	H'E8209040	32
	2SRC0_0 IFS Value Setting Register 1	IFSVR1_2SRC0_0	32	H'E8209044	32
	2SRC0_0 Control Register 1	SRCCR1_2SRC0_0	32	H'E8209048	32
	2SRC0_0 Minimum FS Setting Register 1	MNFSR1_2SRC0_0	32	H'E820904C	32
	2SRC0_0 Buffer Size Setting Register 1	BFSSR1_2SRC0_0	32	H'E8209050	32
	2SRC0_0 SCU2 Status Register 1	SC2SR1_2SRC0_0	32	H'E8209054	32
	2SRC0_0 Wait Time Setting Register 1	WATSR1_2SRC0_0	32	H'E8209058	32
	2SRC0_0 Event Mask Register 1	SEVMR1_2SRC0_0	32	H'E820905C	32
	2SRC0_0 Event Clear Register 1	SEVCR1_2SRC0_0	32	H'E8209064	32
	2SRC0_0 Initialization Register RIF	SRCIRR_2SRC0_0	32	H'E8209068	32
	2SRC0_1 Initialization Register 0	SRCIR0_2SRC0_1	32	H'E8209100	32
	2SRC0_1 Audio Information Register 0	SADIR0_2SRC0_1	32	H'E8209104	32
	2SRC0_1 Bypass Register 0	SRCBR0_2SRC0_1	32	H'E8209108	32
	2SRC0_1 IFS Control Register 0	IFSCR0_2SRC0_1	32	H'E820910C	32
	2SRC0_1 IFS Value Setting Register 0	IFSVR0_2SRC0_1	32	H'E8209110	32
	2SRC0_1 Control Register 0	SRCCR0_2SRC0_1	32	H'E8209114	32
	2SRC0_1 Minimum FS Setting Register 0	MNFSR0_2SRC0_1	32	H'E8209118	32
	2SRC0_1 Buffer Size Setting Register 0	BFSSR0_2SRC0_1	32	H'E820911C	32
	2SRC0_1 SCU2 Status Register 0	SC2SR0_2SRC0_1	32	H'E8209120	32
	2SRC0_1 Wait Time Setting Register 0	WATSR0_2SRC0_1	32	H'E8209124	32
	2SRC0_1 Event Mask Register 0	SEVMR0_2SRC0_1	32	H'E8209128	32
	2SRC0_1 Event Clear Register 0	SEVCR0_2SRC0_1	32	H'E8209130	32
	2SRC0_1 Initialization Register 1	SRCIR1_2SRC0_1	32	H'E8209134	32
	2SRC0_1 Audio Information Register 1	SADIR1_2SRC0_1	32	H'E8209138	32
	2SRC0_1 Bypass Register 1	SRCBR1_2SRC0_1	32	H'E820913C	32
	2SRC0_1 IFS Control Register 1	IFSCR1_2SRC0_1	32	H'E8209140	32
	2SRC0_1 IFS Value Setting Register 1	IFSVR1_2SRC0_1	32	H'E8209144	32
	2SRC0_1 Control Register 1	SRCCR1_2SRC0_1	32	H'E8209148	32
	2SRC0_1 Minimum FS Setting Register 1	MNFSR1_2SRC0_1	32	H'E820914C	32
	2SRC0_1 Buffer Size Setting Register 1	BFSSR1_2SRC0_1	32	H'E8209150	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
SCUX	2SRC0_1 SCU2 Status Register 1	SC2SR1_2SRC0_1	32	H'E8209154	32
	2SRC0_1 Wait Time Setting Register 1	WATSR1_2SRC0_1	32	H'E8209158	32
	2SRC0_1 Event Mask Register 1	SEVMR1_2SRC0_1	32	H'E820915C	32
	2SRC0_1 Event Clear Register 1	SEVCR1_2SRC0_1	32	H'E8209164	32
	2SRC0_1 Initialization Register RIF	SRCIRR_2SRC0_1	32	H'E8209168	32
	DVU0_0 Initialization Register	DVUIR_DVU0_0	32	H'E8209200	32
	DVU0_0 Audio Information Register	VADIR_DVU0_0	32	H'E8209204	32
	DVU0_0 Bypass Register	DVUBR_DVU0_0	32	H'E8209208	32
	DVU0_0 Control Register	DVUCR_DVU0_0	32	H'E820920C	32
	DVU0_0 Zero Cross Mute Control Register	ZCMCR_DVU0_0	32	H'E8209210	32
	DVU0_0 Volume Ramp Control Register	VRCTR_DVU0_0	32	H'E8209214	32
	DVU0_0 Volume Ramp Period Register	VRPDR_DVU0_0	32	H'E8209218	32
	DVU0_0 Volume Ramp Decibel Register	VRDBR_DVU0_0	32	H'E820921C	32
	DVU0_0 Volume Ramp Wait Time Register	VRWTR_DVU0_0	32	H'E8209220	32
	DVU0_0 Volume Value Setting 0 Register	VOL0R_DVU0_0	32	H'E8209224	32
	DVU0_0 Volume Value Setting 1 Register	VOL1R_DVU0_0	32	H'E8209228	32
	DVU0_0 Volume Value Setting 2 Register	VOL2R_DVU0_0	32	H'E820922C	32
	DVU0_0 Volume Value Setting 3 Register	VOL3R_DVU0_0	32	H'E8209230	32
	DVU0_0 Volume Value Setting 4 Register	VOL4R_DVU0_0	32	H'E8209234	32
	DVU0_0 Volume Value Setting 5 Register	VOL5R_DVU0_0	32	H'E8209238	32
	DVU0_0 Volume Value Setting 6 Register	VOL6R_DVU0_0	32	H'E820923C	32
	DVU0_0 Volume Value Setting 7 Register	VOL7R_DVU0_0	32	H'E8209240	32
	DVU0_0 Enable Register	DVUER_DVU0_0	32	H'E8209244	32
	DVU0_0 Status Register	DVUSR_DVU0_0	32	H'E8209248	32
	DVU0_0 Event Mask Register	VEVMR_DVU0_0	32	H'E820924C	32
	DVU0_0 Event Clear Register	VEVCR_DVU0_0	32	H'E8209254	32
	DVU0_1 Initialization Register	DVUIR_DVU0_1	32	H'E8209300	32
	DVU0_1 Audio Information Register	VADIR_DVU0_1	32	H'E8209304	32
	DVU0_1 Bypass Register	DVUBR_DVU0_1	32	H'E8209308	32
	DVU0_1 Control Register	DVUCR_DVU0_1	32	H'E820930C	32
	DVU0_1 Zero Cross Mute Control Register	ZCMCR_DVU0_1	32	H'E8209310	32
	DVU0_1 Volume Ramp Control Register	VRCTR_DVU0_1	32	H'E8209314	32
	DVU0_1 Volume Ramp Period Register	VRPDR_DVU0_1	32	H'E8209318	32
	DVU0_1 Volume Ramp Decibel Register	VRDBR_DVU0_1	32	H'E820931C	32
	DVU0_1 Volume Ramp Wait Time Register	VRWTR_DVU0_1	32	H'E8209320	32
	DVU0_1 Volume Value Setting 0 Register	VOL0R_DVU0_1	32	H'E8209324	32
	DVU0_1 Volume Value Setting 1 Register	VOL1R_DVU0_1	32	H'E8209328	32
	DVU0_1 Volume Value Setting 2 Register	VOL2R_DVU0_1	32	H'E820932C	32
	DVU0_1 Volume Value Setting 3 Register	VOL3R_DVU0_1	32	H'E8209330	32
	DVU0_1 Volume Value Setting 4 Register	VOL4R_DVU0_1	32	H'E8209334	32
	DVU0_1 Volume Value Setting 5 Register	VOL5R_DVU0_1	32	H'E8209338	32
	DVU0_1 Volume Value Setting 6 Register	VOL6R_DVU0_1	32	H'E820933C	32
	DVU0_1 Volume Value Setting 7 Register	VOL7R_DVU0_1	32	H'E8209340	32
	DVU0_1 Enable Register	DVUER_DVU0_1	32	H'E8209344	32
	DVU0_1 Status Register	DVUSR_DVU0_1	32	H'E8209348	32
	DVU0_1 Event Mask Register	VEVMR_DVU0_1	32	H'E820934C	32
	DVU0_1 Event Clear Register	VEVCR_DVU0_1	32	H'E8209354	32
	DVU0_2 Initialization Register	DVUIR_DVU0_2	32	H'E8209400	32
	DVU0_2 Audio Information Register	VADIR_DVU0_2	32	H'E8209404	32
	DVU0_2 Bypass Register	DVUBR_DVU0_2	32	H'E8209408	32
	DVU0_2 Control Register	DVUCR_DVU0_2	32	H'E820940C	32
	DVU0_2 Zero Cross Mute Control Register	ZCMCR_DVU0_2	32	H'E8209410	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
SCUX	DVU0_2 Volume Ramp Control Register	VRCTR_DVU0_2	32	H'E8209414	32
	DVU0_2 Volume Ramp Period Register	VRPDR_DVU0_2	32	H'E8209418	32
	DVU0_2 Volume Ramp Decibel Register	VRDBR_DVU0_2	32	H'E820941C	32
	DVU0_2 Volume Ramp Wait Time Register	VRWTR_DVU0_2	32	H'E8209420	32
	DVU0_2 Volume Value Setting 0 Register	VOL0R_DVU0_2	32	H'E8209424	32
	DVU0_2 Volume Value Setting 1 Register	VOL1R_DVU0_2	32	H'E8209428	32
	DVU0_2 Volume Value Setting 2 Register	VOL2R_DVU0_2	32	H'E820942C	32
	DVU0_2 Volume Value Setting 3 Register	VOL3R_DVU0_2	32	H'E8209430	32
	DVU0_2 Volume Value Setting 4 Register	VOL4R_DVU0_2	32	H'E8209434	32
	DVU0_2 Volume Value Setting 5 Register	VOL5R_DVU0_2	32	H'E8209438	32
	DVU0_2 Volume Value Setting 6 Register	VOL6R_DVU0_2	32	H'E820943C	32
	DVU0_2 Volume Value Setting 7 Register	VOL7R_DVU0_2	32	H'E8209440	32
	DVU0_2 Enable Register	DVUER_DVU0_2	32	H'E8209444	32
	DVU0_2 Status Register	DVUSR_DVU0_2	32	H'E8209448	32
	DVU0_2 Event Mask Register	VEVMR_DVU0_2	32	H'E820944C	32
	DVU0_2 Event Clear Register	VEVCR_DVU0_2	32	H'E8209454	32
	DVU0_3 Initialization Register	DVUIR_DVU0_3	32	H'E8209500	32
	DVU0_3 Audio Information Register	VADIR_DVU0_3	32	H'E8209504	32
	DVU0_3 Bypass Register	DVUBR_DVU0_3	32	H'E8209508	32
	DVU0_3 Control Register	DVUCR_DVU0_3	32	H'E820950C	32
	DVU0_3 Zero Cross Mute Control Register	ZCMCR_DVU0_3	32	H'E8209510	32
	DVU0_3 Volume Ramp Control Register	VRCTR_DVU0_3	32	H'E8209514	32
	DVU0_3 Volume Ramp Period Register	VRPDR_DVU0_3	32	H'E8209518	32
	DVU0_3 Volume Ramp Decibel Register	VRDBR_DVU0_3	32	H'E820951C	32
	DVU0_3 Volume Ramp Wait Time Register	VRWTR_DVU0_3	32	H'E8209520	32
	DVU0_3 Volume Value Setting 0 Register	VOL0R_DVU0_3	32	H'E8209524	32
	DVU0_3 Volume Value Setting 1 Register	VOL1R_DVU0_3	32	H'E8209528	32
	DVU0_3 Volume Value Setting 2 Register	VOL2R_DVU0_3	32	H'E820952C	32
	DVU0_3 Volume Value Setting 3 Register	VOL3R_DVU0_3	32	H'E8209530	32
	DVU0_3 Volume Value Setting 4 Register	VOL4R_DVU0_3	32	H'E8209534	32
	DVU0_3 Volume Value Setting 5 Register	VOL5R_DVU0_3	32	H'E8209538	32
	DVU0_3 Volume Value Setting 6 Register	VOL6R_DVU0_3	32	H'E820953C	32
	DVU0_3 Volume Value Setting 7 Register	VOL7R_DVU0_3	32	H'E8209540	32
	DVU0_3 Enable Register	DVUER_DVU0_3	32	H'E8209544	32
	DVU0_3 Status Register	DVUSR_DVU0_3	32	H'E8209548	32
	DVU0_3 Event Mask Register	VEVMR_DVU0_3	32	H'E820954C	32
	DVU0_3 Event Clear Register	VEVCR_DVU0_3	32	H'E8209554	32
	MIX0_0 Initialization Register	MIXIR_MIX0_0	32	H'E8209600	32
	MIX0_0 Audio Information Register	MADIR_MIX0_0	32	H'E8209604	32
	MIX0_0 Bypass Register	MIXBR_MIX0_0	32	H'E8209608	32
	MIX0_0 Mode Register	MIXMR_MIX0_0	32	H'E820960C	32
	MIX0_0 Volume Period Register	MVPDR_MIX0_0	32	H'E8209610	32
	MIX0_0 Decibel A Register	MDBAR_MIX0_0	32	H'E8209614	32
	MIX0_0 Decibel B Register	MDBBR_MIX0_0	32	H'E8209618	32
	MIX0_0 Decibel C Register	MDBCR_MIX0_0	32	H'E820961C	32
	MIX0_0 Decibel D Register	MDBDR_MIX0_0	32	H'E8209620	32
	MIX0_0 Decibel Enable Register	MDBER_MIX0_0	32	H'E8209624	32
	MIX0_0 Status Register	MIXSR_MIX0_0	32	H'E8209628	32
	Software Reset Register	SWRSR_CIM	32	H'E8209700	32
	DMA Control Register	DMACR_CIM	32	H'E8209704	32
	DMA Transfer Register for FFD0_0 RAM	DMATD0_CIM	32	H'E8209708	16, 32
	DMA Transfer Register for FFD0_1 RAM	DMATD1_CIM	32	H'E820970C	16, 32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
SCUX	DMA Transfer Register for FFD0_2 RAM	DMATD2_CIM	32	H'E8209710	16, 32
	DMA Transfer Register for FFD0_3 RAM	DMATD3_CIM	32	H'E8209714	16, 32
	DMA Transfer Register for FFU0_0 RAM	DMATU0_CIM	32	H'E8209718	16, 32
	DMA Transfer Register for FFU0_1 RAM	DMATU1_CIM	32	H'E820971C	16, 32
	DMA Transfer Register for FFU0_2 RAM	DMATU2_CIM	32	H'E8209720	16, 32
	DMA Transfer Register for FFU0_3 RAM	DMATU3_CIM	32	H'E8209724	16, 32
	SSI route select register	SSIRSEL_CIM	32	H'E8209738	32
	FFD0_0 timing select register	FDTSEL0_CIM	32	H'E820973C	32
	FFD0_1 timing select register	FDTSEL1_CIM	32	H'E8209740	32
	FFD0_2 timing select register	FDTSEL2_CIM	32	H'E8209744	32
	FFD0_3 timing select register	FDTSEL3_CIM	32	H'E8209748	32
	FFU0_0 timing select register	FUTSEL0_CIM	32	H'E820974C	32
	FFU0_1 timing select register	FUTSEL1_CIM	32	H'E8209750	32
	FFU0_2 timing select register	FUTSEL2_CIM	32	H'E8209754	32
	FFU0_3 timing select register	FUTSEL3_CIM	32	H'E8209758	32
	SSI pin mode register	SSIPMD_CIM	32	H'E820975C	32
	SSI control register	SSICTRL_CIM	32	H'E8209760	32
	SRC0 route select register	SRCRSEL0_CIM	32	H'E8209764	32
	SRC1 route select register	SRCRSEL1_CIM	32	H'E8209768	32
	SRC2 route select register	SRCRSEL2_CIM	32	H'E820976C	32
	SRC3 route select register	SRCRSEL3_CIM	32	H'E8209770	32
	MIX route select register	MIXRSEL_CIM	32	H'E8209774	32
SD host interface	Command type register_0	SD_CMD_0	16	H'E804E000	16
	SD command argument register 0_0	SD_ARG0_0	16	H'E804E004	16
	SD command argument register 1_0	SD_ARG1_0	16	H'E804E006	16
	Data stop register_0	SD_STOP_0	16	H'E804E008	16
	Block count register_0	SD_SECCNT_0	16	H'E804E00A	16
	SD card response register 00_0	SD_RSP00_0	16	H'E804E00C	16
	SD card response register 01_0	SD_RSP01_0	16	H'E804E00E	16
	SD card response register 02_0	SD_RSP02_0	16	H'E804E010	16
	SD card response register 03_0	SD_RSP03_0	16	H'E804E012	16
	SD card response register 04_0	SD_RSP04_0	16	H'E804E014	16
	SD card response register 05_0	SD_RSP05_0	16	H'E804E016	16
	SD card response register 06_0	SD_RSP06_0	16	H'E804E018	16
	SD card response register 07_0	SD_RSP07_0	16	H'E804E01A	16
	SD card interrupt flag register 1_0	SD_INFO1_0	16	H'E804E01C	16
	SD card interrupt flag register 2_0	SD_INFO2_0	16	H'E804E01E	16
	SD_INFO1 interrupt mask register_0	SD_INFO1_MASK_0	16	H'E804E020	16
	SD_INFO2 interrupt mask register_0	SD_INFO2_MASK_0	16	H'E804E022	16
	SD clock control register_0	SD_CLK_CTRL_0	16	H'E804E024	16
	Transfer data length register_0	SD_SIZE_0	16	H'E804E026	16
	SD card access control option register_0	SD_OPTION_0	16	H'E804E028	16
	SD error status register 1_0	SD_ERR_STS1_0	16	H'E804E02C	16
	SD error status register 2_0	SD_ERR_STS2_0	16	H'E804E02E	16
	SD buffer read/write register_0	SD_BUF0_0	32	H'E804E030	32
	SDIO mode control register_0	SDIO_MODE_0	16	H'E804E034	16
	SDIO interrupt flag register_0	SDIO_INFO1_0	16	H'E804E036	16
	SDIO_INFO1 interrupt mask register_0	SDIO_INFO1_MASK_0	16	H'E804E038	16
	DMA mode enable register_0	CC_EXT_MODE_0	16	H'E804E0D8	16
	Software reset register_0	SOFT_RST_0	16	H'E804E0E0	16
	Version register_0	VERSION_0	16	H'E804E0E2	16
	Swap control register_0	EXT_SWAP_0	16	H'E804E0F0	16

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
SD host interface	Command type register_1	SD_CMD_1	16	H'E804E800	16
	SD command argument register 0_1	SD_ARG0_1	16	H'E804E804	16
	SD command argument register 1_1	SD_ARG1_1	16	H'E804E806	16
	Data stop register_1	SD_STOP_1	16	H'E804E808	16
	Block count register_1	SD_SECCNT_1	16	H'E804E80A	16
	SD card response register 00_1	SD_RSP00_1	16	H'E804E80C	16
	SD card response register 01_1	SD_RSP01_1	16	H'E804E80E	16
	SD card response register 02_1	SD_RSP02_1	16	H'E804E810	16
	SD card response register 03_1	SD_RSP03_1	16	H'E804E812	16
	SD card response register 04_1	SD_RSP04_1	16	H'E804E814	16
	SD card response register 05_1	SD_RSP05_1	16	H'E804E816	16
	SD card response register 06_1	SD_RSP06_1	16	H'E804E818	16
	SD card response register 07_1	SD_RSP07_1	16	H'E804E81A	16
	SD card interrupt flag register 1_1	SD_INFO1_1	16	H'E804E81C	16
	SD card interrupt flag register 2_1	SD_INFO2_1	16	H'E804E81E	16
	SD_INFO1 interrupt mask register_1	SD_INFO1_MASK_1	16	H'E804E820	16
	SD_INFO2 interrupt mask register_1	SD_INFO2_MASK_1	16	H'E804E822	16
	SD clock control register_1	SD_CLK_CTRL_1	16	H'E804E824	16
	Transfer data length register_1	SD_SIZE_1	16	H'E804E826	16
	SD card access control option register_1	SD_OPTION_1	16	H'E804E828	16
	SD error status register 1_1	SD_ERR_STS1_1	16	H'E804E82C	16
	SD error status register 2_1	SD_ERR_STS2_1	16	H'E804E82E	16
	SD buffer read/write register_1	SD_BUF0_1	32	H'E804E830	32
	SDIO mode control register_1	SDIO_MODE_1	16	H'E804E834	16
	SDIO interrupt flag register_1	SDIO_INFO1_1	16	H'E804E836	16
	SDIO_INFO1 interrupt mask register_1	SDIO_INFO1_MASK_1	16	H'E804E838	16
	DMA mode enable register_1	CC_EXT_MODE_1	16	H'E804E8D8	16
	Software reset register_1	SOFT_RST_1	16	H'E804E8E0	16
	Version register_1	VERSION_1	16	H'E804E8E2	16
	Swap control register_1	EXT_SWAP_1	16	H'E804E8F0	16
MMC host interface	Command setting register	CE_CMD_SET	32	H'E804C800	16
				H'E804C802	16
	Argument register	CE_ARG	32	H'E804C808	32
	Argument register for automatically-issued CMD12	CE_ARG_CMD12	32	H'E804C80C	32
	Command control register	CE_CMD_CTRL	32	H'E804C810	32
	Transfer block setting register	CE_BLOCK_SET	32	H'E804C814	32
	Clock control register	CE_CLK_CTRL	32	H'E804C818	32
	Buffer access configuration register	CE_BUF_ACC	32	H'E804C81C	32
	Response register 3	CE_RESP3	32	H'E804C820	32
	Response register 2	CE_RESP2	32	H'E804C824	32
	Response register 1	CE_RESP1	32	H'E804C828	32
	Response register 0	CE_RESP0	32	H'E804C82C	32
	Response register for automatically-issued CMD12	CE_RESP_CMD12	32	H'E804C830	32
	Data register	CE_DATA	32	H'E804C834	32
	Interrupt flag register	CE_INT	32	H'E804C840	32
	Interrupt enable register	CE_INT_EN	32	H'E804C844	32
	Status register 1	CE_HOST_STS1	32	H'E804C848	32
	Status register 2	CE_HOST_STS2	32	H'E804C84C	32
	DMA mode setting register	CE_DMA_MODE	32	H'E804C85C	32
	Card detection/port control register	CE_DETECT	32	H'E804C870	32
	Special mode setting register	CE_ADD_MODE	32	H'E804C874	32
	Version register	CE_VERSION	32	H'E804C87C	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Ports	Port register_1	P1	16	H'FCFE3004	16
	Port register_2	P2	16	H'FCFE3008	16
	Port register_3	P3	16	H'FCFE300C	16
	Port register_4	P4	16	H'FCFE3010	16
	Port register_5	P5	16	H'FCFE3014	16
	Port register_6	P6	16	H'FCFE3018	16
	Port register_7	P7	16	H'FCFE301C	16
	Port register_8	P8	16	H'FCFE3020	16
	Port register_9	P9	16	H'FCFE3024	16
	Port set/reset register_1	PSR1	32	H'FCFE3104	32
	Port set/reset register_2	PSR2	32	H'FCFE3108	32
	Port set/reset register_3	PSR3	32	H'FCFE310C	32
	Port set/reset register_4	PSR4	32	H'FCFE3110	32
	Port set/reset register_5	PSR5	32	H'FCFE3114	32
	Port set/reset register_6	PSR6	32	H'FCFE3118	32
	Port set/reset register_7	PSR7	32	H'FCFE311C	32
	Port set/reset register_8	PSR8	32	H'FCFE3120	32
	Port set/reset register_9	PSR9	32	H'FCFE3124	32
	Port pin read register_0	PPR0	16	H'FCFE3200	16
	Port pin read register_1	PPR1	16	H'FCFE3204	16
	Port pin read register_2	PPR2	16	H'FCFE3208	16
	Port pin read register_3	PPR3	16	H'FCFE320C	16
	Port pin read register_4	PPR4	16	H'FCFE3210	16
	Port pin read register_5	PPR5	16	H'FCFE3214	16
	Port pin read register_6	PPR6	16	H'FCFE3218	16
	Port pin read register_7	PPR7	16	H'FCFE321C	16
	Port pin read register_8	PPR8	16	H'FCFE3220	16
	Port pin read register_9	PPR9	16	H'FCFE3224	16
	Port mode register_1	PM1	16	H'FCFE3304	16
	Port mode register_2	PM2	16	H'FCFE3308	16
	Port mode register_3	PM3	16	H'FCFE330C	16
	Port mode register_4	PM4	16	H'FCFE3310	16
	Port mode register_5	PM5	16	H'FCFE3314	16
	Port mode register_6	PM6	16	H'FCFE3318	16
	Port mode register_7	PM7	16	H'FCFE331C	16
	Port mode register_8	PM8	16	H'FCFE3320	16
	Port mode register_9	PM9	16	H'FCFE3324	16
	Port mode control register_0	PMC0	16	H'FCFE3400	16
	Port mode control register_1	PMC1	16	H'FCFE3404	16
	Port mode control register_2	PMC2	16	H'FCFE3408	16
	Port mode control register_3	PMC3	16	H'FCFE340C	16
	Port mode control register_4	PMC4	16	H'FCFE3410	16
	Port mode control register_5	PMC5	16	H'FCFE3414	16
	Port mode control register_6	PMC6	16	H'FCFE3418	16
	Port mode control register_7	PMC7	16	H'FCFE341C	16
	Port mode control register_8	PMC8	16	H'FCFE3420	16
	Port mode control register_9	PMC9	16	H'FCFE3424	16
	Port function control register_0	PFC0	16	H'FCFE3500	16
	Port function control register_1	PFC1	16	H'FCFE3504	16
	Port function control register_2	PFC2	16	H'FCFE3508	16

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Ports	Port function control register_3	PFC3	16	H'FCFE350C	16
	Port function control register_4	PFC4	16	H'FCFE3510	16
	Port function control register_5	PFC5	16	H'FCFE3514	16
	Port function control register_6	PFC6	16	H'FCFE3518	16
	Port function control register_7	PFC7	16	H'FCFE351C	16
	Port function control register_8	PFC8	16	H'FCFE3520	16
	Port function control register_9	PFC9	16	H'FCFE3524	16
	Port function control expansion register_0	PFCE0	16	H'FCFE3600	16
	Port function control expansion register_1	PFCE1	16	H'FCFE3604	16
	Port function control expansion register_2	PFCE2	16	H'FCFE3608	16
	Port function control expansion register_3	PFCE3	16	H'FCFE360C	16
	Port function control expansion register_4	PFCE4	16	H'FCFE3610	16
	Port function control expansion register_5	PFCE5	16	H'FCFE3614	16
	Port function control expansion register_6	PFCE6	16	H'FCFE3618	16
	Port function control expansion register_7	PFCE7	16	H'FCFE361C	16
	Port function control expansion register_8	PFCE8	16	H'FCFE3620	16
	Port function control expansion register_9	PFCE9	16	H'FCFE3624	16
	Port NOT register_1	PNOT1	16	H'FCFE3704	16
	Port NOT register_2	PNOT2	16	H'FCFE3708	16
	Port NOT register_3	PNOT3	16	H'FCFE370C	16
	Port NOT register_4	PNOT4	16	H'FCFE3710	16
	Port NOT register_5	PNOT5	16	H'FCFE3714	16
	Port NOT register_6	PNOT6	16	H'FCFE3718	16
	Port NOT register_7	PNOT7	16	H'FCFE371C	16
	Port NOT register_8	PNOT8	16	H'FCFE3720	16
	Port NOT register_9	PNOT9	16	H'FCFE3724	16
	Port mode set/reset register_1	PMSR1	32	H'FCFE3804	32
	Port mode set/reset register_2	PMSR2	32	H'FCFE3808	32
	Port mode set/reset register_3	PMSR3	32	H'FCFE380C	32
	Port mode set/reset register_4	PMSR4	32	H'FCFE3810	32
	Port mode set/reset register_5	PMSR5	32	H'FCFE3814	32
	Port mode set/reset register_6	PMSR6	32	H'FCFE3818	32
	Port mode set/reset register_7	PMSR7	32	H'FCFE381C	32
	Port mode set/reset register_8	PMSR8	32	H'FCFE3820	32
	Port mode set/reset register_9	PMSR9	32	H'FCFE3824	32
	Port mode control set/reset register_0	PMCSR0	32	H'FCFE3900	32
	Port mode control set/reset register_1	PMCSR1	32	H'FCFE3904	32
	Port mode control set/reset register_2	PMCSR2	32	H'FCFE3908	32
	Port mode control set/reset register_3	PMCSR3	32	H'FCFE390C	32
	Port mode control set/reset register_4	PMCSR4	32	H'FCFE3910	32
	Port mode control set/reset register_5	PMCSR5	32	H'FCFE3914	32
	Port mode control set/reset register_6	PMCSR6	32	H'FCFE3918	32
	Port mode control set/reset register_7	PMCSR7	32	H'FCFE391C	32
	Port mode control set/reset register_8	PMCSR8	32	H'FCFE3920	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Ports	Port mode control set/reset register_9	PMCSR9	32	H'FCFE3924	32
	Port function control additional expansion register_0	PFCAE0	16	H'FCFE3A00	16
	Port function control additional expansion register_1	PFCAE1	16	H'FCFE3A04	16
	Port function control additional expansion register_2	PFCAE2	16	H'FCFE3A08	16
	Port function control additional expansion register_3	PFCAE3	16	H'FCFE3A0C	16
	Port function control additional expansion register_4	PFCAE4	16	H'FCFE3A10	16
	Port function control additional expansion register_5	PFCAE5	16	H'FCFE3A14	16
	Port function control additional expansion register_6	PFCAE6	16	H'FCFE3A18	16
	Port function control additional expansion register_7	PFCAE7	16	H'FCFE3A1C	16
	Port function control additional expansion register_8	PFCAE8	16	H'FCFE3A20	16
	Port function control additional expansion register_9	PFCAE9	16	H'FCFE3A24	16
	Port input buffer control register_0	PIBC0	16	H'FCFE7000	16
	Port input buffer control register_1	PIBC1	16	H'FCFE7004	16
	Port input buffer control register_2	PIBC2	16	H'FCFE7008	16
	Port input buffer control register_3	PIBC3	16	H'FCFE700C	16
	Port input buffer control register_4	PIBC4	16	H'FCFE7010	16
	Port input buffer control register_5	PIBC5	16	H'FCFE7014	16
	Port input buffer control register_6	PIBC6	16	H'FCFE7018	16
	Port input buffer control register_7	PIBC7	16	H'FCFE701C	16
	Port input buffer control register_8	PIBC8	16	H'FCFE7020	16
	Port input buffer control register_9	PIBC9	16	H'FCFE7024	16
	Port bi-direction control register_1	PBDC1	16	H'FCFE7104	16
	Port bi-direction control register_2	PBDC2	16	H'FCFE7108	16
	Port bi-direction control register_3	PBDC3	16	H'FCFE710C	16
	Port bi-direction control register_4	PBDC4	16	H'FCFE7110	16
	Port bi-direction control register_5	PBDC5	16	H'FCFE7114	16
	Port bi-direction control register_6	PBDC6	16	H'FCFE7118	16
	Port bi-direction control register_7	PBDC7	16	H'FCFE711C	16
	Port bi-direction control register_8	PBDC8	16	H'FCFE7120	16
	Port bi-direction control register_9	PBDC9	16	H'FCFE7124	16
	Port IP control register_1	PIPC1	16	H'FCFE7204	16
	Port IP control register_2	PIPC2	16	H'FCFE7208	16
	Port IP control register_3	PIPC3	16	H'FCFE720C	16
	Port IP control register_4	PIPC4	16	H'FCFE7210	16
	Port IP control register_5	PIPC5	16	H'FCFE7214	16
	Port IP control register_6	PIPC6	16	H'FCFE7218	16
	Port IP control register_7	PIPC7	16	H'FCFE721C	16
	Port IP control register_8	PIPC8	16	H'FCFE7220	16
	Port IP control register_9	PIPC9	16	H'FCFE7224	16
	Port pin read register_J0	JPPR0	16	H'FCFE7B20	16
	Port mode control register_J0	JPMC0	16	H'FCFE7B40	16
	Port mode control set/reset register_J0	JPMCSR0	32	H'FCFE7B90	32
	Port input buffer control register_J0	JPIBC0	16	H'FCFE7F00	16
	Serial sound interface noise canceler control register	SNCR	32	H'FCFE3C00	32
Power-down modes	Standby control register 1	STBCR1	8	H'FCFE0020	8
	Standby control register 2	STBCR2	8	H'FCFE0024	8
	Standby control register 3	STBCR3	8	H'FCFE0420	8
	Standby control register 4	STBCR4	8	H'FCFE0424	8
	Standby control register 5	STBCR5	8	H'FCFE0428	8
	Standby control register 6	STBCR6	8	H'FCFE042C	8
	Standby control register 7	STBCR7	8	H'FCFE0430	8
	Standby control register 8	STBCR8	8	H'FCFE0434	8

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Power-down modes	Standby control register 9	STBCR9	8	H'FCFE0438	8
	Standby control register 10	STBCR10	8	H'FCFE043C	8
	Standby control register 11	STBCR11	8	H'FCFE0440	8
	Standby control register 12	STBCR12	8	H'FCFE0444	8
	Software reset control register 1	SWRSTCR1	8	H'FCFE0460	8
	Software reset control register 2*2	SWRSTCR2	8	H'FCFE0464	8
	System control register 1	SYSCR1	8	H'FCFE0400	8
	System control register 2	SYSCR2	8	H'FCFE0404	8
	System control register 3	SYSCR3	8	H'FCFE0408	8
	CPU status register	CPUSTS	8	H'FCFE0018	8
	Standby request register 1	STBREQ1	8	H'FCFE0030	8
	Standby request register 2	STBREQ2	8	H'FCFE0034	8
	Standby acknowledge register 1	STBACK1	8	H'FCFE0040	8
	Standby acknowledge register 2	STBACK2	8	H'FCFE0044	8
	On-chip data-retention RAM area setting register	RRAMKP	8	H'FCFF1800	8
	Deep standby control register	DSCTR	8	H'FCFF1802	8
	Deep standby cancel source select register	DSSSR	16	H'FCFF1804	16
	Deep standby cancel edge select register	DSESR	16	H'FCFF1806	16
	Deep standby cancel source flag register	DSFR	16	H'FCFF1808	16
	XTAL crystal oscillator gain control register	XTALCTR	8	H'FCFF1810	8
Debugger interface	DAPROM Peripheral ID4 Register	DAPROM_PERIPHID4	32	H'FC000FD0	32
	DAPROM Peripheral ID0 Register	DAPROM_PERIPHID0	32	H'FC000FE0	32
	DAPROM Peripheral ID1Register	DAPROM_PERIPHID1	32	H'FC000FE4	32
	DAPROM Peripheral ID2 Register	DAPROM_PERIPHID2	32	H'FC000FE8	32
	DAPROM Peripheral ID3 Register	DAPROM_PERIPHID3	32	H'FC000FEC	32
	DAPROM Component ID0 Register	DAPROM_COMPID0	32	H'FC000FF0	32
	DAPROM Component ID1 Register	DAPROM_COMPID1	32	H'FC000FF4	32
	DAPROM Component ID2 Register	DAPROM_COMPID2	32	H'FC000FF8	32
	DAPROM Component ID3 Register	DAPROM_COMPID3	32	H'FC000FFC	32
	Mode reset control register	ICEREGMDRSTCTL	32	H'FC00F000	32
	JTAG trace select register	ICEREGJTTRCSEL	32	H'FC00F004	32
	Clock power control register	ICEREGCLKPWRCTRL	32	H'FC00F014	32
	Lock access register	ICEREGLOCKACCES	32	H'FC00FFB0	32
	2ndDAPROM Peripheral ID4 Register	2ndDAPROM_PERIPHID4	32	H'FC020FD0	32
	2ndDAPROM Peripheral ID0 Register	2ndDAPROM_PERIPHID0	32	H'FC020FE0	32
	2ndDAPROM Peripheral ID1Register	2ndDAPROM_PERIPHID1	32	H'FC020FE4	32
	2ndDAPROM Peripheral ID2 Register	2ndDAPROM_PERIPHID2	32	H'FC020FE8	32
	2ndDAPROM Peripheral ID3 Register	2ndDAPROM_PERIPHID3	32	H'FC020FEC	32
	2ndDAPROM Component ID0 Register	2ndDAPROM_COMPID0	32	H'FC020FF0	32
	2ndDAPROM Component ID1 Register	2ndDAPROM_COMPID1	32	H'FC020FF4	32
	2ndDAPROM Component ID2 Register	2ndDAPROM_COMPID2	32	H'FC020FF8	32
	2ndDAPROM Component ID3 Register	2ndDAPROM_COMPID3	32	H'FC020FFC	32
	CPU-ETF RAM Size Register	CPU_ETF_RSZ	32	H'FC021004	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Debugger interface	CPU-ETF Status Register	CPU_ETF_STS	32	H'FC02100C	32
	CPU-ETF RAM Read Data Register	CPU_ETF_RRD	32	H'FC021010	32
	CPU-ETF RAM Read Pointer Register	CPU_ETF_RRP	32	H'FC021014	32
	CPU-ETF RAM Write Pointer Register	CPU_ETF_RWP	32	H'FC021018	32
	CPU-ETF Trigger Counter Register	CPU_ETF_TRG	32	H'FC02101C	32
	CPU-ETF Control Register	CPU_ETF_CTL	32	H'FC021020	32
	CPU-ETF RAM Write Data Register	CPU_ETF_RWD	32	H'FC021024	32
	CPU-ETF Mode Register	CPU_ETF_MODE	32	H'FC021028	32
	CPU-ETF Latched Buffer Fill Level Register	CPU_ETF_LBUFLEVEL	32	H'FC02102C	32
	CPU-ETF Current Buffer Fill Level Register	CPU_ETF_CBUFLEVEL	32	H'FC021030	32
	CPU-ETF Buffer Level Water Mark Register	CPU_ETF_BUFWM	32	H'FC021034	32
	CPU-ETF RAM Read Pointer High Register	CPU_ETF_RRPHI	32	H'FC021038	32
	CPU-ETF RAM Write Pointer High Register	CPU_ETF_RWPHI	32	H'FC02103C	32
	CPU-ETF Formatter and Flush Status Register	CPU_ETF_FFSR	32	H'FC021300	32
	CPU-ETF Formatter and Flush Control Register	CPU_ETF_FFCR	32	H'FC021304	32
	CPU-ETF Periodic Synchronization Counter Register	CPU_ETF_PSCR	32	H'FC021308	32
	CPU-ETF Claim Tag Set Register	CPU_ETF_CLAIMSET	32	H'FC021FA0	32
	CPU-ETF Claim Tag Clear Register	CPU_ETF_CLAIMCLR	32	H'FC021FA4	32
	CPU-ETF Lock Access Register	CPU_ETF_LAR	32	H'FC021FB0	32
	CPU-ETF Lock Status Register	CPU_ETF_LSR	32	H'FC021FB4	32
	CPU-ETF Authentication Status Register	CPU_ETF_AUTHSTATUS	32	H'FC021FB8	32
	CPU-ETF Device Configuration Register	CPU_ETF_DEVID	32	H'FC021FC8	32
	CPU-ETF Device Type Identifier Register	CPU_ETF_DEVTYPE	32	H'FC021FCC	32
	CPU-ETF Peripheral ID4 Register	CPU_ETF_PERIPHID4	32	H'FC021FD0	32
	CPU-ETF Peripheral ID0 Register	CPU_ETF_PERIPHID0	32	H'FC021FE0	32
	CPU-ETF Peripheral ID1 Register	CPU_ETF_PERIPHID1	32	H'FC021FE4	32
	CPU-ETF Peripheral ID2 Register	CPU_ETF_PERIPHID2	32	H'FC021FE8	32
	CPU-ETF Peripheral ID3 Register	CPU_ETF_PERIPHID3	32	H'FC021FEC	32
	CPU-ETF Component ID0 Register	CPU_ETF_COMPID0	32	H'FC021FF0	32
	CPU-ETF Component ID1 Register	CPU_ETF_COMPID1	32	H'FC021FF4	32
	CPU-ETF Component ID2 Register	CPU_ETF_COMPID2	32	H'FC021FF8	32
	CPU-ETF Component ID3 Register	CPU_ETF_COMPID3	32	H'FC021FFC	32
	CPU-CTICS CTI Control Register	CPU_CTICS_CTICONTROL	32	H'FC022000	32
	CPU-CTICS CTI Interrupt Acknowledge Register	CPU_CTICS_CTIINTACK	32	H'FC022010	32
	CPU-CTICS CTI Application Trigger Set Register	CPU_CTICS_CTIAPPSET	32	H'FC022014	32
	CPU-CTICS CTI Application Trigger Clear Register	CPU_CTICS_CTIAPPCLEAR	32	H'FC022018	32
	CPU-CTICS CTI Application Pulse Register	CPU_CTICS_CTIAPPPULSE	32	H'FC02201C	32
	CPU-CTICS CTI Trigger to Channel Enable Register0	CPU_CTICS_CTIINEN0	32	H'FC022020	32
	CPU-CTICS CTI Trigger to Channel Enable Register1	CPU_CTICS_CTIINEN1	32	H'FC022024	32
	CPU-CTICS CTI Trigger to Channel Enable Register2	CPU_CTICS_CTIINEN2	32	H'FC022028	32
	CPU-CTICS CTI Trigger to Channel Enable Register3	CPU_CTICS_CTIINEN3	32	H'FC02202C	32
	CPU-CTICS CTI Trigger to Channel Enable Register4	CPU_CTICS_CTIINEN4	32	H'FC022030	32
	CPU-CTICS CTI Trigger to Channel Enable Register5	CPU_CTICS_CTIINEN5	32	H'FC022034	32
	CPU-CTICS CTI Trigger to Channel Enable Register6	CPU_CTICS_CTIINEN6	32	H'FC022038	32
	CPU-CTICS CTI Trigger to Channel Enable Register7	CPU_CTICS_CTIINEN7	32	H'FC02203C	32
	CPU-CTICS CTI Channel to Trigger Enable Register0	CPU_CTICS_CTIOUTEN0	32	H'FC0220A0	32
	CPU-CTICS CTI Channel to Trigger Enable Register1	CPU_CTICS_CTIOUTEN1	32	H'FC0220A4	32
	CPU-CTICS CTI Channel to Trigger Enable Register2	CPU_CTICS_CTIOUTEN2	32	H'FC0220A8	32
	CPU-CTICS CTI Channel to Trigger Enable Register3	CPU_CTICS_CTIOUTEN3	32	H'FC0220AC	32
	CPU-CTICS CTI Channel to Trigger Enable Register4	CPU_CTICS_CTIOUTEN4	32	H'FC0220B0	32
	CPU-CTICS CTI Channel to Trigger Enable Register5	CPU_CTICS_CTIOUTEN5	32	H'FC0220B4	32
	CPU-CTICS CTI Channel to Trigger Enable Register6	CPU_CTICS_CTIOUTEN6	32	H'FC0220B8	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Debugger interface	CPU-CTICS CTI Channel to Trigger Enable Register7	CPU_CTICS_CTIOUTEN7	32	H'FC0220BC	32
	CPU-CTICS CTI Trigger In Status Register	CPU_CTICS_ CTITRIGINSTATUS	32	H'FC022130	32
	CPU-CTICS CTI Trigger Out Status Register	CPU_CTICS_ CTITRIGOUTSTATUS	32	H'FC022134	32
	CPU-CTICS CTI Channel In Status Register	CPU_CTICS_ CTICHINSTATUS	32	H'FC022138	32
	CPU-CTICS CTI Channel Out Status Register	CPU_CTICS_ CTICHOUTSTATUS	32	H'FC02213C	32
	CPU-CTICS Enable CTI Channel Gate Register	CPU_CTICS_CTIGATE	32	H'FC022140	32
	CPU-CTICS External Multiplexor Control Register	CPU_CTICS_ASICCTL	32	H'FC022144	32
	CPU-CTICS Claim Tag Set Register	CPU_CTICS_CLAIMSET	32	H'FC022FA0	32
	CPU-CTICS Claim Tag Clear Register	CPU_CTICS_CLAIMCLR	32	H'FC022FA4	32
	CPU-CTICS Lock Access Register	CPU_CTICS_LAR	32	H'FC022FB0	32
	CPU-CTICS Lock Status Register	CPU_CTICS_LSR	32	H'FC022FB4	32
	CPU-CTICS Authentication Status Register	CPU_CTICS_AUTHSTATUS	32	H'FC022FB8	32
	CPU-CTICS Device Configuration Register	CPU_CTICS_DEVID	32	H'FC022FC8	32
	CPU-CTICS Device Type Identifier Register	CPU_CTICS_DEVTYPE	32	H'FC022FCC	32
	CPU-CTICS Peripheral ID4 Register	CPU_CTICS_PERIPHID4	32	H'FC022FD0	32
	CPU-CTICS Peripheral ID0 Register	CPU_CTICS_PERIPHID0	32	H'FC022FE0	32
	CPU-CTICS Peripheral ID1 Register	CPU_CTICS_PERIPHID1	32	H'FC022FE4	32
	CPU-CTICS Peripheral ID2 Register	CPU_CTICS_PERIPHID2	32	H'FC022FE8	32
	CPU-CTICS Peripheral ID3 Register	CPU_CTICS_PERIPHID3	32	H'FC022FEC	32
	CPU-CTICS Component ID0 Register	CPU_CTICS_COMPID0	32	H'FC022FF0	32
	CPU-CTICS Component ID1 Register	CPU_CTICS_COMPID1	32	H'FC022FF4	32
	CPU-CTICS Component ID2 Register	CPU_CTICS_COMPID2	32	H'FC022FF8	32
	CPU-CTICS Component ID3 Register	CPU_CTICS_COMPID3	32	H'FC022FFC	32
	CPU-TPIU Supported Port Size Register	CPU_TPIU_Supported port sizes	32	H'FC023000	32
	CPU-TPIU Current Port Size Register	CPU_TPIU_Current port size	32	H'FC023004	32
	CPU-TPIU Trigger Modes Register	CPU_TPIU_Supported trigger modes	32	H'FC023100	32
	CPU-TPIU Trigger Counter Register	CPU_TPIU_Trigger counter value	32	H'FC023104	32
	CPU-TPIU Trigger Multiplier Register	CPU_TPIU_Trigger multiplier	32	H'FC023108	32
	CPU-TPIU Supported Test Patterns/Modes Register	CPU_TPIU_Supported test pattern/modes	32	H'FC023200	32
	CPU-TPIU Current Test Patterns/Modes Register	CPU_TPIU_Current test pattern/mode	32	H'FC023204	32
	CPU-TPIU TPIU Test Pattern Repeat Register	CPU_TPIU_Test pattern repeat counter	32	H'FC023208	32
	CPU-TPIU Formatter and Flush Status Register	CPU_TPIU_Formatter and flush status	32	H'FC023300	32
	CPU-TPIU Formatter and Flush Control Register	CPU_TPIU_Formatter and flush control	32	H'FC023304	32
	CPU-TPIU Formatter Synchronization counter Register	CPU_TPIU_Formatter synchronization counter	32	H'FC023308	32
	CPU-TPIU Claim Tag Set Register	CPU_TPIU_CLAIMSET	32	H'FC023FA0	32
	CPU-TPIU Claim Tag Clear Register	CPU_TPIU_CLAIMCLR	32	H'FC023FA4	32
	CPU-TPIU Lock Access Register	CPU_TPIU_LAR	32	H'FC023FB0	32
	CPU-TPIU Lock Status Register	CPU_TPIU_LSR	32	H'FC023FB4	32
	CPU-TPIU Authentication Status Register	CPU_TPIU_AUTHSTATUS	32	H'FC023FB8	32
	CPU-TPIU Device Configuration Register	CPU_TPIU_DEVID	32	H'FC023FC8	32
	CPU-TPIU Device Type Identifier Register	CPU_TPIU_DEVTYPE	32	H'FC023FCC	32
	CPU-TPIU Peripheral ID4 Register	CPU_TPIU_PERIPHID4	32	H'FC023FD0	32
	CPU-TPIU Peripheral ID0 Register	CPU_TPIU_PERIPHID0	32	H'FC023FE0	32
	CPU-TPIU Peripheral ID1 Register	CPU_TPIU_PERIPHID1	32	H'FC023FE4	32
	CPU-TPIU Peripheral ID2 Register	CPU_TPIU_PERIPHID2	32	H'FC023FE8	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Debugger interface	CPU-TPIU Peripheral ID3 Register	CPU_TPIU_PERIPHID3	32	H'FC023FEC	32
	CPU-TPIU Component ID0 Register	CPU_TPIU_COMPID0	32	H'FC023FF0	32
	CPU-TPIU Component ID1 Register	CPU_TPIU_COMPID1	32	H'FC023FF4	32
	CPU-TPIU Component ID2 Register	CPU_TPIU_COMPID2	32	H'FC023FF8	32
	CPU-TPIU Component ID3 Register	CPU_TPIU_COMPID3	32	H'FC023FFC	32
	CPU-TraceFunnel CSTF Control Register	CPU_TraceFunnel_FUNCTL	32	H'FC024000	32
	CPU-TraceFunnel CSTF Priority Control Register	CPU_TraceFunnel_PRICTL	32	H'FC024004	32
	CPU-TraceFunnel Claim Tag Set Register	CPU_TraceFunnel_CLAIMSET	32	H'FC024FA0	32
	CPU-TraceFunnel Claim Tag Clear Register	CPU_TraceFunnel_CLAIMCLR	32	H'FC024FA4	32
	CPU-TraceFunnel Lock Access Register	CPU_TraceFunnel_LAR	32	H'FC024FB0	32
	CPU-TraceFunnel Lock Status Register	CPU_TraceFunnel_LSR	32	H'FC024FB4	32
	CPU-TraceFunnel Authentication Status Register	CPU_TraceFunnel_ AUTHSTATUS	32	H'FC024FB8	32
	CPU-TraceFunnel Device Configuration Register	CPU_TraceFunnel_DEVID	32	H'FC024FC8	32
	CPU-TraceFunnel Device Type Identifier Register	CPU_TraceFunnel_DEVTYPE	32	H'FC024FCC	32
	CPU-TraceFunnel Peripheral ID4 Register	CPU_TraceFunnel_PERIPHID 4	32	H'FC024FD0	32
	CPU-TraceFunnel Peripheral ID0 Register	CPU_TraceFunnel_PERIPHID 0	32	H'FC024FE0	32
	CPU-TraceFunnel Peripheral ID1 Register	CPU_TraceFunnel_PERIPHID 1	32	H'FC024FE4	32
	CPU-TraceFunnel Peripheral ID2 Register	CPU_TraceFunnel_PERIPHID 2	32	H'FC024FE8	32
	CPU-TraceFunnel Peripheral ID3 Register	CPU_TraceFunnel_PERIPHID 3	32	H'FC024FEC	32
	CPU-TraceFunnel Component ID0 Register	CPU_TraceFunnel_COMPID0	32	H'FC024FF0	32
	CPU-TraceFunnel Component ID1 Register	CPU_TraceFunnel_COMPID1	32	H'FC024FF4	32
	CPU-TraceFunnel Component ID2 Register	CPU_TraceFunnel_COMPID2	32	H'FC024FF8	32
	CPU-TraceFunnel Component ID3 Register	CPU_TraceFunnel_COMPID3	32	H'FC024FFC	32
	CA9-DBG Debug ID Register	CA9_DBG_DBGDIDR	32	H'FC030000	32
	CA9-DBG Watchpoint Fault Address Register	CA9_DBG_DBGWFAR	32	H'FC030018	32
	CA9-DBG Vector Catch Register	CA9_DBG_DBGVCR	32	H'FC03001C	32
	CA9-DBG Host to Target Data Transfer Register	CA9_DBG_DBGDTRRXext	32	H'FC030080	32
	CA9-DBG Instruction Transfer/Program Counter Sampling Register	CA9_DBG_DBGITR/ DBGPCSR	32	H'FC030084	32
	CA9-DBG Debug Status and Control Register	CA9_DBG_DBGDSCRext	32	H'FC030088	32
	CA9-DBG Target to Host Data Transfer Register	CA9_DBG_DBGDTRTXext	32	H'FC03008C	32
	CA9-DBG Debug Run Control Register	CA9_DBG_DBGDRCR	32	H'FC030090	32
	CA9-DBG Breakpoint Value Register 0	CA9_DBG_DBGBVR0	32	H'FC030100	32
	CA9-DBG Breakpoint Value Register 1	CA9_DBG_DBGBVR1	32	H'FC030104	32
	CA9-DBG Breakpoint Value Register 2	CA9_DBG_DBGBVR2	32	H'FC030108	32
	CA9-DBG Breakpoint Value Register 3	CA9_DBG_DBGBVR3	32	H'FC03010C	32
	CA9-DBG Breakpoint Value Register 4	CA9_DBG_DBGBVR4	32	H'FC030110	32
	CA9-DBG Breakpoint Value Register 5	CA9_DBG_DBGBVR5	32	H'FC030114	32
	CA9-DBG Breakpoint Control Register 0	CA9_DBG_DBGBCR0	32	H'FC030140	32
	CA9-DBG Breakpoint Control Register 1	CA9_DBG_DBGBCR1	32	H'FC030144	32
	CA9-DBG Breakpoint Control Register 2	CA9_DBG_DBGBCR2	32	H'FC030148	32
	CA9-DBG Breakpoint Control Register 3	CA9_DBG_DBGBCR3	32	H'FC03014C	32
	CA9-DBG Breakpoint Control Register 4	CA9_DBG_DBGBCR4	32	H'FC030150	32
	CA9-DBG Breakpoint Control Register 5	CA9_DBG_DBGBCR5	32	H'FC030154	32
	CA9-DBG Watchpoint Value Register 0	CA9_DBG_DBGWVR0	32	H'FC030180	32
	CA9-DBG Watchpoint Value Register 1	CA9_DBG_DBGWVR1	32	H'FC030184	32
	CA9-DBG Watchpoint Value Register 2	CA9_DBG_DBGWVR2	32	H'FC030188	32
	CA9-DBG Watchpoint Value Register 3	CA9_DBG_DBGWVR3	32	H'FC03018C	32
	CA9-DBG Watchpoint Control Register 0	CA9_DBG_DBGWCR0	32	H'FC0301C0	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Debugger interface	CA9-DBG Watchpoint Control Register 1	CA9_DBG_DBGWCR1	32	H'FC0301C4	32
	CA9-DBG Watchpoint Control Register 2	CA9_DBG_DBGWCR2	32	H'FC0301C8	32
	CA9-DBG Watchpoint Control Register 3	CA9_DBG_DBGWCR3	32	H'FC0301CC	32
	CA9-DBG Main ID Register	CA9_DBG_MIDR	32	H'FC030D00	32
	CA9-DBG Cache Type Register	CA9_DBG_CTR	32	H'FC030D04	32
	CA9-DBG TLB Type Register	CA9_DBG_TLBTR	32	H'FC030D0C	32
	CA9-DBG Multiprocessor Affinity Register	CA9_DBG_MPIDR	32	H'FC030D10	32
	CA9-DBG Revision ID register	CA9_DBG_REVIDR	32	H'FC030D14	32
	CA9-DBG Processor Feature Register 0	CA9_DBG_ID_PFR0	32	H'FC030D20	32
	CA9-DBG Processor Feature Register 1	CA9_DBG_ID_PFR1	32	H'FC030D24	32
	CA9-DBG Debug Feature Register 0	CA9_DBG_ID_DFR0	32	H'FC030D28	32
	CA9-DBG Memory Model Feature Register 0	CA9_DBG_ID_MMFR0	32	H'FC030D30	32
	CA9-DBG Memory Model Feature Register 1	CA9_DBG_ID_MMFR1	32	H'FC030D34	32
	CA9-DBG Memory Model Feature Register 2	CA9_DBG_ID_MMFR2	32	H'FC030D38	32
	CA9-DBG Memory Model Feature Register 3	CA9_DBG_ID_MMFR3	32	H'FC030D3C	32
	CA9-DBG Instruction Set Attribute Register 0	CA9_DBG_ID_ISAR0	32	H'FC030D40	32
	CA9-DBG Instruction Set Attribute Register 1	CA9_DBG_ID_ISAR1	32	H'FC030D44	32
	CA9-DBG Instruction Set Attribute Register 2	CA9_DBG_ID_ISAR2	32	H'FC030D48	32
	CA9-DBG Instruction Set Attribute Register 3	CA9_DBG_ID_ISAR3	32	H'FC030D4C	32
	CA9-DBG Instruction Set Attribute Register 4	CA9_DBG_ID_ISAR4	32	H'FC030D50	32
	CA9-DBG Claim Tag Set Register	CA9_DBG_CLAIMSET	32	H'FC030FA0	32
	CA9-DBG Claim Tag Clear Register	CA9_DBG_CLAIMCLR	32	H'FC030FA4	32
	CA9-DBG Lock Access Register	CA9_DBG_LAR	32	H'FC030FB0	32
	CA9-DBG Lock Status Register	CA9_DBG_LSR	32	H'FC030FB4	32
	CA9-DBG Authentication Status Register	CA9_DBG_AUTHSTATUS	32	H'FC030FB8	32
	CA9-DBG Device Configuration Register	CA9_DBG_DEVID	32	H'FC030FC8	32
	CA9-DBG Device Type Identifier Register	CA9_DBG_DEVTYPE	32	H'FC030FCC	32
	CA9-DBG Peripheral ID4 Register	CA9_DBG_PERIPHID4	32	H'FC030FD0	32
	CA9-DBG Peripheral ID0 Register	CA9_DBG_PERIPHID0	32	H'FC030FE0	32
	CA9-DBG Peripheral ID1 Register	CA9_DBG_PERIPHID1	32	H'FC030FE4	32
	CA9-DBG Peripheral ID2 Register	CA9_DBG_PERIPHID2	32	H'FC030FE8	32
	CA9-DBG Peripheral ID3 Register	CA9_DBG_PERIPHID3	32	H'FC030FEC	32
	CA9-DBG Component ID0 Register	CA9_DBG_COMPID0	32	H'FC030FF0	32
	CA9-DBG Component ID1 Register	CA9_DBG_COMPID1	32	H'FC030FF4	32
	CA9-DBG Component ID2 Register	CA9_DBG_COMPID2	32	H'FC030FF8	32
	CA9-DBG Component ID3 Register	CA9_DBG_COMPID3	32	H'FC030FFC	32
	CA9-PMU Event Count Register 0	CA9_PMU_PMXEVCNTR0	32	H'FC031000	32
	CA9-PMU Event Count Register 1	CA9_PMU_PMXEVCNTR1	32	H'FC031004	32
	CA9-PMU Event Count Register 2	CA9_PMU_PMXEVCNTR2	32	H'FC031008	32
	CA9-PMU Event Count Register 3	CA9_PMU_PMXEVCNTR3	32	H'FC03100C	32
	CA9-PMU Event Count Register 4	CA9_PMU_PMXEVCNTR4	32	H'FC031010	32
	CA9-PMU Event Count Register 5	CA9_PMU_PMXEVCNTR5	32	H'FC031014	32
	CA9-PMU Cycle Count Register	CA9_PMU_PMCCNTR	32	H'FC03107C	32
	CA9-PMU Event Counter Selection Register 0	CA9_PMU_PMXEVTYPER0	32	H'FC031400	32
	CA9-PMU Event Counter Selection Register 1	CA9_PMU_PMXEVTYPER1	32	H'FC031404	32
	CA9-PMU Event Counter Selection Register 2	CA9_PMU_PMXEVTYPER2	32	H'FC031408	32
	CA9-PMU Event Counter Selection Register 3	CA9_PMU_PMXEVTYPER3	32	H'FC03140C	32
	CA9-PMU Event Counter Selection Register 4	CA9_PMU_PMXEVTYPER4	32	H'FC031410	32
	CA9-PMU Event Counter Selection Register 5	CA9_PMU_PMXEVTYPER5	32	H'FC031414	32
	CA9-PMU Count Enable Set Register	CA9_PMU_PMCNTENSET	32	H'FC031C00	32
	CA9-PMU Count Enable Clear Register	CA9_PMU_PMCNTENCLR	32	H'FC031C20	32
	CA9-PMU Interrupt Enable Set Register	CA9_PMU_PMINTENSET	32	H'FC031C40	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Debugger interface	CA9-PMU Interrupt Enable Clear Register	CA9_PMU_PMINTENCLR	32	H'FC031C60	32
	CA9-PMU Overflow Flag Status Register	CA9_PMU_PMOVSR	32	H'FC031C80	32
	CA9-PMU Software Increment Register	CA9_PMU_PMSWINC	32	H'FC031CA0	32
	CA9-PMU Performance Monitor Control Register	CA9_PMU_PMCR	32	H'FC031E04	32
	CA9-PMU User Enable Register	CA9_PMU_PMUSERENR	32	H'FC031E08	32
	CA9-PMU Claim Tag Set Register	CA9_PMU_CLAIMSET	32	H'FC031FA0	32
	CA9-PMU Claim Tag Clear Register	CA9_PMU_CLAIMCLR	32	H'FC031FA4	32
	CA9-PMU Lock Access Register	CA9_PMU_LAR	32	H'FC031FB0	32
	CA9-PMU Lock Status Register	CA9_PMU_LSR	32	H'FC031FB4	32
	CA9-PMU Authentication Status Register	CA9_PMU_AUTHSTATUS	32	H'FC031FB8	32
	CA9-PMU Device Configuration Register	CA9_PMU_DEVID	32	H'FC031FC8	32
	CA9-PMU Device Type Identifier Register	CA9_PMU_DEVTYPE	32	H'FC031FCC	32
	CA9-PMU Peripheral ID4 Register	CA9_PMU_PERIPHID4	32	H'FC031FD0	32
	CA9-PMU Peripheral ID0 Register	CA9_PMU_PERIPHID0	32	H'FC031FE0	32
	CA9-PMU Peripheral ID1 Register	CA9_PMU_PERIPHID1	32	H'FC031FE4	32
	CA9-PMU Peripheral ID2 Register	CA9_PMU_PERIPHID2	32	H'FC031FE8	32
	CA9-PMU Peripheral ID3 Register	CA9_PMU_PERIPHID3	32	H'FC031FEC	32
	CA9-PMU Component ID0 Register	CA9_PMU_COMPID0	32	H'FC031FF0	32
	CA9-PMU Component ID1 Register	CA9_PMU_COMPID1	32	H'FC031FF4	32
	CA9-PMU Component ID2 Register	CA9_PMU_COMPID2	32	H'FC031FF8	32
	CA9-PMU Component ID3 Register	CA9_PMU_COMPID3	32	H'FC031FFC	32
	CA9-CTI CTI Control Register	CA9_CTI_CTICONTROL	32	H'FC038000	32
	CA9-CTI CTI Interrupt Acknowledge Register	CA9_CTI_CTIINTACK	32	H'FC038010	32
	CA9-CTI CTI Application Trigger Set Register	CA9_CTI_CTIAPPSET	32	H'FC038014	32
	CA9-CTI CTI Application Trigger Clear Register	CA9_CTI_CTIAPPCLEAR	32	H'FC038018	32
	CA9-CTI CTI Application Pulse Register	CA9_CTI_CTIAPPPULSE	32	H'FC03801C	32
	CA9-CTI CTI Trigger to Channel Enable Register0	CA9_CTI_CTIINEN0	32	H'FC038020	32
	CA9-CTI CTI Trigger to Channel Enable Register1	CA9_CTI_CTIINEN1	32	H'FC038024	32
	CA9-CTI CTI Trigger to Channel Enable Register2	CA9_CTI_CTIINEN2	32	H'FC038028	32
	CA9-CTI CTI Trigger to Channel Enable Register3	CA9_CTI_CTIINEN3	32	H'FC03802C	32
	CA9-CTI CTI Trigger to Channel Enable Register4	CA9_CTI_CTIINEN4	32	H'FC038030	32
	CA9-CTI CTI Trigger to Channel Enable Register5	CA9_CTI_CTIINEN5	32	H'FC038034	32
	CA9-CTI CTI Trigger to Channel Enable Register6	CA9_CTI_CTIINEN6	32	H'FC038038	32
	CA9-CTI CTI Trigger to Channel Enable Register7	CA9_CTI_CTIINEN7	32	H'FC03803C	32
	CA9-CTI CTI Channel to Trigger Enable Register0	CA9_CTI_CTIOUTEN0	32	H'FC0380A0	32
	CA9-CTI CTI Channel to Trigger Enable Register1	CA9_CTI_CTIOUTEN1	32	H'FC0380A4	32
	CA9-CTI CTI Channel to Trigger Enable Register2	CA9_CTI_CTIOUTEN2	32	H'FC0380A8	32
	CA9-CTI CTI Channel to Trigger Enable Register3	CA9_CTI_CTIOUTEN3	32	H'FC0380AC	32
	CA9-CTI CTI Channel to Trigger Enable Register4	CA9_CTI_CTIOUTEN4	32	H'FC0380B0	32
	CA9-CTI CTI Channel to Trigger Enable Register5	CA9_CTI_CTIOUTEN5	32	H'FC0380B4	32
	CA9-CTI CTI Channel to Trigger Enable Register6	CA9_CTI_CTIOUTEN6	32	H'FC0380B8	32
	CA9-CTI CTI Channel to Trigger Enable Register7	CA9_CTI_CTIOUTEN7	32	H'FC0380BC	32
	CA9-CTI CTI Trigger In Status Register	CA9_CTI_CTITRIGINSTATUS	32	H'FC038130	32
	CA9-CTI CTI Trigger Out Status Register	CA9_CTI_ CTITRIGOUTSTATUS	32	H'FC038134	32
	CA9-CTI CTI Channel In Status Register	CA9_CTI_CTICHINSTATUS	32	H'FC038138	32
	CA9-CTI CTI Channel Out Status Register	CA9_CTI_CTICHOUTSTATUS	32	H'FC03813C	32
	CA9-CTI Enable CTI Channel Gate Register	CA9_CTI_CTIGATE	32	H'FC038140	32
	CA9-CTI External Multiplexor Control Register	CA9_CTI_ASICCTL	32	H'FC038144	32
	CA9-CTI Claim Tag Set Register	CA9_CTI_CLAIMSET	32	H'FC038FA0	32
	CA9-CTI Claim Tag Clear Register	CA9_CTI_CLAIMCLR	32	H'FC038FA4	32
	CA9-CTI Lock Access Register	CA9_CTI_LAR	32	H'FC038FB0	32
	CA9-CTI Lock Status Register	CA9_CTI_LSR	32	H'FC038FB4	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Debugger interface	CA9-CTI Authentication Status Register	CA9_CTI_AUTHSTATUS	32	H'FC038FB8	32
	CA9-CTI Device Configuration Register	CA9_CTI_DEVID	32	H'FC038FC8	32
	CA9-CTI Device Type Identifier Register	CA9_CTI_DEVTYPE	32	H'FC038FCC	32
	CA9-CTI Peripheral ID4 Register	CA9_CTI_PERIPHID4	32	H'FC038FD0	32
	CA9-CTI Peripheral ID0 Register	CA9_CTI_PERIPHID0	32	H'FC038FE0	32
	CA9-CTI Peripheral ID1Register	CA9_CTI_PERIPHID1	32	H'FC038FE4	32
	CA9-CTI Peripheral ID2 Register	CA9_CTI_PERIPHID2	32	H'FC038FE8	32
	CA9-CTI Peripheral ID3 Register	CA9_CTI_PERIPHID3	32	H'FC038FEC	32
	CA9-CTI Component ID0 Register	CA9_CTI_COMPID0	32	H'FC038FF0	32
	CA9-CTI Component ID1 Register	CA9_CTI_COMPID1	32	H'FC038FF4	32
	CA9-CTI Component ID2 Register	CA9_CTI_COMPID2	32	H'FC038FF8	32
	CA9-CTI Component ID3 Register	CA9_CTI_COMPID3	32	H'FC038FFC	32
	PTM-A9 Main Control Register	PTM_A9_ETMCR	32	H'FC03C000	32
	PTM-A9 Configuration Code Register	PTM_A9_ETMCCR	32	H'FC03C004	32
	PTM-A9 Trigger Event Register	PTM_A9_ETMTRIGGER	32	H'FC03C008	32
	PTM-A9 Status Register	PTM_A9_ETMSR	32	H'FC03C010	32
	PTM-A9 System Configuration Register	PTM_A9_ETMSCR	32	H'FC03C014	32
	PTM-A9 TraceEnable Start/Stop Control Register	PTM_A9_ETMTSSCR	32	H'FC03C018	32
	PTM-A9 TraceEnable Event Register	PTM_A9_ETMTEEVR	32	H'FC03C020	32
	PTM-A9 TraceEnable Control Register 1	PTM_A9_ETMTECR1	32	H'FC03C024	32
	PTM-A9 Address Comparator Value Register 1	PTM_A9_ETMACVR1	32	H'FC03C040	32
	PTM-A9 Address Comparator Value Register 2	PTM_A9_ETMACVR2	32	H'FC03C044	32
	PTM-A9 Address Comparator Value Register 3	PTM_A9_ETMACVR3	32	H'FC03C048	32
	PTM-A9 Address Comparator Value Register 4	PTM_A9_ETMACVR4	32	H'FC03C04C	32
	PTM-A9 Address Comparator Value Register 5	PTM_A9_ETMACVR5	32	H'FC03C050	32
	PTM-A9 Address Comparator Value Register 6	PTM_A9_ETMACVR6	32	H'FC03C054	32
	PTM-A9 Address Comparator Value Register 7	PTM_A9_ETMACVR7	32	H'FC03C058	32
	PTM-A9 Address Comparator Value Register 8	PTM_A9_ETMACVR8	32	H'FC03C05C	32
	PTM-A9 Address Comparator Access Type Register 1	PTM_A9_ETMACTR1	32	H'FC03C080	32
	PTM-A9 Address Comparator Access Type Register 2	PTM_A9_ETMACTR2	32	H'FC03C084	32
	PTM-A9 Address Comparator Access Type Register 3	PTM_A9_ETMACTR3	32	H'FC03C088	32
	PTM-A9 Address Comparator Access Type Register 4	PTM_A9_ETMACTR4	32	H'FC03C08C	32
	PTM-A9 Address Comparator Access Type Register 5	PTM_A9_ETMACTR5	32	H'FC03C090	32
	PTM-A9 Address Comparator Access Type Register 6	PTM_A9_ETMACTR6	32	H'FC03C094	32
	PTM-A9 Address Comparator Access Type Register 7	PTM_A9_ETMACTR7	32	H'FC03C098	32
	PTM-A9 Address Comparator Access Type Register 8	PTM_A9_ETMACTR8	32	H'FC03C09C	32
	PTM-A9 Counter Reload Value Register 1	PTM_A9_ETMCNTRLDVR1	32	H'FC03C140	32
	PTM-A9 Counter Reload Value Register 2	PTM_A9_ETMCNTRLDVR2	32	H'FC03C144	32
	PTM-A9 Counter Enable Event Register 1	PTM_A9_ETMCNTENR1	32	H'FC03C150	32
	PTM-A9 Counter Enable Event Register 2	PTM_A9_ETMCNTENR2	32	H'FC03C154	32
	PTM-A9 Counter Reload Event Register 1	PTM_A9_ETMCNTRLDEVR1	32	H'FC03C160	32
	PTM-A9 Counter Reload Event Register 2	PTM_A9_ETMCNTRLDEVR2	32	H'FC03C164	32
	PTM-A9 Counter Value Register 1	PTM_A9_ETMCNTVR1	32	H'FC03C170	32
	PTM-A9 Counter Value Register 2	PTM_A9_ETMCNTVR2	32	H'FC03C174	32
	PTM-A9 State 1 to State 2 Transition Event Register	PTM_A9_ETMSQ12EVR	32	H'FC03C180	32
	PTM-A9 State 2 to State 1 Transition Event Register	PTM_A9_ETMSQ21EVR	32	H'FC03C184	32
	PTM-A9 State 2 to State 3 Transition Event Register	PTM_A9_ETMSQ23EVR	32	H'FC03C188	32
	PTM-A9 State 3 to State 1 Transition Event Register	PTM_A9_ETMSQ31EVR	32	H'FC03C18C	32
	PTM-A9 State 3 to State 2 Transition Event Register	PTM_A9_ETMSQ32EVR	32	H'FC03C190	32
	PTM-A9 State 1 to State 3 Transition Event Register	PTM_A9_ETMSQ13EVR	32	H'FC03C194	32
	PTM-A9 Current Sequencer State Register	PTM_A9_ETMSQR	32	H'FC03C19C	32
	PTM-A9 External Output Event Register 1	PTM_A9_ETMEXTOUTEVR1	32	H'FC03C1A0	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
Debugger interface	PTM-A9 External Output Event Register 2	PTM_A9_ETMEXTOUTEVR2	32	H'FC03C1A4	32
	PTM-A9 Context ID Comparator Value 1 Register	PTM_A9_ETMCIDCVR1	32	H'FC03C1B0	32
	PTM-A9 Context ID Comparator Mask Register	PTM_A9_ETMCIDCMR	32	H'FC03C1BC	32
	PTM-A9 Synchronization Frequency Register	PTM_A9_ETMSYNCFR	32	H'FC03C1E0	32
	PTM-A9 ID Register	PTM_A9_ETMIDR	32	H'FC03C1E4	32
	PTM-A9 Configuration Code Extension Register	PTM_A9_ETMCCER	32	H'FC03C1E8	32
	PTM-A9 Extended External Input Selection Register	PTM_A9_ETMEXTINSELR	32	H'FC03C1EC	32
	PTM-A9 Timestamp Event Register	PTM_A9_ETMTSEVR	32	H'FC03C1F8	32
	PTM-A9 Auxiliary Control Register	PTM_A9_ETMAUXCR	32	H'FC03C1FC	32
	PTM-A9 CoreSight Trace ID Register	PTM_A9_ETMTRACEIDR	32	H'FC03C200	32
	PTM-A9 OS Lock Status Register	PTM_A9_OSLSR	32	H'FC03C304	32
	PTM-A9 Claim Tag Set Register	PTM_A9_CLAIMSET	32	H'FC031FA0	32
	PTM-A9 Claim Tag Clear Register	PTM_A9_CLAIMCLR	32	H'FC031FA4	32
	PTM-A9 Lock Access Register	PTM_A9_LAR	32	H'FC031FB0	32
	PTM-A9 Lock Status Register	PTM_A9_LSR	32	H'FC031FB4	32
	PTM-A9 Authentication Status Register	PTM_A9_AUTHSTATUS	32	H'FC031FB8	32
	PTM-A9 Device Type Identifier Register	PTM_A9_DEVTYPE	32	H'FC031FCC	32
	PTM-A9 Peripheral ID4 Register	PTM_A9_PERIPHID4	32	H'FC031FD0	32
	PTM-A9 Peripheral ID0 Register	PTM_A9_PERIPHID0	32	H'FC031FE0	32
	PTM-A9 Peripheral ID1 Register	PTM_A9_PERIPHID1	32	H'FC031FE4	32
	PTM-A9 Peripheral ID2 Register	PTM_A9_PERIPHID2	32	H'FC031FE8	32
	PTM-A9 Peripheral ID3 Register	PTM_A9_PERIPHID3	32	H'FC031FEC	32
	PTM-A9 Component ID0 Register	PTM_A9_COMPID0	32	H'FC031FF0	32
	PTM-A9 Component ID1 Register	PTM_A9_COMPID1	32	H'FC031FF4	32
	PTM-A9 Component ID2 Register	PTM_A9_COMPID2	32	H'FC031FF8	32
	PTM-A9 Component ID3 Register	PTM_A9_COMPID3	32	H'FC031FFC	32
JPEG codec unit*2	JPEG code mode register	JCMOD	8	H'E8017000	8
	JPEG code command register	JCCMD	8	H'E8017001	8
	JPEG code quantization table number register	JCQTN	8	H'E8017003	8
	JPEG code Huffman table number register	JCHTN	8	H'E8017004	8
	JPEG code DRI upper register	JCDRIU	8	H'E8017005	8
	JPEG code DRI lower register	JCDRID	8	H'E8017006	8
	JPEG code vertical size upper register	JCVSZU	8	H'E8017007	8
	JPEG code vertical size lower register	JCVSZD	8	H'E8017008	8
	JPEG code horizontal size upper register	JCHSZU	8	H'E8017009	8
	JPEG code horizontal size lower register	JCHSZD	8	H'E801700A	8
	JPEG code data count upper register	JCDTCU	8	H'E801700B	8
	JPEG code data count middle register	JCDTCM	8	H'E801700C	8
	JPEG code data count lower register	JCDTCD	8	H'E801700D	8
	JPEG interrupt enable register 0	JINTE0	8	H'E801700E	8
	JPEG interrupt status register 0	JINTS0	8	H'E801700F	8
	JPEG code decode error register	JCDERR	8	H'E8017010	8
	JPEG code reset register	JCRST	8	H'E8017011	8
	JPEG interface compression control register	JIFECNT	32	H'E8017040	32
	JPEG interface compression source address register	JIFESA	32	H'E8017044	32
	JPEG interface compression line offset register	JIFESOFST	32	H'E8017048	32
	JPEG interface compression destination address register	JIFEDA	32	H'E801704C	32
	JPEG interface compression source line count register	JIFESLC	32	H'E8017050	32
	JPEG interface compression destination register	JIFEDDC	32	H'E8017054	32
	JPEG interface decompression control register	JIFDCNT	32	H'E8017058	32
	JPEG interface decompression source address register	JIFDSA	32	H'E801705C	32
	JPEG interface decompression destination offset register	JIFDDOFST	32	H'E8017060	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
JPEG codec unit*2	JPEG interface decompression destination address register	JIFDDA	32	H'E8017064	32
	JPEG interface decompression source count register	JIFDSDC	32	H'E8017068	32
	JPEG interface decompression destination line count register	JIFDDLC	32	H'E801706C	32
	JPEG interface decompression α setting register	JIFDADT	32	H'E8017070	32
	JPEG interrupt enable register 1	JINTE1	32	H'E801708C	32
	JPEG interrupt status register 1	JINTS1	32	H'E8017090	32
	JPEG input image data CbCr range setting register	JIFESVSZ	32	H'E8017094	32
	JPEG output image data CbCr range setting register	JIFESHSZ	32	H'E8017098	32
	JPEG code quantization table 0 register	JCQTBL0	512	H'E8017100 to H'E801713F	8
	JPEG code quantization table 1 register	JCQTBL1	512	H'E8017140 to H'E801717F	8
	JPEG code quantization table 2 register	JCQTBL2	512	H'E8017180 to H'E80171BF	8
	JPEG code quantization table 3 register	JCQTBL3	512	H'E80171C0 to H'E80171FF	8
	JPEG code Huffman table DC0 register	JCHTBD0	224	H'E8017200 to H'E801721B	8
	JPEG code Huffman table AC0 register	JCHTBA0	1416	H'E8017220 to H'E80172D1	8
	JPEG code Huffman table DC1 register	JCHTBD1	224	H'E8017300 to H'E801731B	8
	JPEG code Huffman table AC1 register	JCHTBA1	1416	H'E8017320 to H'E80173D1	8
EthernetAVB*2	AVB-DMAC mode register	CCC	32	H'E8215000	32
	Descriptor base address table register	DBAT	32	H'E8215004	32
	Descriptor base address load request register	DLR	32	H'E8215008	32
	AVB-DMAC status register	CSR	32	H'E821500C	32
	Current descriptor address register 0	CDAR0	32	H'E8215010	32
	Current descriptor address register 1	CDAR1	32	H'E8215014	32
	Current descriptor address register 2	CDAR2	32	H'E8215018	32
	Current descriptor address register 3	CDAR3	32	H'E821501C	32
	Current descriptor address register 4	CDAR4	32	H'E8215020	32
	Current descriptor address register 5	CDAR5	32	H'E8215024	32
	Current descriptor address register 6	CDAR6	32	H'E8215028	32
	Current descriptor address register 7	CDAR7	32	H'E821502C	32
	Current descriptor address register 8	CDAR8	32	H'E8215030	32
	Current descriptor address register 9	CDAR9	32	H'E8215034	32
	Current descriptor address register 10	CDAR10	32	H'E8215038	32
	Current descriptor address register 11	CDAR11	32	H'E821503C	32
	Current descriptor address register 12	CDAR12	32	H'E8215040	32
	Current descriptor address register 13	CDAR13	32	H'E8215044	32
	Current descriptor address register 14	CDAR14	32	H'E8215048	32
	Current descriptor address register 15	CDAR15	32	H'E821504C	32
	Current descriptor address register 16	CDAR16	32	H'E8215050	32
	Current descriptor address register 17	CDAR17	32	H'E8215054	32
	Current descriptor address register 18	CDAR18	32	H'E8215058	32
	Current descriptor address register 19	CDAR19	32	H'E821505C	32
	Current descriptor address register 20	CDAR20	32	H'E8215060	32
	Current descriptor address register 21	CDAR21	32	H'E8215064	32
	Error status register	ESR	32	H'E8215088	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
EthernetAVB*2	Receive configuration register	RCR	32	H'E8215090	32
	Receive queue configuration register 0	RQC0	32	H'E8215094	32
	Receive queue configuration register 1	RQC1	32	H'E8215098	32
	Receive queue configuration register 2	RQC2	32	H'E821509C	32
	Receive queue configuration register 3	RQC3	32	H'E82150A0	32
	Receive queue configuration register 4	RQC4	32	H'E82150A4	32
	Receive padding configuration register	RPC	32	H'E82150B0	32
	Unread frame counter stop level register	UFCS	32	H'E82150C0	32
	Unread frame counter register 0	UFCV0	32	H'E82150C4	32
	Unread frame counter register 1	UFCV1	32	H'E82150C8	32
	Unread frame counter register 2	UFCV2	32	H'E82150CC	32
	Unread frame counter register 3	UFCV3	32	H'E82150D0	32
	Unread frame counter register 4	UFCV4	32	H'E82150D4	32
	Unread frame counter decrement register 0	UFCD0	32	H'E82150E0	32
	Unread frame counter decrement register 1	UFCD1	32	H'E82150E4	32
	Unread frame counter decrement register 2	UFCD2	32	H'E82150E8	32
	Unread frame counter decrement register 3	UFCD3	32	H'E82150EC	32
	Unread frame counter decrement register 4	UFCD4	32	H'E82150F0	32
	Separation filter offset register	SFO	32	H'E82150FC	32
	Separation filter pattern register 0	SFP0	32	H'E8215100	32
	Separation filter pattern register 1	SFP1	32	H'E8215104	32
	Separation filter pattern register 2	SFP2	32	H'E8215108	32
	Separation filter pattern register 3	SFP3	32	H'E821510C	32
	Separation filter pattern register 4	SFP4	32	H'E8215110	32
	Separation filter pattern register 5	SFP5	32	H'E8215114	32
	Separation filter pattern register 6	SFP6	32	H'E8215118	32
	Separation filter pattern register 7	SFP7	32	H'E821511C	32
	Separation filter pattern register 8	SFP8	32	H'E8215120	32
	Separation filter pattern register 9	SFP9	32	H'E8215124	32
	Separation filter pattern register 10	SFP10	32	H'E8215128	32
	Separation filter pattern register 11	SFP11	32	H'E821512C	32
	Separation filter pattern register 12	SFP12	32	H'E8215130	32
	Separation filter pattern register 13	SFP13	32	H'E8215134	32
	Separation filter pattern register 14	SFP14	32	H'E8215138	32
	Separation filter pattern register 15	SFP15	32	H'E821513C	32
	Separation filter pattern register 16	SFP16	32	H'E8215140	32
	Separation filter pattern register 17	SFP17	32	H'E8215144	32
	Separation filter pattern register 18	SFP18	32	H'E8215148	32
	Separation filter pattern register 19	SFP19	32	H'E821514C	32
	Separation filter pattern register 20	SFP20	32	H'E8215150	32
	Separation filter pattern register 21	SFP21	32	H'E8215154	32
	Separation filter pattern register 22	SFP22	32	H'E8215158	32
	Separation filter pattern register 23	SFP23	32	H'E821515C	32
	Separation filter pattern register 24	SFP24	32	H'E8215160	32
	Separation filter pattern register 25	SFP25	32	H'E8215164	32
	Separation filter pattern register 26	SFP26	32	H'E8215168	32
	Separation filter pattern register 27	SFP27	32	H'E821516C	32
	Separation filter pattern register 28	SFP28	32	H'E8215170	32
	Separation filter pattern register 29	SFP29	32	H'E8215174	32
	Separation filter pattern register 30	SFP30	32	H'E8215178	32
	Separation filter pattern register 31	SFP31	32	H'E821517C	32
	Separation filter mask register 0	SFM0	32	H'E82151C0	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
EthernetAVB*2	Separation filter mask register 1	SFM1	32	H'E82151C4	32
	Transmit configuration register	TGC	32	H'E8215300	32
	Transmit configuration control register	TCCR	32	H'E8215304	32
	Transmit status register	TSR	32	H'E8215308	32
	Time stamp FIFO access register 0	TFA0	32	H'E8215310	32
	Time stamp FIFO access register 1	TFA1	32	H'E8215314	32
	Time stamp FIFO access register 2	TFA2	32	H'E8215318	32
	CBS increment value register 0	CIVR0	32	H'E8215320	32
	CBS increment value register 1	CIVR1	32	H'E8215324	32
	CBS decrement value register 0	CDVR0	32	H'E8215328	32
	CBS decrement value register 1	CDVR1	32	H'E821532C	32
	CBS upper limit register 0	CUL0	32	H'E8215330	32
	CBS upper limit register 1	CUL1	32	H'E8215334	32
	CBS lower limit register 0	CLL0	32	H'E8215338	32
	CBS lower limit register 1	CLL1	32	H'E821533C	32
	Descriptor interrupt control register	DIC	32	H'E8215350	32
	Descriptor interrupt status register	DIS	32	H'E8215354	32
	Error interrupt control register	EIC	32	H'E8215358	32
	Error interrupt status register	EIS	32	H'E821535C	32
	Receive interrupt control register 0	RIC0	32	H'E8215360	32
	Receive interrupt status register 0	RIS0	32	H'E8215364	32
	Receive interrupt control register 1	RIC1	32	H'E8215368	32
	Receive interrupt status register 1	RIS1	32	H'E821536C	32
	Receive interrupt control register 2	RIC2	32	H'E8215370	32
	Receive interrupt status register 2	RIS2	32	H'E8215374	32
	Transmit interrupt control register	TIC	32	H'E8215378	32
	Transmit interrupt status register	TIS	32	H'E821537C	32
	Interrupt summary status register	ISS	32	H'E8215380	32
	gPTP configuration control register	GCCR	32	H'E8215390	32
	gPTP maximum transit time register	GMTT	32	H'E8215394	32
	gPTP presentation time comparison register	GPTC	32	H'E8215398	32
	gPTP timer increment register	GTI	32	H'E821539C	32
	gPTP timer offset configuration register 0	GTO0	32	H'E82153A0	32
	gPTP timer offset configuration register 1	GTO1	32	H'E82153A4	32
	gPTP timer offset configuration register 2	GTO2	32	H'E82153A8	32
	gPTP interrupt control register	GIC	32	H'E82153AC	32
	gPTP interrupt status register	GIS	32	H'E82153B0	32
	gPTP presentation time capture register	GCPT	32	H'E82153B4	32
	gPTP timer capture register 0	GCT0	32	H'E82153B8	32
	gPTP timer capture register 1	GCT1	32	H'E82153BC	32
	gPTP timer capture register 2	GCT2	32	H'E82153C0	32
	gPTP capture event control register	GCEC	32	H'E82153D8	32
	E-MAC mode register	ECMR	32	H'E8215500	32
	Receive frame length register	RFLR	32	H'E8215508	32
	E-MAC status register	ECSR	32	H'E8215510	32
	E-MAC interrupt permission register	ECSIPR	32	H'E8215518	32
	PHY interface register	PIR	32	H'E8215520	32
	Automatic PAUSE frame register	APR	32	H'E8215554	32
	Manual PAUSE frame register	MPR	32	H'E8215558	32
	PAUSE frame transmit counter	PFTCR	32	H'E821555C	32
	PAUSE frame receive counter	PFRCR	32	H'E8215560	32
	Automatic PAUSE frame retransmission count register	TPAUSER	32	H'E8215564	32

chanhvinh.com

Table 46.1 Register Addresses

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
EthernetAVB*2	MAC address high register	MAHR	32	H'E82155C0	32
	MAC address low register	MALR	32	H'E82155C8	32
	CRC error frame receive counter register	CEFCR	32	H'E8215740	32
	Frame receive error counter register	FRECR	32	H'E8215748	32
	Too-short frame receive counter register	TSFRCR	32	H'E8215750	32
	Too-long frame receive counter register	TLFRCR	32	H'E8215758	32
	Residual-bit frame receive counter register	RFCR	32	H'E8215760	32
	Multicast address frame receive counter register	MAFCR	32	H'E8215778	32

Note 1. RZ/A1L only
Note 2. RZ/A1LU only

chanhvinh.com

46.2 Register Bits

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Secondary cache	reg0_cache_id	Implementer[7]	Implementer[6]	Implementer[5]	Implementer[4]	Implementer[3]	Implementer[2]	Implementer[1]	Implementer[0]
		-	-	-	-	-	-	-	-
		CACHE ID[5]	CACHE ID[4]	CACHE ID[3]	CACHE ID[2]	CACHE ID[1]	CACHE ID[0]	Part Number[3]	Part Number[2]
		Part Number[1]	Part Number[0]	RTL release[5]	RTL release[4]	RTL release[3]	RTL release[2]	RTL release[1]	RTL release[0]
	reg0_cache_type	Data banking	-	-	ctype[3]	ctype[2]	ctype[1]	ctype[0]	-
		Dsize[4]	Dsize[3]	Dsize[2]	Dsize[1]	Dsize[0]	L2 associativity	-	-
		-	-	L2 cache line length[1]	L2 cache line length[0]	Isize[4]	Isize[3]	Isize[2]	Isize[1]
		Isize[0]	L2 associativity	-	-	-	-	L2 cache line length[1]	L2 cache line length[0]
	reg1_control	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	L2 Cache enable
	reg1_aux_control	-	Early BRESP enable	Instruction prefetch enable	Data prefetch enable	Non-secure interrupt access control	Non-secure lockdown enable	Cache replacement policy	Force write allocate[1]
		Force write allocate[0]	Shared attribute override enable	Parity enable	Event monitor bus enable	Way-size[2]	Way-size[1]	Way-size[0]	Associativity
		-	-	Shared Attribute Invalidate Enable	Exclusive cache configuration	Store buffer device limitation Enable	High Priority for SO and Dev Reads Enable	-	-
		-	-	-	-	-	-	-	Full Line of Zero Enable
	reg1_tag_ram_ control	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	RAM write access latency[2]	RAM write access latency[1]	RAM write access latency[0]
		0	RAM read access latency[2]	RAM read access latency[1]	RAM read access latency[0]	0	RAM setup latency[2]	RAM setup latency[1]	RAM setup latency[0]
	reg1_data_ram_ control	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	RAM write access latency[2]	RAM write access latency[1]	RAM write access latency[0]
		0	RAM read access latency[2]	RAM read access latency[1]	RAM read access latency[0]	0	RAM setup latency [2]	RAM setup latency [1]	RAM setup latency [0]
	reg2_ev_counter_ ctrl	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	Counter reset[1]	Counter reset[0]	Event counter enable
	reg2_ev_counter1_ cfg	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	Counter event source[3]	Counter event source[2]	Counter event source[1]	Counter event source[0]	Event counter interrupt generation[1]	Event counter interrupt generation[0]
	reg2_ev_counter0_ cfg	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	Counter event source[3]	Counter event source[2]	Counter event source[1]	Counter event source[0]	Event counter interrupt generation[1]	Event counter interrupt generation[0]
	reg2_ev_counter1	Counter value[31]	Counter value[30]	Counter value[29]	Counter value[28]	Counter value[27]	Counter value[26]	Counter value[25]	Counter value[24]
		Counter value[23]	Counter value[22]	Counter value[21]	Counter value[20]	Counter value[19]	Counter value[18]	Counter value[17]	Counter value[16]
		Counter value[15]	Counter value[14]	Counter value[13]	Counter value[12]	Counter value[11]	Counter value[10]	Counter value[9]	Counter value[8]
		Counter value[7]	Counter value[6]	Counter value[5]	Counter value[4]	Counter value[3]	Counter value[2]	Counter value[1]	Counter value[0]
	reg2_ev_counter0	Counter value[31]	Counter value[30]	Counter value[29]	Counter value[28]	Counter value[27]	Counter value[26]	Counter value[25]	Counter value[24]
		Counter value[23]	Counter value[22]	Counter value[21]	Counter value[20]	Counter value[19]	Counter value[18]	Counter value[17]	Counter value[16]
		Counter value[15]	Counter value[14]	Counter value[13]	Counter value[12]	Counter value[11]	Counter value[10]	Counter value[9]	Counter value[8]
		Counter value[7]	Counter value[6]	Counter value[5]	Counter value[4]	Counter value[3]	Counter value[2]	Counter value[1]	Counter value[0]
	reg2_int_mask	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	DECERR
		SLVERR	ERRRD	ERRRT	ERRWD	ERRWT	PARRD	PARRT	ECNTR

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Secondary cache	reg2_int_mask_ status	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	DECERR
		SLVERR	ERRRD	ERRRT	ERRWD	ERRWT	PARRD	PARRT	ECNTR
	reg2_int_raw_status	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	DECERR
		SLVERR	ERRRD	ERRRT	ERRWD	ERRWT	PARRD	PARRT	ECNTR
	reg2_int_clear	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	DECERR
		SLVERR	ERRRD	ERRRT	ERRWD	ERRWT	PARRD	PARRT	ECNTR
	reg7_cache_sync	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	C
	reg7_inv_pa	TAG[17]	TAG[16]	TAG[15]	TAG[14]	TAG[13]	TAG[12]	TAG[11]	TAG[10]
		TAG[9]	TAG[8]	TAG[7]	TAG[6]	TAG[5]	TAG[4]	TAG[3]	TAG[2]
		TAG[1]	TAG[0]	INDEX[8]	INDEX[7]	INDEX[6]	INDEX[5]	INDEX[4]	INDEX[3]
		INDEX[2]	INDEX[1]	INDEX[0]	-	-	-	-	C
	reg7_inv_way	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Way bits[7]	Way bits[6]	Way bits[5]	Way bits[4]	Way bits[3]	Way bits[2]	Way bits[1]	Way bits[0]
	reg7_clean_pa	TAG[17]	TAG[16]	TAG[15]	TAG[14]	TAG[13]	TAG[12]	TAG[11]	TAG[10]
		TAG[9]	TAG[8]	TAG[7]	TAG[6]	TAG[5]	TAG[4]	TAG[3]	TAG[2]
		TAG[1]	TAG[0]	INDEX[8]	INDEX[7]	INDEX[6]	INDEX[5]	INDEX[4]	INDEX[3]
		INDEX[2]	INDEX[1]	INDEX[0]	-	-	-	-	C
	reg7_clean_index	-	Way[2]	Way[1]	Way[0]	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	INDEX[8]	INDEX[7]	INDEX[6]	INDEX[5]	INDEX[4]	INDEX[3]
		INDEX[2]	INDEX[1]	INDEX[0]	-	-	-	-	C
	reg7_clean_way	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Way bits[7]	Way bits[6]	Way bits[5]	Way bits[4]	Way bits[3]	Way bits[2]	Way bits[1]	Way bits[0]
	reg7_clean_inv_pa	TAG[17]	TAG[16]	TAG[15]	TAG[14]	TAG[13]	TAG[12]	TAG[11]	TAG[10]
		TAG[9]	TAG[8]	TAG[7]	TAG[6]	TAG[5]	TAG[4]	TAG[3]	TAG[2]
		TAG[1]	TAG[0]	INDEX[8]	INDEX[7]	INDEX[6]	INDEX[5]	INDEX[4]	INDEX[3]
		INDEX[2]	INDEX[1]	INDEX[0]	-	-	-	-	C
	reg7_clean_inv_ index	-	Way[2]	Way[1]	Way[0]	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	INDEX[8]	INDEX[7]	INDEX[6]	INDEX[5]	INDEX[4]	INDEX[3]
		INDEX[2]	INDEX[1]	INDEX[0]	-	-	-	-	C
	reg7_clean_inv_way	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Way_bits[7]	Way_bits[6]	Way_bits[5]	Way_bits[4]	Way_bits[3]	Way_bits[2]	Way_bits[1]	Way_bits[0]
	reg9_d_lockdown0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DATALOCK000[7]	DATALOCK000[6]	DATALOCK000[5]	DATALOCK000[4]	DATALOCK000[3]	DATALOCK000[2]	DATALOCK000[1]	DATALOCK000[0]
	reg9_i_lockdown0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		INSTRLOCK000 [7]	INSTRLOCK000 [6]	INSTRLOCK000 [5]	INSTRLOCK000 [4]	INSTRLOCK000 [3]	INSTRLOCK000 [2]	INSTRLOCK000 [1]	INSTRLOCK000 [0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Secondary cache	reg9_d_lockdown1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DATALOCK001[7]	DATALOCK001[6]	DATALOCK001[5]	DATALOCK001[4]	DATALOCK001[3]	DATALOCK001[2]	DATALOCK001[1]	DATALOCK001[0]
	reg9_i_lockdown1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		INSTRLOCK001 [7]	INSTRLOCK001 [6]	INSTRLOCK001 [5]	INSTRLOCK001 [4]	INSTRLOCK001 [3]	INSTRLOCK001 [2]	INSTRLOCK001 [1]	INSTRLOCK001 [0]
	reg9_d_lockdown2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DATALOCK002[7]	DATALOCK002[6]	DATALOCK002[5]	DATALOCK002[4]	DATALOCK002[3]	DATALOCK002[2]	DATALOCK002[1]	DATALOCK002[0]
	reg9_i_lockdown2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		INSTRLOCK002 [7]	INSTRLOCK002 [6]	INSTRLOCK002 [5]	INSTRLOCK002 [4]	INSTRLOCK002 [3]	INSTRLOCK002 [2]	INSTRLOCK002 [1]	INSTRLOCK002 [0]
	reg9_d_lockdown3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DATALOCK003[7]	DATALOCK003[6]	DATALOCK003[5]	DATALOCK003[4]	DATALOCK003[3]	DATALOCK003[2]	DATALOCK003[1]	DATALOCK003[0]
	reg9_i_lockdown3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		INSTRLOCK003 [7]	INSTRLOCK003 [6]	INSTRLOCK003 [5]	INSTRLOCK003 [4]	INSTRLOCK003 [3]	INSTRLOCK003 [2]	INSTRLOCK003 [1]	INSTRLOCK003 [0]
	reg9_d_lockdown4	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DATALOCK004[7]	DATALOCK004[6]	DATALOCK004[5]	DATALOCK004[4]	DATALOCK004[3]	DATALOCK004[2]	DATALOCK004[1]	DATALOCK004[0]
	reg9_i_lockdown4	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		INSTRLOCK004 [7]	INSTRLOCK004[6]	INSTRLOCK004 [5]	INSTRLOCK004 [4]	INSTRLOCK004 [3]	INSTRLOCK004 [2]	INSTRLOCK004 [1]	INSTRLOCK004 [0]
	reg9_d_lockdown5	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DATALOCK005[7]	DATALOCK005[6]	DATALOCK005[5]	DATALOCK005[4]	DATALOCK005[3]	DATALOCK005[2]	DATALOCK005[1]	DATALOCK005[0]
	reg9_i_lockdown5	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		INSTRLOCK005 [7]	INSTRLOCK005 [6]	INSTRLOCK005 [5]	INSTRLOCK005 [4]	INSTRLOCK005 [3]	INSTRLOCK005 [2]	INSTRLOCK005 [1]	INSTRLOCK005 [0]
	reg9_d_lockdown6	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DATALOCK006[7]	DATALOCK006[6]	DATALOCK006[5]	DATALOCK006[4]	DATALOCK006[3]	DATALOCK006[2]	DATALOCK006[1]	DATALOCK006[0]
	reg9_i_lockdown6	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		INSTRLOCK006 [7]	INSTRLOCK006 [6]	INSTRLOCK006 [5]	INSTRLOCK006 [4]	INSTRLOCK006 [3]	INSTRLOCK006 [2]	INSTRLOCK006 [1]	INSTRLOCK006 [0]
	reg9_d_lockdown7	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DATALOCK007[7]	DATALOCK007[6]	DATALOCK007[5]	DATALOCK007[4]	DATALOCK007[3]	DATALOCK007[2]	DATALOCK007[1]	DATALOCK007[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Secondary cache	reg9_i_lockdown7	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		INSTRLOCK007 [7]	INSTRLOCK007 [6]	INSTRLOCK007 [5]	INSTRLOCK007 [4]	INSTRLOCK007 [3]	INSTRLOCK007 [2]	INSTRLOCK007 [1]	INSTRLOCK007 [0]
	reg9_lock_line_en	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	lockdown_by_line_ enable
	reg9_unlock_way	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		unlock_all_lines_ by_way_operation[7]	unlock_all_lines_ by_way_operation[6]	unlock_all_lines_ by_way_operation[5]	unlock_all_lines_ by_way_operation[4]	unlock_all_lines_ by_way_operation[3]	unlock_all_lines_ by_way_operation[2]	unlock_all_lines_ by_way_operation[1]	unlock_all_lines_ by_way_operation[0]
	reg12_addr_filtering_ start	address_filtering_start [11]	address_filtering_start [10]	address_filtering_start [9]	address_filtering_start [8]	address_filtering_start [7]	address_filtering_start [6]	address_filtering_start [5]	address_filtering_start [4]
		address_filtering_start [3]	address_filtering_start [2]	address_filtering_start [1]	address_filtering_start [0]	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	address_filtering_ enable
	reg12_addr_filtering_ end	address_filtering_end [11]	address_filtering_end [10]	address_filtering_end [9]	address_filtering_end [8]	address_filtering_end [7]	address_filtering_end [6]	address_filtering_end [5]	address_filtering_end [4]
		address_filtering_end [3]	address_filtering_end [2]	address_filtering_end [1]	address_filtering_end [0]	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	reg15_debug_ctrl	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	SPNIDEN	DWB	DCL
	reg15_prefetch_ctrl	-	Double linefill enable	Instruction prefetch enable	Data prefetch enable	Double linefill on WRAP read disable	-	-	Prefetch drop enable
		Incr double Linefill enable	-	Not same ID on exclusive sequence enable	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	Prefetch offset[4]	Prefetch offset[3]	Prefetch offset[2]	Prefetch offset[1]	Prefetch offset[0]
	reg15_power_ctrl	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	dynamic_clk_ gating_en	standby_mode_en
LSI internal bus	RMPR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	AXI128	-
	AXIBUSCTL0	-	-	-	-	JCUARCACHE[3]*4	JCUARCACHE[2]*4	JCUARCACHE[1]*4	JCUARCACHE[0]*4
		-	-	-	-	JCUAWCACHE[3]*4	JCUAWCACHE[2]*4	JCUAWCACHE[1]*4	JCUAWCACHE[0]*4
		-	-	-	-	ETHARCACHE[3]	ETHARCACHE[2]	ETHARCACHE[1]	ETHARCACHE[0]
		-	-	-	-	ETHAWCACHE[3]	ETHAWCACHE[2]	ETHAWCACHE[1]	ETHAWCACHE[0]
	AXIBUSCTL2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CEUAWCACHE [3]	CEUAWCACHE [2]	CEUAWCACHE [1]	CEUAWCACHE [0]
	AXIBUSCTL5*1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	MLBAxCACHE[1]	MLBAxCACHE[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
LSI internal bus	AXIBUSCTL6	-	-	-	-	VDC501AR CACHE[3]	VDC501AR CACHE[2]	VDC501AR CACHE[1]	VDC501AR CACHE[0]
		-	-	-	-	VDC501AW CACHE[3]	VDC501AW CACHE[2]	VDC501AW CACHE[1]	VDC501AW CACHE[0]
		-	-	-	-	VDC502AR CACHE[3]	VDC502AR CACHE[2]	VDC502AR CACHE[1]	VDC502AR CACHE[0]
		-	-	-	-	-	-	-	-
	AXIBUSCTL7	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	VDC504AR CACHE[3]	VDC504AR CACHE[2]	VDC504AR CACHE[1]	VDC504AR CACHE[0]
		-	-	-	-	-	-	-	-
	AXIRERRCTL0	-	-	-	JCURERREN*4	-	-	-	ETHRERREN
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	CEURERREN
		-	-	-	-	-	-	-	-
	AXIRERRCTL2	-	-	-	VDC501RERREN	-	-	-	VDC502RERREN
		-	-	-	-	-	-	-	VDC504RERREN
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	AXIRERRST0	JCURRESP[1]*4	JCURRESP[0]*4	JCUBRESP[1]*4	JCUBRESP[0]*4	ETHRRESP[1]	ETHRRESP[0]	ETHBRESP[1]	ETHBRESP[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	CEUBRESP[1]	CEUBRESP[0]
		-	-	-	-	-	-	-	-
	AXIRERRST2	VDC501RRESP [1]	VDC501RRESP [0]	VDC501BRESP [1]	VDC501BRESP [0]	VDC502RRESP [1]	VDC502RRESP [0]	-	-
		-	-	-	-	VDC504RRESP [1]	VDC504RRESP [0]	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	AXIRERRCLR0	-	JCURRESPCLR*4	-	JCUBRESPCLR*4	-	ETHRRESPCLR	-	ETHBRESPCLR
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	CEUBRESPCLR
		-	-	-	-	-	-	-	-
	AXIRERRCLR2	-	VDC501RRESP CLR	-	VDC501BRESP CLR	-	VDC502RRESP CLR	-	-
		-	-	-	-	-	VDC504RRESP CLR	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
Clock pulse generator	FRQCR	-	CKOEN2	CKOEN[1]	CKOEN[0]	-	-	IFC[1]	IFC[0]
		-	-	-	-	-	-	-	-
Interrupt controller	ICR0	NMIL	-	-	-	-	-	-	NMIE
		-	-	-	-	-	-	NMIF	-
	ICR1	IRQ71S	IRQ70S	IRQ61S	IRQ60S	IRQ51S	IRQ50S	IRQ41S	IRQ40S
		IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S
	IRQRR	-	-	-	-	-	-	-	-
		IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
	ICDDCR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	Enable
	ICDICTR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		LSPI[4]	LSPI[3]	LSPI[2]	LSPI[1]	LSPI[0]	SecurityExtn	-	-
		CPUNumber[2]	CPUNumber[1]	CPUNumber[0]	ITLinesNumber[4]	ITLinesNumber[3]	ITLinesNumber[2]	ITLinesNumber[1]	ITLinesNumber[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Interrupt controller	ICDIIDR	ProductID[7]	ProductID[6]	ProductID[5]	ProductID[4]	ProductID[3]	ProductID[2]	ProductID[1]	ProductID[0]
		-	-	-	-	Variant[3]	Variant[2]	Variant[1]	Variant[0]
		Revision[3]	Revision[2]	Revision[1]	Revision[0]	Implementer[11]	Implementer[10]	Implementer[9]	Implementer[8]
		Implementer[7]	Implementer[6]	Implementer[5]	Implementer[4]	Implementer[3]	Implementer[2]	Implementer[1]	Implementer[0]
	ICDISRn n = 0 to 16	Security status bits[31]	Security status bits[30]	Security status bits[29]	Security status bits[28]	Security status bits[27]	Security status bits[26]	Security status bits[25]	Security status bits[24]
		Security status bits[23]	Security status bits[22]	Security status bits[21]	Security status bits[20]	Security status bits[19]	Security status bits[18]	Security status bits[17]	Security status bits[16]
		Security status bits[15]	Security status bits[14]	Security status bits[13]	Security status bits[12]	Security status bits[11]	Security status bits[10]	Security status bits[9]	Security status bits[8]
		Security status bits[7]	Security status bits[6]	Security status bits[5]	Security status bits[4]	Security status bits[3]	Security status bits[2]	Security status bits[1]	Security status bits[0]
	ICDISERn n = 0 to 16	Set-enable bits[31]	Set-enable bits[30]	Set-enable bits[29]	Set-enable bits[28]	Set-enable bits[27]	Set-enable bits[26]	Set-enable bits[25]	Set-enable bits[24]
		Set-enable bits[23]	Set-enable bits[22]	Set-enable bits[21]	Set-enable bits[20]	Set-enable bits[19]	Set-enable bits[18]	Set-enable bits[17]	Set-enable bits[16]
		Set-enable bits[15]	Set-enable bits[14]	Set-enable bits[13]	Set-enable bits[12]	Set-enable bits[11]	Set-enable bits[10]	Set-enable bits[9]	Set-enable bits[8]
		Set-enable bits[7]	Set-enable bits[6]	Set-enable bits[5]	Set-enable bits[4]	Set-enable bits[3]	Set-enable bits[2]	Set-enable bits[1]	Set-enable bits[0]
	ICDICERn n = 0 to 16	Clear-enable bits[31]	Clear-enable bits[30]	Clear-enable bits[29]	Clear-enable bits[28]	Clear-enable bits[27]	Clear-enable bits[26]	Clear-enable bits[25]	Clear-enable bits[24]
		Clear-enable bits[23]	Clear-enable bits[22]	Clear-enable bits[21]	Clear-enable bits[20]	Clear-enable bits[19]	Clear-enable bits[18]	Clear-enable bits[17]	Clear-enable bits[16]
		Clear-enable bits[15]	Clear-enable bits[14]	Clear-enable bits[13]	Clear-enable bits[12]	Clear-enable bits[11]	Clear-enable bits[10]	Clear-enable bits[9]	Clear-enable bits[8]
		Clear-enable bits[7]	Clear-enable bits[6]	Clear-enable bits[5]	Clear-enable bits[4]	Clear-enable bits[3]	Clear-enable bits[2]	Clear-enable bits[1]	Clear-enable bits[0]
	ICDISPRn n = 0 to 16	Set-pending bits[31]	Set-pending bits[30]	Set-pending bits[29]	Set-pending bits[28]	Set-pending bits[27]	Set-pending bits[26]	Set-pending bits[25]	Set-pending bits[24]
		Set-pending bits[23]	Set-pending bits[22]	Set-pending bits[21]	Set-pending bits[20]	Set-pending bits[19]	Set-pending bits[18]	Set-pending bits[17]	Set-pending bits[16]
		Set-pending bits[15]	Set-pending bits[14]	Set-pending bits[13]	Set-pending bits[12]	Set-pending bits[11]	Set-pending bits[10]	Set-pending bits[9]	Set-pending bits[8]
		Set-pending bits[7]	Set-pending bits[6]	Set-pending bits[5]	Set-pending bits[4]	Set-pending bits[3]	Set-pending bits[2]	Set-pending bits[1]	Set-pending bits[0]
	ICDICPRn n = 0 to 16	Clear-pending bits[31]	Clear-pending bits[30]	Clear-pending bits[29]	Clear-pending bits[28]	Clear-pending bits[27]	Clear-pending bits[26]	Clear-pending bits[25]	Clear-pending bits[24]
		Clear-pending bits[23]	Clear-pending bits[22]	Clear-pending bits[21]	Clear-pending bits[20]	Clear-pending bits[19]	Clear-pending bits[18]	Clear-pending bits[17]	Clear-pending bits[16]
		Clear-pending bits[15]	Clear-pending bits[14]	Clear-pending bits[13]	Clear-pending bits[12]	Clear-pending bits[11]	Clear-pending bits[10]	Clear-pending bits[9]	Clear-pending bits[8]
		Clear-pending bits[7]	Clear-pending bits[6]	Clear-pending bits[5]	Clear-pending bits[4]	Clear-pending bits[3]	Clear-pending bits[2]	Clear-pending bits[1]	Clear-pending bits[0]
	ICDABRn n = 0 to 16	Active bits[31]	Active bits[30]	Active bits[29]	Active bits[28]	Active bits[27]	Active bits[26]	Active bits[25]	Active bits[24]
		Active bits[23]	Active bits[22]	Active bits[21]	Active bits[20]	Active bits[19]	Active bits[18]	Active bits[17]	Active bits[16]
		Active bits[15]	Active bits[14]	Active bits[13]	Active bits[12]	Active bits[11]	Active bits[10]	Active bits[9]	Active bits[8]
		Active bits[7]	Active bits[6]	Active bits[5]	Active bits[4]	Active bits[3]	Active bits[2]	Active bits[1]	Active bits[0]
	ICDIPRn n = 0 to 134	Priority, byte offset 3[7]	Priority, byte offset 3[6]	Priority, byte offset 3[5]	Priority, byte offset 3[4]	Priority, byte offset 3[3]	Priority, byte offset 3[2]	Priority, byte offset 3[1]	Priority, byte offset 3[0]
		Priority, byte offset 2[7]	Priority, byte offset 2[6]	Priority, byte offset 2[5]	Priority, byte offset 2[4]	Priority, byte offset 2[3]	Priority, byte offset 2[2]	Priority, byte offset 2[1]	Priority, byte offset 2[0]
		Priority, byte offset 1[7]	Priority, byte offset 1[6]	Priority, byte offset 1[5]	Priority, byte offset 1[4]	Priority, byte offset 1[3]	Priority, byte offset 1[2]	Priority, byte offset 1[1]	Priority, byte offset 1[0]
		Priority, byte offset 0[7]	Priority, byte offset 0[6]	Priority, byte offset 0[5]	Priority, byte offset 0[4]	Priority, byte offset 0[3]	Priority, byte offset 0[2]	Priority, byte offset 0[1]	Priority, byte offset 0[0]
	ICDIPTRn n = 0 to 134	CPU targets, byte offset 3[7]	CPU targets, byte offset 3[6]	CPU targets, byte offset 3[5]	CPU targets, byte offset 3[4]	CPU targets, byte offset 3[3]	CPU targets, byte offset 3[2]	CPU targets, byte offset 3[1]	CPU targets, byte offset 3[0]
		CPU targets, byte offset 2[7]	CPU targets, byte offset 2[6]	CPU targets, byte offset 2[5]	CPU targets, byte offset 2[4]	CPU targets, byte offset 2[3]	CPU targets, byte offset 2[2]	CPU targets, byte offset 2[1]	CPU targets, byte offset 2[0]
		CPU targets, byte offset 1[7]	CPU targets, byte offset 1[6]	CPU targets, byte offset 1[5]	CPU targets, byte offset 1[4]	CPU targets, byte offset 1[3]	CPU targets, byte offset 1[2]	CPU targets, byte offset 1[1]	CPU targets, byte offset 1[0]
		CPU targets, byte offset 0[7]	CPU targets, byte offset 0[6]	CPU targets, byte offset 0[5]	CPU targets, byte offset 0[4]	CPU targets, byte offset 0[3]	CPU targets, byte offset 0[2]	CPU targets, byte offset 0[1]	CPU targets, byte offset 0[0]
	ICDICFRn n = 0 to 33	Int_config[1], field 15	Int_config[0], field 15	Int_config[1], field 14	Int_config[0], field 14	Int_config[1], field 13	Int_config[0], field 13	Int_config[1], field 12	Int_config[0], field 12
		Int_config[1], field 11	Int_config[0], field 11	Int_config[1], field 10	Int_config[0], field 10	Int_config[1], field 9	Int_config[0], field 9	Int_config[1], field 8	Int_config[0], field 8
		Int_config[1],field7	Int_config[0],field7	Int_config[1],field6	Int_config[0],field6	Int_config[1],field5	Int_config[0],field5	Int_config[1],field4	Int_config[0],field4
		Int_config[1],field3	Int_config[0],field3	Int_config[1],field2	Int_config[0],field2	Int_config[1],field1	Int_config[0],field1	Int_config[1],field0	Int_config[0],field0
	PPI Status Register	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ppi_status[15]	ppi_status[14]	ppi_status[13]	ppi_status[12]	ppi_status[11]	ppi_status[10]	ppi_status[9]	ppi_status[8]
		ppi_status[7]	ppi_status[6]	ppi_status[5]	ppi_status[4]	ppi_status[3]	ppi_status[2]	ppi_status[1]	ppi_status[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Interrupt controller	SPI Status Registersn n = 0 to 14	spi_status[31]	spi_status[30]	spi_status[29]	spi_status[28]	spi_status[27]	spi_status[26]	spi_status[25]	spi_status[24]
		spi_status[23]	spi_status[22]	spi_status[21]	spi_status[20]	spi_status[19]	spi_status[18]	spi_status[17]	spi_status[16]
		spi_status[15]	spi_status[14]	spi_status[13]	spi_status[12]	spi_status[11]	spi_status[10]	spi_status[9]	spi_status[8]
		spi_status[7]	spi_status[6]	spi_status[5]	spi_status[4]	spi_status[3]	spi_status[2]	spi_status[1]	spi_status[0]
	ICDSGIR	-	-	-	-	-	-	TargetListFilter[1]	TargetListFilter[0]
		CPUTargetList[7]	CPUTargetList[6]	CPUTargetList[5]	CPUTargetList[4]	CPUTargetList[3]	CPUTargetList[2]	CPUTargetList[1]	CPUTargetList[0]
		SATT	-	-	-	-	-	-	-
		-	-	-	-	SGIINTID[3]	SGIINTID[2]	SGIINTID[1]	SGIINTID[0]
	ICCICR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	SBPR	FIQEn	AckCtl	EnableNS	EnableS
	ICCPMR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Priority[7]	Priority[6]	Priority[5]	Priority[4]	Priority[3]	Priority[2]	Priority[1]	Priority[0]
	ICCBPR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	Binary point[2]	Binary point[1]	Binary point[0]
	ICCIAR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	CPUID[2]	CPUID[1]	CPUID[0]	ACKINTID[9]	ACKINTID[8]
		ACKINTID[7]	ACKINTID[6]	ACKINTID[5]	ACKINTID[4]	ACKINTID[3]	ACKINTID[2]	ACKINTID[1]	ACKINTID[0]
	ICCEOIR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	CPUID[2]	CPUID[1]	CPUID[0]	EOIINTID[9]	EOIINTID[8]
		EOIINTID[7]	EOIINTID[6]	EOIINTID[5]	EOIINTID[4]	EOIINTID[3]	EOIINTID[2]	EOIINTID[1]	EOIINTID[0]
	ICCRPR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Priority[7]	Priority[6]	Priority[5]	Priority[4]	Priority[3]	Priority[2]	Priority[1]	Priority[0]
	ICCHPIR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	CPUID[2]	CPUID[1]	CPUID[0]	PENDINTID[9]	PENDINTID[8]
		PENDINTID[7]	PENDINTID[6]	PENDINTID[5]	PENDINTID[4]	PENDINTID[3]	PENDINTID[2]	PENDINTID[1]	PENDINTID[0]
	ICCABPR								
	ICCIIDR	ProductID[11]	ProductID[10]	ProductID[9]	ProductID[8]	ProductID[7]	ProductID[6]	ProductID[5]	ProductID[4]
		ProductID[3]	ProductID[2]	ProductID[1]	ProductID[0]	Architectureversion[3]	Architectureversion[2]	Architectureversion[1]	Architectureversion[0]
		Revision[3]	Revision[2]	Revision[1]	Revision[0]	Implementer[11]	Implementer[10]	Implementer[9]	Implementer[8]
		Implementer[7]	Implementer[6]	Implementer[5]	Implementer[4]	Implementer[3]	Implementer[2]	Implementer[1]	Implementer[0]
Bus state controller	CMNCR	-	-	-	TL0	-	-	-	AL0
		-	-	-	-	-	-	-	-
		-	-	-	-	-	DPRTY[1]	DPRTY[0]	-
		-	-	-	-	-	-	HIZMEM	HIZCNT
	CS0BCR	-	IWW[2]	IWW[1]	IWW[0]	IWRWD[2]	IWRWD[1]	IWRWD[0]	IWRWS[2]
		IWRWS[1]	IWRWS[0]	IWRRD[2]	IWRRD[1]	IWRRD[0]	IWRRS[2]	IWRRS[1]	IWRRS[0]
		-	TYPE[2]	TYPE[1]	TYPE[0]	-	BSZ[1]	BSZ[0]	-
		-	-	-	-	-	-	-	-
	CS1BCR	-	IWW[2]	IWW[1]	IWW[0]	IWRWD[2]	IWRWD[1]	IWRWD[0]	IWRWS[2]
		IWRWS[1]	IWRWS[0]	IWRRD[2]	IWRRD[1]	IWRRD[0]	IWRRS[2]	IWRRS[1]	IWRRS[0]
		-	TYPE[2]	TYPE[1]	TYPE[0]	-	BSZ[1]	BSZ[0]	-
		-	-	-	-	-	-	-	-
	CS2BCR	-	IWW[2]	IWW[1]	IWW[0]	IWRWD[2]	IWRWD[1]	IWRWD[0]	IWRWS[2]
		IWRWS[1]	IWRWS[0]	IWRRD[2]	IWRRD[1]	IWRRD[0]	IWRRS[2]	IWRRS[1]	IWRRS[0]
		-	TYPE[2]	TYPE[1]	TYPE[0]	-	BSZ[1]	BSZ[0]	-
		-	-	-	-	-	-	-	-

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Bus state controller	CS3BCR	-	IWW[2]	IWW[1]	IWW[0]	IWRWD[2]	IWRWD[1]	IWRWD[0]	IWRWS[2]
		IWRWS[1]	IWRWS[0]	IWRRD[2]	IWRRD[1]	IWRRD[0]	IWRRS[2]	IWRRS[1]	IWRRS[0]
		-	TYPE[2]	TYPE[1]	TYPE[0]	-	BSZ[1]	BSZ[0]	-
		-	-	-	-	-	-	-	-
	CS4BCR	-	IWW[2]	IWW[1]	IWW[0]	IWRWD[2]	IWRWD[1]	IWRWD[0]	IWRWS[2]
		IWRWS[1]	IWRWS[0]	IWRRD[2]	IWRRD[1]	IWRRD[0]	IWRRS[2]	IWRRS[1]	IWRRS[0]
		-	TYPE[2]	TYPE[1]	TYPE[0]	-	BSZ[1]	BSZ[0]	-
		-	-	-	-	-	-	-	-
	CS5BCR	-	IWW[2]	IWW[1]	IWW[0]	IWRWD[2]	IWRWD[1]	IWRWD[0]	IWRWS[2]
		IWRWS[1]	IWRWS[0]	IWRRD[2]	IWRRD[1]	IWRRD[0]	IWRRS[2]	IWRRS[1]	IWRRS[0]
		-	TYPE[2]	TYPE[1]	TYPE[0]	-	BSZ[1]	BSZ[0]	-
		-	-	-	-	-	-	-	-
	CS0WCR Normal space, SRAM with byte selection, MPX-I/O	-	-	-	-	-	-	-	-
		-	-	-	BAS	-	-	-	-
		-	-	-	SW[1]	SW[0]	WR[3]	WR[2]	WR[1]
		WR[0]	WM	-	-	-	-	HW[1]	HW[0]
	CS0WCR Burst ROM (clock asynchronous)	-	-	-	-	-	-	-	-
		-	-	BST[1]	BST[0]	-	-	BW[1]	BW[0]
		-	-	-	-	-	W[3]	W[2]	W[1]
		W[0]	WM	-	-	-	-	-	-
	CS0WCR Burst ROM (clock synchronous)	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	BW[1]	BW[0]
		-	-	-	-	-	W[3]	W[2]	W[1]
		W[0]	WM	-	-	-	-	-	-
	CS1WCR	-	-	-	-	-	-	-	-
		-	-	-	BAS	-	WW[2]	WW[1]	WW[0]
		-	-	-	SW[1]	SW[0]	WR[3]	WR[2]	WR[1]
		WR[0]	WM	-	-	-	-	HW[1]	HW[0]
	CS2WCR Normal space, SRAM with byte selection, MPX-I/O	-	-	-	-	-	-	-	-
		-	-	-	BAS	-	-	-	-
		-	-	-	-	-	WR[3]	WR[2]	WR[1]
		WR[0]	WM	-	-	-	-	-	-
	CS2WCR Burst ROM (clock synchronous)	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	A2CL[1]
		A2CL[0]	-	-	-	-	-	-	-
	CS3WCR Normal space, SRAM with byte selection, MPX-I/O	-	-	-	-	-	-	-	-
		-	-	-	BAS	-	-	-	-
		-	-	-	-	-	WR[3]	WR[2]	WR[1]
		WR[0]	WM	-	-	-	-	-	-
	CS3WCR Burst ROM (clock synchronous)	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	WTRP[1]	WTRP[0]	-	WTRCD[1]	WTRCD[0]	-	A3CL[1]
		A3CL[0]	-	-	TRWL[1]	TRWL[0]	-	WTRC[1]	WTRC[0]
	CS4WCR Normal space, SRAM with byte selection, MPX-I/O	-	-	-	-	-	-	-	-
		-	-	-	BAS	-	WW[2]	WW[1]	WW[0]
		-	-	-	SW[1]	SW[0]	WR[3]	WR[2]	WR[1]
		WR[0]	WM	-	-	-	-	HW[1]	HW[0]
	CS4WCR Burst ROM (clock synchronous)	-	-	-	-	-	-	-	-
		-	-	BST[1]	BST[0]	-	-	BW[1]	BW[0]
		-	-	-	SW[1]	SW[0]	W[3]	W[2]	W[1]
		W[0]	WM	-	-	-	-	HW[1]	HW[0]
	CS5WCR Normal space, SRAM with byte selection, MPX-I/O	-	-	-	-	-	-	-	-
		-	-	SZSEL	MPXW/BAS	-	WW[2]	WW[1]	WW[0]
		-	-	-	SW[1]	SW[0]	WR[3]	WR[2]	WR[1]
		WR[0]	WM	-	-	-	-	HW[1]	HW[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Bus state controller	SDCR	-	-	-	-	-	-	-	-
		-	-	-	A2ROW[1]	A2ROW[0]	-	A2COL[1]	A2COL[0]
		-	-	DEEP	-	RFSH	RMODE	PDOWN	BACTV
		-	-	-	A3ROW[1]	A3ROW[0]	-	A3COL[1]	A3COL[0]
	RTCSR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		CMF	CMIE	CKS[2]	CKS[1]	CKS[0]	RRC[2]	RRC[1]	RRC[0]
	RTCNT	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	RTCOR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	TOSCOR0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	TOSCOR1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	TOSCOR2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	TOSCOR3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	TOSCOR4	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	TOSCOR5	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	TOSTR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	CS5TOSTF	CS4TOSTF	CS3TOSTF	CS2TOSTF	CS1TOSTF	CS0TOSTF
	TOENR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	CS5TOEN	CS4TOEN	CS3TOEN	CS2TOEN	CS1TOEN	CS0TOEN
Direct memory access controller	N0SA_0	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N0DA_0	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N0TB_0	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Direct memory access controller	N1SA_0	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N1DA_0	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N1TB_0	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	CRSA_0	CRSA[31]	CRSA[30]	CRSA[29]	CRSA[28]	CRSA[27]	CRSA[26]	CRSA[25]	CRSA[24]
		CRSA[23]	CRSA[22]	CRSA[21]	CRSA[20]	CRSA[19]	CRSA[18]	CRSA[17]	CRSA[16]
		CRSA[15]	CRSA[14]	CRSA[13]	CRSA[12]	CRSA[11]	CRSA[10]	CRSA[9]	CRSA[8]
		CRSA[7]	CRSA[6]	CRSA[5]	CRSA[4]	CRSA[3]	CRSA[2]	CRSA[1]	CRSA[0]
	CRDA_0	CRDA[31]	CRDA[30]	CRDA[29]	CRDA[28]	CRDA[27]	CRDA[26]	CRDA[25]	CRDA[24]
		CRDA[23]	CRDA[22]	CRDA[21]	CRDA[20]	CRDA[19]	CRDA[18]	CRDA[17]	CRDA[16]
		CRDA[15]	CRDA[14]	CRDA[13]	CRDA[12]	CRDA[11]	CRDA[10]	CRDA[9]	CRDA[8]
		CRDA[7]	CRDA[6]	CRDA[5]	CRDA[4]	CRDA[3]	CRDA[2]	CRDA[1]	CRDA[0]
	CRTB_0	CRTB[31]	CRTB[30]	CRTB[29]	CRTB[28]	CRTB[27]	CRTB[26]	CRTB[25]	CRTB[24]
		CRTB[23]	CRTB[22]	CRTB[21]	CRTB[20]	CRTB[19]	CRTB[18]	CRTB[17]	CRTB[16]
		CRTB[15]	CRTB[14]	CRTB[13]	CRTB[12]	CRTB[11]	CRTB[10]	CRTB[9]	CRTB[8]
		CRTB[7]	CRTB[6]	CRTB[5]	CRTB[4]	CRTB[3]	CRTB[2]	CRTB[1]	CRTB[0]
	CHSTAT_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INTMSK
		-	-	-	-	MODE	DER	DW	DL
		SR	TC	END	ER	SUS	TACT	RQST	EN
	CHCTRL_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	CLRINTMSK	SETINTMSK
		-	-	-	-	-	-	CLRSUS	SETSUS
		-	CLRTC	CLREND	CLRRQ	SWRST	STG	CLREN	SETEN
	CHCFG_0	DMS	REN	RSW	RSEL	SBE	-	-	DEM
		-	TM	DAD	SAD	DDS[3]	DDS[2]	DDS[1]	DDS[0]
		SDS[3]	SDS[2]	SDS[1]	SDS[0]	-	AM[2]	AM[1]	AM[0]
		-	LVL	HIEN	LOEN	REQD	SEL[2]	SEL[1]	SEL[0]
	CHITVL_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ITVL[15]	ITVL[14]	ITVL[13]	ITVL[12]	ITVL[11]	ITVL[10]	ITVL[9]	ITVL[8]
		ITVL[7]	ITVL[6]	ITVL[5]	ITVL[4]	ITVL[3]	ITVL[2]	ITVL[1]	ITVL[0]
	CHEXT_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DCA[3]	DCA[2]	DCA[1]	DCA[0]	-	-	-	-
		SCA[3]	SCA[2]	SCA[1]	SCA[0]	-	-	-	-
	NXLA_0	NXLA[31]	NXLA[30]	NXLA[29]	NXLA[28]	NXLA[27]	NXLA[26]	NXLA[25]	NXLA[24]
		NXLA[23]	NXLA[22]	NXLA[21]	NXLA[20]	NXLA[19]	NXLA[18]	NXLA[17]	NXLA[16]
		NXLA[15]	NXLA[14]	NXLA[13]	NXLA[12]	NXLA[11]	NXLA[10]	NXLA[9]	NXLA[8]
		NXLA[7]	NXLA[6]	NXLA[5]	NXLA[4]	NXLA[3]	NXLA[2]	NXLA[1]	NXLA[0]
	CRLA_0	CRLA[31]	CRLA[30]	CRLA[29]	CRLA[28]	CRLA[27]	CRLA[26]	CRLA[25]	CRLA[24]
		CRLA[23]	CRLA[22]	CRLA[21]	CRLA[20]	CRLA[19]	CRLA[18]	CRLA[17]	CRLA[16]
		CRLA[15]	CRLA[14]	CRLA[13]	CRLA[12]	CRLA[11]	CRLA[10]	CRLA[9]	CRLA[8]
		CRLA[7]	CRLA[6]	CRLA[5]	CRLA[4]	CRLA[3]	CRLA[2]	CRLA[1]	CRLA[0]
	N0SA_1	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N0DA_1	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Direct memory access controller	N0TB_1	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	N1SA_1	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N1DA_1	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N1TB_1	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	CRSA_1	CRSA[31]	CRSA[30]	CRSA[29]	CRSA[28]	CRSA[27]	CRSA[26]	CRSA[25]	CRSA[24]
		CRSA[23]	CRSA[22]	CRSA[21]	CRSA[20]	CRSA[19]	CRSA[18]	CRSA[17]	CRSA[16]
		CRSA[15]	CRSA[14]	CRSA[13]	CRSA[12]	CRSA[11]	CRSA[10]	CRSA[9]	CRSA[8]
		CRSA[7]	CRSA[6]	CRSA[5]	CRSA[4]	CRSA[3]	CRSA[2]	CRSA[1]	CRSA[0]
	CRDA_1	CRDA[31]	CRDA[30]	CRDA[29]	CRDA[28]	CRDA[27]	CRDA[26]	CRDA[25]	CRDA[24]
		CRDA[23]	CRDA[22]	CRDA[21]	CRDA[20]	CRDA[19]	CRDA[18]	CRDA[17]	CRDA[16]
		CRDA[15]	CRDA[14]	CRDA[13]	CRDA[12]	CRDA[11]	CRDA[10]	CRDA[9]	CRDA[8]
		CRDA[7]	CRDA[6]	CRDA[5]	CRDA[4]	CRDA[3]	CRDA[2]	CRDA[1]	CRDA[0]
	CRTB_1	CRTB[31]	CRTB[30]	CRTB[29]	CRTB[28]	CRTB[27]	CRTB[26]	CRTB[25]	CRTB[24]
		CRTB[23]	CRTB[22]	CRTB[21]	CRTB[20]	CRTB[19]	CRTB[18]	CRTB[17]	CRTB[16]
		CRTB[15]	CRTB[14]	CRTB[13]	CRTB[12]	CRTB[11]	CRTB[10]	CRTB[9]	CRTB[8]
		CRTB[7]	CRTB[6]	CRTB[5]	CRTB[4]	CRTB[3]	CRTB[2]	CRTB[1]	CRTB[0]
	CHSTAT_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INTMSK
		-	-	-	-	MODE	DER	DW	DL
		SR	TC	END	ER	SUS	TACT	RQST	EN
	CHCTRL_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	CLRINTMSK	SETINTMSK
		-	-	-	-	-	-	CLRSUS	SETSUS
		-	CLRTC	CLREND	CLRRQ	SWRST	STG	CLREN	SETEN
	CHCFG_1	DMS	REN	RSW	RSEL	SBE	-	-	DEM
		-	TM	DAD	SAD	DDS[3]	DDS[2]	DDS[1]	DDS[0]
		SDS[3]	SDS[2]	SDS[1]	SDS[0]	-	AM[2]	AM[1]	AM[0]
		-	LVL	HIEN	LOEN	REQD	SEL[2]	SEL[1]	SEL[0]
	CHITVL_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ITVL[15]	ITVL[14]	ITVL[13]	ITVL[12]	ITVL[11]	ITVL[10]	ITVL[9]	ITVL[8]
		ITVL[7]	ITVL[6]	ITVL[5]	ITVL[4]	ITVL[3]	ITVL[2]	ITVL[1]	ITVL[0]
	CHEXT_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DCA[3]	DCA[2]	DCA[1]	DCA[0]	-	-	-	-
		SCA[3]	SCA[2]	SCA[1]	SCA[0]	-	-	-	-
	NXLA_1	NXLA[31]	NXLA[30]	NXLA[29]	NXLA[28]	NXLA[27]	NXLA[26]	NXLA[25]	NXLA[24]
		NXLA[23]	NXLA[22]	NXLA[21]	NXLA[20]	NXLA[19]	NXLA[18]	NXLA[17]	NXLA[16]
		NXLA[15]	NXLA[14]	NXLA[13]	NXLA[12]	NXLA[11]	NXLA[10]	NXLA[9]	NXLA[8]
		NXLA[7]	NXLA[6]	NXLA[5]	NXLA[4]	NXLA[3]	NXLA[2]	NXLA[1]	NXLA[0]
	CRLA_1	CRLA[31]	CRLA[30]	CRLA[29]	CRLA[28]	CRLA[27]	CRLA[26]	CRLA[25]	CRLA[24]
		CRLA[23]	CRLA[22]	CRLA[21]	CRLA[20]	CRLA[19]	CRLA[18]	CRLA[17]	CRLA[16]
		CRLA[15]	CRLA[14]	CRLA[13]	CRLA[12]	CRLA[11]	CRLA[10]	CRLA[9]	CRLA[8]
		CRLA[7]	CRLA[6]	CRLA[5]	CRLA[4]	CRLA[3]	CRLA[2]	CRLA[1]	CRLA[0]
	N0SA_2	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Direct memory access controller	N0DA_2	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N0TB_2	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	N1SA_2	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N1DA_2	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N1TB_2	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	CRSA_2	CRSA[31]	CRSA[30]	CRSA[29]	CRSA[28]	CRSA[27]	CRSA[26]	CRSA[25]	CRSA[24]
		CRSA[23]	CRSA[22]	CRSA[21]	CRSA[20]	CRSA[19]	CRSA[18]	CRSA[17]	CRSA[16]
		CRSA[15]	CRSA[14]	CRSA[13]	CRSA[12]	CRSA[11]	CRSA[10]	CRSA[9]	CRSA[8]
		CRSA[7]	CRSA[6]	CRSA[5]	CRSA[4]	CRSA[3]	CRSA[2]	CRSA[1]	CRSA[0]
	CRDA_2	CRDA[31]	CRDA[30]	CRDA[29]	CRDA[28]	CRDA[27]	CRDA[26]	CRDA[25]	CRDA[24]
		CRDA[23]	CRDA[22]	CRDA[21]	CRDA[20]	CRDA[19]	CRDA[18]	CRDA[17]	CRDA[16]
		CRDA[15]	CRDA[14]	CRDA[13]	CRDA[12]	CRDA[11]	CRDA[10]	CRDA[9]	CRDA[8]
		CRDA[7]	CRDA[6]	CRDA[5]	CRDA[4]	CRDA[3]	CRDA[2]	CRDA[1]	CRDA[0]
	CRTB_2	CRTB[31]	CRTB[30]	CRTB[29]	CRTB[28]	CRTB[27]	CRTB[26]	CRTB[25]	CRTB[24]
		CRTB[23]	CRTB[22]	CRTB[21]	CRTB[20]	CRTB[19]	CRTB[18]	CRTB[17]	CRTB[16]
		CRTB[15]	CRTB[14]	CRTB[13]	CRTB[12]	CRTB[11]	CRTB[10]	CRTB[9]	CRTB[8]
		CRTB[7]	CRTB[6]	CRTB[5]	CRTB[4]	CRTB[3]	CRTB[2]	CRTB[1]	CRTB[0]
	CHSTAT_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INTMSK
		-	-	-	-	MODE	DER	DW	DL
		SR	TC	END	ER	SUS	TACT	RQST	EN
	CHCTRL_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	CLRINTMSK	SETINTMSK
		-	-	-	-	-	-	CLRSUS	SETSUS
		-	CLRTC	CLREND	CLRRQ	SWRST	STG	CLREN	SETEN
	CHCFG_2	DMS	REN	RSW	RSEL	SBE	-	-	DEM
		-	TM	DAD	SAD	DDS[3]	DDS[2]	DDS[1]	DDS[0]
		SDS[3]	SDS[2]	SDS[1]	SDS[0]	-	AM[2]	AM[1]	AM[0]
		-	LVL	HIEN	LOEN	REQD	SEL[2]	SEL[1]	SEL[0]
	CHITVL_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ITVL[15]	ITVL[14]	ITVL[13]	ITVL[12]	ITVL[11]	ITVL[10]	ITVL[9]	ITVL[8]
		ITVL[7]	ITVL[6]	ITVL[5]	ITVL[4]	ITVL[3]	ITVL[2]	ITVL[1]	ITVL[0]
	CHEXT_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DCA[3]	DCA[2]	DCA[1]	DCA[0]	-	-	-	-
		SCA[3]	SCA[2]	SCA[1]	SCA[0]	-	-	-	-
	NXLA_2	NXLA[31]	NXLA[30]	NXLA[29]	NXLA[28]	NXLA[27]	NXLA[26]	NXLA[25]	NXLA[24]
		NXLA[23]	NXLA[22]	NXLA[21]	NXLA[20]	NXLA[19]	NXLA[18]	NXLA[17]	NXLA[16]
		NXLA[15]	NXLA[14]	NXLA[13]	NXLA[12]	NXLA[11]	NXLA[10]	NXLA[9]	NXLA[8]
		NXLA[7]	NXLA[6]	NXLA[5]	NXLA[4]	NXLA[3]	NXLA[2]	NXLA[1]	NXLA[0]
	CRLA_2	CRLA[31]	CRLA[30]	CRLA[29]	CRLA[28]	CRLA[27]	CRLA[26]	CRLA[25]	CRLA[24]
		CRLA[23]	CRLA[22]	CRLA[21]	CRLA[20]	CRLA[19]	CRLA[18]	CRLA[17]	CRLA[16]
		CRLA[15]	CRLA[14]	CRLA[13]	CRLA[12]	CRLA[11]	CRLA[10]	CRLA[9]	CRLA[8]
		CRLA[7]	CRLA[6]	CRLA[5]	CRLA[4]	CRLA[3]	CRLA[2]	CRLA[1]	CRLA[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Direct memory access controller	N0SA_3	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N0DA_3	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N0TB_3	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	N1SA_3	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N1DA_3	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N1TB_3	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	CRSA_3	CRSA[31]	CRSA[30]	CRSA[29]	CRSA[28]	CRSA[27]	CRSA[26]	CRSA[25]	CRSA[24]
		CRSA[23]	CRSA[22]	CRSA[21]	CRSA[20]	CRSA[19]	CRSA[18]	CRSA[17]	CRSA[16]
		CRSA[15]	CRSA[14]	CRSA[13]	CRSA[12]	CRSA[11]	CRSA[10]	CRSA[9]	CRSA[8]
		CRSA[7]	CRSA[6]	CRSA[5]	CRSA[4]	CRSA[3]	CRSA[2]	CRSA[1]	CRSA[0]
	CRDA_3	CRDA[31]	CRDA[30]	CRDA[29]	CRDA[28]	CRDA[27]	CRDA[26]	CRDA[25]	CRDA[24]
		CRDA[23]	CRDA[22]	CRDA[21]	CRDA[20]	CRDA[19]	CRDA[18]	CRDA[17]	CRDA[16]
		CRDA[15]	CRDA[14]	CRDA[13]	CRDA[12]	CRDA[11]	CRDA[10]	CRDA[9]	CRDA[8]
		CRDA[7]	CRDA[6]	CRDA[5]	CRDA[4]	CRDA[3]	CRDA[2]	CRDA[1]	CRDA[0]
	CRTB_3	CRTB[31]	CRTB[30]	CRTB[29]	CRTB[28]	CRTB[27]	CRTB[26]	CRTB[25]	CRTB[24]
		CRTB[23]	CRTB[22]	CRTB[21]	CRTB[20]	CRTB[19]	CRTB[18]	CRTB[17]	CRTB[16]
		CRTB[15]	CRTB[14]	CRTB[13]	CRTB[12]	CRTB[11]	CRTB[10]	CRTB[9]	CRTB[8]
		CRTB[7]	CRTB[6]	CRTB[5]	CRTB[4]	CRTB[3]	CRTB[2]	CRTB[1]	CRTB[0]
	CHSTAT_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INTMSK
		-	-	-	-	MODE	DER	DW	DL
		SR	TC	END	ER	SUS	TACT	RQST	EN
	CHCTRL_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	CLRINTMSK	SETINTMSK
		-	-	-	-	-	-	CLRSUS	SETSUS
		-	CLRTC	CLREND	CLRRQ	SWRST	STG	CLREN	SETEN
	CHCFG_3	DMS	REN	RSW	RSEL	SBE	-	-	DEM
		-	TM	DAD	SAD	DDS[3]	DDS[2]	DDS[1]	DDS[0]
		SDS[3]	SDS[2]	SDS[1]	SDS[0]	-	AM[2]	AM[1]	AM[0]
		-	LVL	HIEN	LOEN	REQD	SEL[2]	SEL[1]	SEL[0]
	CHITVL_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ITVL[15]	ITVL[14]	ITVL[13]	ITVL[12]	ITVL[11]	ITVL[10]	ITVL[9]	ITVL[8]
		ITVL[7]	ITVL[6]	ITVL[5]	ITVL[4]	ITVL[3]	ITVL[2]	ITVL[1]	ITVL[0]
	CHEXT_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DCA[3]	DCA[2]	DCA[1]	DCA[0]	-	-	-	-
		SCA[3]	SCA[2]	SCA[1]	SCA[0]	-	-	-	-
	NXLA_3	NXLA[31]	NXLA[30]	NXLA[29]	NXLA[28]	NXLA[27]	NXLA[26]	NXLA[25]	NXLA[24]
		NXLA[23]	NXLA[22]	NXLA[21]	NXLA[20]	NXLA[19]	NXLA[18]	NXLA[17]	NXLA[16]
		NXLA[15]	NXLA[14]	NXLA[13]	NXLA[12]	NXLA[11]	NXLA[10]	NXLA[9]	NXLA[8]
		NXLA[7]	NXLA[6]	NXLA[5]	NXLA[4]	NXLA[3]	NXLA[2]	NXLA[1]	NXLA[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Direct memory access controller	CRLA_3	CRLA[31]	CRLA[30]	CRLA[29]	CRLA[28]	CRLA[27]	CRLA[26]	CRLA[25]	CRLA[24]
		CRLA[23]	CRLA[22]	CRLA[21]	CRLA[20]	CRLA[19]	CRLA[18]	CRLA[17]	CRLA[16]
		CRLA[15]	CRLA[14]	CRLA[13]	CRLA[12]	CRLA[11]	CRLA[10]	CRLA[9]	CRLA[8]
		CRLA[7]	CRLA[6]	CRLA[5]	CRLA[4]	CRLA[3]	CRLA[2]	CRLA[1]	CRLA[0]
	N0SA_4	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N0DA_4	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N0TB_4	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	N1SA_4	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N1DA_4	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N1TB_4	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	CRSA_4	CRSA[31]	CRSA[30]	CRSA[29]	CRSA[28]	CRSA[27]	CRSA[26]	CRSA[25]	CRSA[24]
		CRSA[23]	CRSA[22]	CRSA[21]	CRSA[20]	CRSA[19]	CRSA[18]	CRSA[17]	CRSA[16]
		CRSA[15]	CRSA[14]	CRSA[13]	CRSA[12]	CRSA[11]	CRSA[10]	CRSA[9]	CRSA[8]
		CRSA[7]	CRSA[6]	CRSA[5]	CRSA[4]	CRSA[3]	CRSA[2]	CRSA[1]	CRSA[0]
	CRDA_4	CRDA[31]	CRDA[30]	CRDA[29]	CRDA[28]	CRDA[27]	CRDA[26]	CRDA[25]	CRDA[24]
		CRDA[23]	CRDA[22]	CRDA[21]	CRDA[20]	CRDA[19]	CRDA[18]	CRDA[17]	CRDA[16]
		CRDA[15]	CRDA[14]	CRDA[13]	CRDA[12]	CRDA[11]	CRDA[10]	CRDA[9]	CRDA[8]
		CRDA[7]	CRDA[6]	CRDA[5]	CRDA[4]	CRDA[3]	CRDA[2]	CRDA[1]	CRDA[0]
	CRTB_4	CRTB[31]	CRTB[30]	CRTB[29]	CRTB[28]	CRTB[27]	CRTB[26]	CRTB[25]	CRTB[24]
		CRTB[23]	CRTB[22]	CRTB[21]	CRTB[20]	CRTB[19]	CRTB[18]	CRTB[17]	CRTB[16]
		CRTB[15]	CRTB[14]	CRTB[13]	CRTB[12]	CRTB[11]	CRTB[10]	CRTB[9]	CRTB[8]
		CRTB[7]	CRTB[6]	CRTB[5]	CRTB[4]	CRTB[3]	CRTB[2]	CRTB[1]	CRTB[0]
	CHSTAT_4	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INTMSK
		-	-	-	-	MODE	DER	DW	DL
		SR	TC	END	ER	SUS	TACT	RQST	EN
	CHCTRL_4	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	CLRINTMSK	SETINTMSK
		-	-	-	-	-	-	CLRSUS	SETSUS
		-	CLRTC	CLREND	CLRRQ	SWRST	STG	CLREN	SETEN
	CHCFG_4	DMS	REN	RSW	RSEL	SBE	-	-	DEM
		-	TM	DAD	SAD	DDS[3]	DDS[2]	DDS[1]	DDS[0]
		SDS[3]	SDS[2]	SDS[1]	SDS[0]	-	AM[2]	AM[1]	AM[0]
		-	LVL	HIEN	LOEN	REQD	SEL[2]	SEL[1]	SEL[0]
	CHITVL_4	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ITVL[15]	ITVL[14]	ITVL[13]	ITVL[12]	ITVL[11]	ITVL[10]	ITVL[9]	ITVL[8]
		ITVL[7]	ITVL[6]	ITVL[5]	ITVL[4]	ITVL[3]	ITVL[2]	ITVL[1]	ITVL[0]
	CHEXT_4	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DCA[3]	DCA[2]	DCA[1]	DCA[0]	-	-	-	-
		SCA[3]	SCA[2]	SCA[1]	SCA[0]	-	-	-	-

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Direct memory access controller	NXLA_4	NXLA[31]	NXLA[30]	NXLA[29]	NXLA[28]	NXLA[27]	NXLA[26]	NXLA[25]	NXLA[24]
		NXLA[23]	NXLA[22]	NXLA[21]	NXLA[20]	NXLA[19]	NXLA[18]	NXLA[17]	NXLA[16]
		NXLA[15]	NXLA[14]	NXLA[13]	NXLA[12]	NXLA[11]	NXLA[10]	NXLA[9]	NXLA[8]
		NXLA[7]	NXLA[6]	NXLA[5]	NXLA[4]	NXLA[3]	NXLA[2]	NXLA[1]	NXLA[0]
	CRLA_4	CRLA[31]	CRLA[30]	CRLA[29]	CRLA[28]	CRLA[27]	CRLA[26]	CRLA[25]	CRLA[24]
		CRLA[23]	CRLA[22]	CRLA[21]	CRLA[20]	CRLA[19]	CRLA[18]	CRLA[17]	CRLA[16]
		CRLA[15]	CRLA[14]	CRLA[13]	CRLA[12]	CRLA[11]	CRLA[10]	CRLA[9]	CRLA[8]
		CRLA[7]	CRLA[6]	CRLA[5]	CRLA[4]	CRLA[3]	CRLA[2]	CRLA[1]	CRLA[0]
	N0SA_5	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N0DA_5	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N0TB_5	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	N1SA_5	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N1DA_5	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N1TB_5	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	CRSA_5	CRSA[31]	CRSA[30]	CRSA[29]	CRSA[28]	CRSA[27]	CRSA[26]	CRSA[25]	CRSA[24]
		CRSA[23]	CRSA[22]	CRSA[21]	CRSA[20]	CRSA[19]	CRSA[18]	CRSA[17]	CRSA[16]
		CRSA[15]	CRSA[14]	CRSA[13]	CRSA[12]	CRSA[11]	CRSA[10]	CRSA[9]	CRSA[8]
		CRSA[7]	CRSA[6]	CRSA[5]	CRSA[4]	CRSA[3]	CRSA[2]	CRSA[1]	CRSA[0]
	CRDA_5	CRDA[31]	CRDA[30]	CRDA[29]	CRDA[28]	CRDA[27]	CRDA[26]	CRDA[25]	CRDA[24]
		CRDA[23]	CRDA[22]	CRDA[21]	CRDA[20]	CRDA[19]	CRDA[18]	CRDA[17]	CRDA[16]
		CRDA[15]	CRDA[14]	CRDA[13]	CRDA[12]	CRDA[11]	CRDA[10]	CRDA[9]	CRDA[8]
		CRDA[7]	CRDA[6]	CRDA[5]	CRDA[4]	CRDA[3]	CRDA[2]	CRDA[1]	CRDA[0]
	CRTB_5	CRTB[31]	CRTB[30]	CRTB[29]	CRTB[28]	CRTB[27]	CRTB[26]	CRTB[25]	CRTB[24]
		CRTB[23]	CRTB[22]	CRTB[21]	CRTB[20]	CRTB[19]	CRTB[18]	CRTB[17]	CRTB[16]
		CRTB[15]	CRTB[14]	CRTB[13]	CRTB[12]	CRTB[11]	CRTB[10]	CRTB[9]	CRTB[8]
		CRTB[7]	CRTB[6]	CRTB[5]	CRTB[4]	CRTB[3]	CRTB[2]	CRTB[1]	CRTB[0]
	CHSTAT_5	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INTMSK
		-	-	-	-	MODE	DER	DW	DL
		SR	TC	END	ER	SUS	TACT	RQST	EN
	CHCTRL_5	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	CLRINTMSK	SETINTMSK
		-	-	-	-	-	-	CLRSUS	SETSUS
		-	CLRTC	CLREND	CLRRQ	SWRST	STG	CLREN	SETEN
	CHCFG_5	DMS	REN	RSW	RSEL	SBE	-	-	DEM
		-	TM	DAD	SAD	DDS[3]	DDS[2]	DDS[1]	DDS[0]
		SDS[3]	SDS[2]	SDS[1]	SDS[0]	-	AM[2]	AM[1]	AM[0]
		-	LVL	HIEN	LOEN	REQD	SEL[2]	SEL[1]	SEL[0]
	CHITVL_5	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ITVL[15]	ITVL[14]	ITVL[13]	ITVL[12]	ITVL[11]	ITVL[10]	ITVL[9]	ITVL[8]
		ITVL[7]	ITVL[6]	ITVL[5]	ITVL[4]	ITVL[3]	ITVL[2]	ITVL[1]	ITVL[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Direct memory access controller	CHEXT_5	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DCA[3]	DCA[2]	DCA[1]	DCA[0]	-	-	-	-
		SCA[3]	SCA[2]	SCA[1]	SCA[0]	-	-	-	-
	NXLA_5	NXLA[31]	NXLA[30]	NXLA[29]	NXLA[28]	NXLA[27]	NXLA[26]	NXLA[25]	NXLA[24]
		NXLA[23]	NXLA[22]	NXLA[21]	NXLA[20]	NXLA[19]	NXLA[18]	NXLA[17]	NXLA[16]
		NXLA[15]	NXLA[14]	NXLA[13]	NXLA[12]	NXLA[11]	NXLA[10]	NXLA[9]	NXLA[8]
		NXLA[7]	NXLA[6]	NXLA[5]	NXLA[4]	NXLA[3]	NXLA[2]	NXLA[1]	NXLA[0]
	CRLA_5	CRLA[31]	CRLA[30]	CRLA[29]	CRLA[28]	CRLA[27]	CRLA[26]	CRLA[25]	CRLA[24]
		CRLA[23]	CRLA[22]	CRLA[21]	CRLA[20]	CRLA[19]	CRLA[18]	CRLA[17]	CRLA[16]
		CRLA[15]	CRLA[14]	CRLA[13]	CRLA[12]	CRLA[11]	CRLA[10]	CRLA[9]	CRLA[8]
		CRLA[7]	CRLA[6]	CRLA[5]	CRLA[4]	CRLA[3]	CRLA[2]	CRLA[1]	CRLA[0]
	N0SA_6	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N0DA_6	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N0TB_6	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	N1SA_6	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N1DA_6	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N1TB_6	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	CRSA_6	CRSA[31]	CRSA[30]	CRSA[29]	CRSA[28]	CRSA[27]	CRSA[26]	CRSA[25]	CRSA[24]
		CRSA[23]	CRSA[22]	CRSA[21]	CRSA[20]	CRSA[19]	CRSA[18]	CRSA[17]	CRSA[16]
		CRSA[15]	CRSA[14]	CRSA[13]	CRSA[12]	CRSA[11]	CRSA[10]	CRSA[9]	CRSA[8]
		CRSA[7]	CRSA[6]	CRSA[5]	CRSA[4]	CRSA[3]	CRSA[2]	CRSA[1]	CRSA[0]
	CRDA_6	CRDA[31]	CRDA[30]	CRDA[29]	CRDA[28]	CRDA[27]	CRDA[26]	CRDA[25]	CRDA[24]
		CRDA[23]	CRDA[22]	CRDA[21]	CRDA[20]	CRDA[19]	CRDA[18]	CRDA[17]	CRDA[16]
		CRDA[15]	CRDA[14]	CRDA[13]	CRDA[12]	CRDA[11]	CRDA[10]	CRDA[9]	CRDA[8]
		CRDA[7]	CRDA[6]	CRDA[5]	CRDA[4]	CRDA[3]	CRDA[2]	CRDA[1]	CRDA[0]
	CRTB_6	CRTB[31]	CRTB[30]	CRTB[29]	CRTB[28]	CRTB[27]	CRTB[26]	CRTB[25]	CRTB[24]
		CRTB[23]	CRTB[22]	CRTB[21]	CRTB[20]	CRTB[19]	CRTB[18]	CRTB[17]	CRTB[16]
		CRTB[15]	CRTB[14]	CRTB[13]	CRTB[12]	CRTB[11]	CRTB[10]	CRTB[9]	CRTB[8]
		CRTB[7]	CRTB[6]	CRTB[5]	CRTB[4]	CRTB[3]	CRTB[2]	CRTB[1]	CRTB[0]
	CHSTAT_6	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INTMSK
		-	-	-	-	MODE	DER	DW	DL
		SR	TC	END	ER	SUS	TACT	RQST	EN
	CHCTRL_6	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	CLRINTMSK	SETINTMSK
		-	-	-	-	-	-	CLRSUS	SETSUS
		-	CLRTC	CLREND	CLRRQ	SWRST	STG	CLREN	SETEN
	CHCFG_6	DMS	REN	RSW	RSEL	SBE	-	-	DEM
		-	TM	DAD	SAD	DDS[3]	DDS[2]	DDS[1]	DDS[0]
		SDS[3]	SDS[2]	SDS[1]	SDS[0]	-	AM[2]	AM[1]	AM[0]
		-	LVL	HIEN	LOEN	REQD	SEL[2]	SEL[1]	SEL[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Direct memory access controller	CHITVL_6	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ITVL[15]	ITVL[14]	ITVL[13]	ITVL[12]	ITVL[11]	ITVL[10]	ITVL[9]	ITVL[8]
		ITVL[7]	ITVL[6]	ITVL[5]	ITVL[4]	ITVL[3]	ITVL[2]	ITVL[1]	ITVL[0]
	CHEXT_6	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DCA[3]	DCA[2]	DCA[1]	DCA[0]	-	-	-	-
		SCA[3]	SCA[2]	SCA[1]	SCA[0]	-	-	-	-
	NXLA_6	NXLA[31]	NXLA[30]	NXLA[29]	NXLA[28]	NXLA[27]	NXLA[26]	NXLA[25]	NXLA[24]
		NXLA[23]	NXLA[22]	NXLA[21]	NXLA[20]	NXLA[19]	NXLA[18]	NXLA[17]	NXLA[16]
		NXLA[15]	NXLA[14]	NXLA[13]	NXLA[12]	NXLA[11]	NXLA[10]	NXLA[9]	NXLA[8]
		NXLA[7]	NXLA[6]	NXLA[5]	NXLA[4]	NXLA[3]	NXLA[2]	NXLA[1]	NXLA[0]
	CRLA_6	CRLA[31]	CRLA[30]	CRLA[29]	CRLA[28]	CRLA[27]	CRLA[26]	CRLA[25]	CRLA[24]
		CRLA[23]	CRLA[22]	CRLA[21]	CRLA[20]	CRLA[19]	CRLA[18]	CRLA[17]	CRLA[16]
		CRLA[15]	CRLA[14]	CRLA[13]	CRLA[12]	CRLA[11]	CRLA[10]	CRLA[9]	CRLA[8]
		CRLA[7]	CRLA[6]	CRLA[5]	CRLA[4]	CRLA[3]	CRLA[2]	CRLA[1]	CRLA[0]
	N0SA_7	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N0DA_7	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N0TB_7	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	N1SA_7	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N1DA_7	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N1TB_7	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	CRSA_7	CRSA[31]	CRSA[30]	CRSA[29]	CRSA[28]	CRSA[27]	CRSA[26]	CRSA[25]	CRSA[24]
		CRSA[23]	CRSA[22]	CRSA[21]	CRSA[20]	CRSA[19]	CRSA[18]	CRSA[17]	CRSA[16]
		CRSA[15]	CRSA[14]	CRSA[13]	CRSA[12]	CRSA[11]	CRSA[10]	CRSA[9]	CRSA[8]
		CRSA[7]	CRSA[6]	CRSA[5]	CRSA[4]	CRSA[3]	CRSA[2]	CRSA[1]	CRSA[0]
	CRDA_7	CRDA[31]	CRDA[30]	CRDA[29]	CRDA[28]	CRDA[27]	CRDA[26]	CRDA[25]	CRDA[24]
		CRDA[23]	CRDA[22]	CRDA[21]	CRDA[20]	CRDA[19]	CRDA[18]	CRDA[17]	CRDA[16]
		CRDA[15]	CRDA[14]	CRDA[13]	CRDA[12]	CRDA[11]	CRDA[10]	CRDA[9]	CRDA[8]
		CRDA[7]	CRDA[6]	CRDA[5]	CRDA[4]	CRDA[3]	CRDA[2]	CRDA[1]	CRDA[0]
	CRTB_7	CRTB[31]	CRTB[30]	CRTB[29]	CRTB[28]	CRTB[27]	CRTB[26]	CRTB[25]	CRTB[24]
		CRTB[23]	CRTB[22]	CRTB[21]	CRTB[20]	CRTB[19]	CRTB[18]	CRTB[17]	CRTB[16]
		CRTB[15]	CRTB[14]	CRTB[13]	CRTB[12]	CRTB[11]	CRTB[10]	CRTB[9]	CRTB[8]
		CRTB[7]	CRTB[6]	CRTB[5]	CRTB[4]	CRTB[3]	CRTB[2]	CRTB[1]	CRTB[0]
	CHSTAT_7	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INTMSK
		-	-	-	-	MODE	DER	DW	DL
		SR	TC	END	ER	SUS	TACT	RQST	EN
	CHCTRL_7	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	CLRINTMSK	SETINTMSK
		-	-	-	-	-	-	CLRSUS	SETSUS
		-	CLRTC	CLREND	CLRRQ	SWRST	STG	CLREN	SETEN

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Direct memory access controller	CHCFG_7	DMS	REN	RSW	RSEL	SBE	-	-	DEM
		-	TM	DAD	SAD	DDS[3]	DDS[2]	DDS[1]	DDS[0]
		SDS[3]	SDS[2]	SDS[1]	SDS[0]	-	AM[2]	AM[1]	AM[0]
		-	LVL	HIEN	LOEN	REQD	SEL[2]	SEL[1]	SEL[0]
	CHITVL_7	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ITVL[15]	ITVL[14]	ITVL[13]	ITVL[12]	ITVL[11]	ITVL[10]	ITVL[9]	ITVL[8]
		ITVL[7]	ITVL[6]	ITVL[5]	ITVL[4]	ITVL[3]	ITVL[2]	ITVL[1]	ITVL[0]
	CHEXT_7	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DCA[3]	DCA[2]	DCA[1]	DCA[0]	-	-	-	-
		SCA[3]	SCA[2]	SCA[1]	SCA[0]	-	-	-	-
	NXLA_7	NXLA[31]	NXLA[30]	NXLA[29]	NXLA[28]	NXLA[27]	NXLA[26]	NXLA[25]	NXLA[24]
		NXLA[23]	NXLA[22]	NXLA[21]	NXLA[20]	NXLA[19]	NXLA[18]	NXLA[17]	NXLA[16]
		NXLA[15]	NXLA[14]	NXLA[13]	NXLA[12]	NXLA[11]	NXLA[10]	NXLA[9]	NXLA[8]
		NXLA[7]	NXLA[6]	NXLA[5]	NXLA[4]	NXLA[3]	NXLA[2]	NXLA[1]	NXLA[0]
	CRLA_7	CRLA[31]	CRLA[30]	CRLA[29]	CRLA[28]	CRLA[27]	CRLA[26]	CRLA[25]	CRLA[24]
		CRLA[23]	CRLA[22]	CRLA[21]	CRLA[20]	CRLA[19]	CRLA[18]	CRLA[17]	CRLA[16]
		CRLA[15]	CRLA[14]	CRLA[13]	CRLA[12]	CRLA[11]	CRLA[10]	CRLA[9]	CRLA[8]
		CRLA[7]	CRLA[6]	CRLA[5]	CRLA[4]	CRLA[3]	CRLA[2]	CRLA[1]	CRLA[0]
	DCTRL_0_7	LWCA[3]	LWCA[2]	LWCA[1]	LWCA[0]	-	LWPR[2]	LWPR[1]	LWPR[0]
		LDCA[3]	LDCA[2]	LDCA[1]	LDCA[0]	-	LDPR[2]	LDPR[1]	LDPR[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	LVINT	PR
	DSTAT_EN_0_7	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
	DSTAT_ER_0_7	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0
	DSTAT_END_0_7	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		END7	END6	END5	END4	END3	END2	END1	END0
	DSTAT_TC_0_7	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
	DSTAT_SUS_0_7	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		SUS7	SUS6	SUS5	SUS4	SUS3	SUS2	SUS1	SUS0
	N0SA_8	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N0DA_8	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N0TB_8	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	N1SA_8	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Direct memory access controller	N1DA_8	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N1TB_8	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	CRSA_8	CRSA[31]	CRSA[30]	CRSA[29]	CRSA[28]	CRSA[27]	CRSA[26]	CRSA[25]	CRSA[24]
		CRSA[23]	CRSA[22]	CRSA[21]	CRSA[20]	CRSA[19]	CRSA[18]	CRSA[17]	CRSA[16]
		CRSA[15]	CRSA[14]	CRSA[13]	CRSA[12]	CRSA[11]	CRSA[10]	CRSA[9]	CRSA[8]
		CRSA[7]	CRSA[6]	CRSA[5]	CRSA[4]	CRSA[3]	CRSA[2]	CRSA[1]	CRSA[0]
	CRDA_8	CRDA[31]	CRDA[30]	CRDA[29]	CRDA[28]	CRDA[27]	CRDA[26]	CRDA[25]	CRDA[24]
		CRDA[23]	CRDA[22]	CRDA[21]	CRDA[20]	CRDA[19]	CRDA[18]	CRDA[17]	CRDA[16]
		CRDA[15]	CRDA[14]	CRDA[13]	CRDA[12]	CRDA[11]	CRDA[10]	CRDA[9]	CRDA[8]
		CRDA[7]	CRDA[6]	CRDA[5]	CRDA[4]	CRDA[3]	CRDA[2]	CRDA[1]	CRDA[0]
	CRTB_8	CRTB[31]	CRTB[30]	CRTB[29]	CRTB[28]	CRTB[27]	CRTB[26]	CRTB[25]	CRTB[24]
		CRTB[23]	CRTB[22]	CRTB[21]	CRTB[20]	CRTB[19]	CRTB[18]	CRTB[17]	CRTB[16]
		CRTB[15]	CRTB[14]	CRTB[13]	CRTB[12]	CRTB[11]	CRTB[10]	CRTB[9]	CRTB[8]
		CRTB[7]	CRTB[6]	CRTB[5]	CRTB[4]	CRTB[3]	CRTB[2]	CRTB[1]	CRTB[0]
	CHSTAT_8	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INTMSK
		-	-	-	-	MODE	DER	DW	DL
		SR	TC	END	ER	SUS	TACT	RQST	EN
	CHCTRL_8	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	CLRINTMSK	SETINTMSK
		-	-	-	-	-	-	CLRSUS	SETSUS
		-	CLRTC	CLREND	CLRRQ	SWRST	STG	CLREN	SETEN
	CHCFG_8	DMS	REN	RSW	RSEL	SBE	-	-	DEM
		-	TM	DAD	SAD	DDS[3]	DDS[2]	DDS[1]	DDS[0]
		SDS[3]	SDS[2]	SDS[1]	SDS[0]	-	AM[2]	AM[1]	AM[0]
		-	LVL	HIEN	LOEN	REQD	SEL[2]	SEL[1]	SEL[0]
	CHITVL_8	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ITVL[15]	ITVL[14]	ITVL[13]	ITVL[12]	ITVL[11]	ITVL[10]	ITVL[9]	ITVL[8]
		ITVL[7]	ITVL[6]	ITVL[5]	ITVL[4]	ITVL[3]	ITVL[2]	ITVL[1]	ITVL[0]
	CHEXT_8	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DCA[3]	DCA[2]	DCA[1]	DCA[0]	-	-	-	-
		SCA[3]	SCA[2]	SCA[1]	SCA[0]	-	-	-	-
	NXLA_8	NXLA[31]	NXLA[30]	NXLA[29]	NXLA[28]	NXLA[27]	NXLA[26]	NXLA[25]	NXLA[24]
		NXLA[23]	NXLA[22]	NXLA[21]	NXLA[20]	NXLA[19]	NXLA[18]	NXLA[17]	NXLA[16]
		NXLA[15]	NXLA[14]	NXLA[13]	NXLA[12]	NXLA[11]	NXLA[10]	NXLA[9]	NXLA[8]
		NXLA[7]	NXLA[6]	NXLA[5]	NXLA[4]	NXLA[3]	NXLA[2]	NXLA[1]	NXLA[0]
	CRLA_8	CRLA[31]	CRLA[30]	CRLA[29]	CRLA[28]	CRLA[27]	CRLA[26]	CRLA[25]	CRLA[24]
		CRLA[23]	CRLA[22]	CRLA[21]	CRLA[20]	CRLA[19]	CRLA[18]	CRLA[17]	CRLA[16]
		CRLA[15]	CRLA[14]	CRLA[13]	CRLA[12]	CRLA[11]	CRLA[10]	CRLA[9]	CRLA[8]
		CRLA[7]	CRLA[6]	CRLA[5]	CRLA[4]	CRLA[3]	CRLA[2]	CRLA[1]	CRLA[0]
	N0SA_9	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N0DA_9	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N0TB_9	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Direct memory access controller	N1SA_9	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N1DA_9	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N1TB_9	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	CRSA_9	CRSA[31]	CRSA[30]	CRSA[29]	CRSA[28]	CRSA[27]	CRSA[26]	CRSA[25]	CRSA[24]
		CRSA[23]	CRSA[22]	CRSA[21]	CRSA[20]	CRSA[19]	CRSA[18]	CRSA[17]	CRSA[16]
		CRSA[15]	CRSA[14]	CRSA[13]	CRSA[12]	CRSA[11]	CRSA[10]	CRSA[9]	CRSA[8]
		CRSA[7]	CRSA[6]	CRSA[5]	CRSA[4]	CRSA[3]	CRSA[2]	CRSA[1]	CRSA[0]
	CRDA_9	CRDA[31]	CRDA[30]	CRDA[29]	CRDA[28]	CRDA[27]	CRDA[26]	CRDA[25]	CRDA[24]
		CRDA[23]	CRDA[22]	CRDA[21]	CRDA[20]	CRDA[19]	CRDA[18]	CRDA[17]	CRDA[16]
		CRDA[15]	CRDA[14]	CRDA[13]	CRDA[12]	CRDA[11]	CRDA[10]	CRDA[9]	CRDA[8]
		CRDA[7]	CRDA[6]	CRDA[5]	CRDA[4]	CRDA[3]	CRDA[2]	CRDA[1]	CRDA[0]
	CRTB_9	CRTB[31]	CRTB[30]	CRTB[29]	CRTB[28]	CRTB[27]	CRTB[26]	CRTB[25]	CRTB[24]
		CRTB[23]	CRTB[22]	CRTB[21]	CRTB[20]	CRTB[19]	CRTB[18]	CRTB[17]	CRTB[16]
		CRTB[15]	CRTB[14]	CRTB[13]	CRTB[12]	CRTB[11]	CRTB[10]	CRTB[9]	CRTB[8]
		CRTB[7]	CRTB[6]	CRTB[5]	CRTB[4]	CRTB[3]	CRTB[2]	CRTB[1]	CRTB[0]
	CHSTAT_9	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INTMSK
		-	-	-	-	MODE	DER	DW	DL
		SR	TC	END	ER	SUS	TACT	RQST	EN
	CHCTRL_9	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	CLRINTMSK	SETINTMSK
		-	-	-	-	-	-	CLRSUS	SETSUS
		-	CLRTC	CLREND	CLRRQ	SWRST	STG	CLREN	SETEN
	CHCFG_9	DMS	REN	RSW	RSEL	SBE	-	-	DEM
		-	TM	DAD	SAD	DDS[3]	DDS[2]	DDS[1]	DDS[0]
		SDS[3]	SDS[2]	SDS[1]	SDS[0]	-	AM[2]	AM[1]	AM[0]
		-	LVL	HIEN	LOEN	REQD	SEL[2]	SEL[1]	SEL[0]
	CHITVL_9	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ITVL[15]	ITVL[14]	ITVL[13]	ITVL[12]	ITVL[11]	ITVL[10]	ITVL[9]	ITVL[8]
		ITVL[7]	ITVL[6]	ITVL[5]	ITVL[4]	ITVL[3]	ITVL[2]	ITVL[1]	ITVL[0]
	CHEXT_9	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DCA[3]	DCA[2]	DCA[1]	DCA[0]	-	-	-	-
		SCA[3]	SCA[2]	SCA[1]	SCA[0]	-	-	-	-
	NXLA_9	NXLA[31]	NXLA[30]	NXLA[29]	NXLA[28]	NXLA[27]	NXLA[26]	NXLA[25]	NXLA[24]
		NXLA[23]	NXLA[22]	NXLA[21]	NXLA[20]	NXLA[19]	NXLA[18]	NXLA[17]	NXLA[16]
		NXLA[15]	NXLA[14]	NXLA[13]	NXLA[12]	NXLA[11]	NXLA[10]	NXLA[9]	NXLA[8]
		NXLA[7]	NXLA[6]	NXLA[5]	NXLA[4]	NXLA[3]	NXLA[2]	NXLA[1]	NXLA[0]
	CRLA_9	CRLA[31]	CRLA[30]	CRLA[29]	CRLA[28]	CRLA[27]	CRLA[26]	CRLA[25]	CRLA[24]
		CRLA[23]	CRLA[22]	CRLA[21]	CRLA[20]	CRLA[19]	CRLA[18]	CRLA[17]	CRLA[16]
		CRLA[15]	CRLA[14]	CRLA[13]	CRLA[12]	CRLA[11]	CRLA[10]	CRLA[9]	CRLA[8]
		CRLA[7]	CRLA[6]	CRLA[5]	CRLA[4]	CRLA[3]	CRLA[2]	CRLA[1]	CRLA[0]
	N0SA_10	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N0DA_10	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Direct memory access controller	N0TB_10	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	N1SA_10	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N1DA_10	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N1TB_10	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	CRSA_10	CRSA[31]	CRSA[30]	CRSA[29]	CRSA[28]	CRSA[27]	CRSA[26]	CRSA[25]	CRSA[24]
		CRSA[23]	CRSA[22]	CRSA[21]	CRSA[20]	CRSA[19]	CRSA[18]	CRSA[17]	CRSA[16]
		CRSA[15]	CRSA[14]	CRSA[13]	CRSA[12]	CRSA[11]	CRSA[10]	CRSA[9]	CRSA[8]
		CRSA[7]	CRSA[6]	CRSA[5]	CRSA[4]	CRSA[3]	CRSA[2]	CRSA[1]	CRSA[0]
	CRDA_10	CRDA[31]	CRDA[30]	CRDA[29]	CRDA[28]	CRDA[27]	CRDA[26]	CRDA[25]	CRDA[24]
		CRDA[23]	CRDA[22]	CRDA[21]	CRDA[20]	CRDA[19]	CRDA[18]	CRDA[17]	CRDA[16]
		CRDA[15]	CRDA[14]	CRDA[13]	CRDA[12]	CRDA[11]	CRDA[10]	CRDA[9]	CRDA[8]
		CRDA[7]	CRDA[6]	CRDA[5]	CRDA[4]	CRDA[3]	CRDA[2]	CRDA[1]	CRDA[0]
	CRTB_10	CRTB[31]	CRTB[30]	CRTB[29]	CRTB[28]	CRTB[27]	CRTB[26]	CRTB[25]	CRTB[24]
		CRTB[23]	CRTB[22]	CRTB[21]	CRTB[20]	CRTB[19]	CRTB[18]	CRTB[17]	CRTB[16]
		CRTB[15]	CRTB[14]	CRTB[13]	CRTB[12]	CRTB[11]	CRTB[10]	CRTB[9]	CRTB[8]
		CRTB[7]	CRTB[6]	CRTB[5]	CRTB[4]	CRTB[3]	CRTB[2]	CRTB[1]	CRTB[0]
	CHSTAT_10	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INTMSK
		-	-	-	-	MODE	DER	DW	DL
		SR	TC	END	ER	SUS	TACT	RQST	EN
	CHCTRL_10	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	CLRINTMSK	SETINTMSK
		-	-	-	-	-	-	CLRSUS	SETSUS
		-	CLRTC	CLREND	CLRRQ	SWRST	STG	CLREN	SETEN
	CHCFG_10	DMS	REN	RSW	RSEL	SBE	-	-	DEM
		-	TM	DAD	SAD	DDS[3]	DDS[2]	DDS[1]	DDS[0]
		SDS[3]	SDS[2]	SDS[1]	SDS[0]	-	AM[2]	AM[1]	AM[0]
		-	LVL	HIEN	LOEN	REQD	SEL[2]	SEL[1]	SEL[0]
	CHITVL_10	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ITVL[15]	ITVL[14]	ITVL[13]	ITVL[12]	ITVL[11]	ITVL[10]	ITVL[9]	ITVL[8]
		ITVL[7]	ITVL[6]	ITVL[5]	ITVL[4]	ITVL[3]	ITVL[2]	ITVL[1]	ITVL[0]
	CHEXT_10	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DCA[3]	DCA[2]	DCA[1]	DCA[0]	-	-	-	-
		SCA[3]	SCA[2]	SCA[1]	SCA[0]	-	-	-	-
	NXLA_10	NXLA[31]	NXLA[30]	NXLA[29]	NXLA[28]	NXLA[27]	NXLA[26]	NXLA[25]	NXLA[24]
		NXLA[23]	NXLA[22]	NXLA[21]	NXLA[20]	NXLA[19]	NXLA[18]	NXLA[17]	NXLA[16]
		NXLA[15]	NXLA[14]	NXLA[13]	NXLA[12]	NXLA[11]	NXLA[10]	NXLA[9]	NXLA[8]
		NXLA[7]	NXLA[6]	NXLA[5]	NXLA[4]	NXLA[3]	NXLA[2]	NXLA[1]	NXLA[0]
	CRLA_10	CRLA[31]	CRLA[30]	CRLA[29]	CRLA[28]	CRLA[27]	CRLA[26]	CRLA[25]	CRLA[24]
		CRLA[23]	CRLA[22]	CRLA[21]	CRLA[20]	CRLA[19]	CRLA[18]	CRLA[17]	CRLA[16]
		CRLA[15]	CRLA[14]	CRLA[13]	CRLA[12]	CRLA[11]	CRLA[10]	CRLA[9]	CRLA[8]
		CRLA[7]	CRLA[6]	CRLA[5]	CRLA[4]	CRLA[3]	CRLA[2]	CRLA[1]	CRLA[0]
	N0SA_11	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Direct memory access controller	N0DA_11	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N0TB_11	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	N1SA_11	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N1DA_11	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N1TB_11	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	CRSA_11	CRSA[31]	CRSA[30]	CRSA[29]	CRSA[28]	CRSA[27]	CRSA[26]	CRSA[25]	CRSA[24]
		CRSA[23]	CRSA[22]	CRSA[21]	CRSA[20]	CRSA[19]	CRSA[18]	CRSA[17]	CRSA[16]
		CRSA[15]	CRSA[14]	CRSA[13]	CRSA[12]	CRSA[11]	CRSA[10]	CRSA[9]	CRSA[8]
		CRSA[7]	CRSA[6]	CRSA[5]	CRSA[4]	CRSA[3]	CRSA[2]	CRSA[1]	CRSA[0]
	CRDA_11	CRDA[31]	CRDA[30]	CRDA[29]	CRDA[28]	CRDA[27]	CRDA[26]	CRDA[25]	CRDA[24]
		CRDA[23]	CRDA[22]	CRDA[21]	CRDA[20]	CRDA[19]	CRDA[18]	CRDA[17]	CRDA[16]
		CRDA[15]	CRDA[14]	CRDA[13]	CRDA[12]	CRDA[11]	CRDA[10]	CRDA[9]	CRDA[8]
		CRDA[7]	CRDA[6]	CRDA[5]	CRDA[4]	CRDA[3]	CRDA[2]	CRDA[1]	CRDA[0]
	CRTB_11	CRTB[31]	CRTB[30]	CRTB[29]	CRTB[28]	CRTB[27]	CRTB[26]	CRTB[25]	CRTB[24]
		CRTB[23]	CRTB[22]	CRTB[21]	CRTB[20]	CRTB[19]	CRTB[18]	CRTB[17]	CRTB[16]
		CRTB[15]	CRTB[14]	CRTB[13]	CRTB[12]	CRTB[11]	CRTB[10]	CRTB[9]	CRTB[8]
		CRTB[7]	CRTB[6]	CRTB[5]	CRTB[4]	CRTB[3]	CRTB[2]	CRTB[1]	CRTB[0]
	CHSTAT_11	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INTMSK
		-	-	-	-	MODE	DER	DW	DL
		SR	TC	END	ER	SUS	TACT	RQST	EN
	CHCTRL_11	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	CLRINTMSK	SETINTMSK
		-	-	-	-	-	-	CLRSUS	SETSUS
		-	CLRTC	CLREND	CLRRQ	SWRST	STG	CLREN	SETEN
	CHCFG_11	DMS	REN	RSW	RSEL	SBE	-	-	DEM
		-	TM	DAD	SAD	DDS[3]	DDS[2]	DDS[1]	DDS[0]
		SDS[3]	SDS[2]	SDS[1]	SDS[0]	-	AM[2]	AM[1]	AM[0]
		-	LVL	HIEN	LOEN	REQD	SEL[2]	SEL[1]	SEL[0]
	CHITVL_11	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ITVL[15]	ITVL[14]	ITVL[13]	ITVL[12]	ITVL[11]	ITVL[10]	ITVL[9]	ITVL[8]
		ITVL[7]	ITVL[6]	ITVL[5]	ITVL[4]	ITVL[3]	ITVL[2]	ITVL[1]	ITVL[0]
	CHEXT_11	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DCA[3]	DCA[2]	DCA[1]	DCA[0]	-	-	-	-
		SCA[3]	SCA[2]	SCA[1]	SCA[0]	-	-	-	-
	NXLA_11	NXLA[31]	NXLA[30]	NXLA[29]	NXLA[28]	NXLA[27]	NXLA[26]	NXLA[25]	NXLA[24]
		NXLA[23]	NXLA[22]	NXLA[21]	NXLA[20]	NXLA[19]	NXLA[18]	NXLA[17]	NXLA[16]
		NXLA[15]	NXLA[14]	NXLA[13]	NXLA[12]	NXLA[11]	NXLA[10]	NXLA[9]	NXLA[8]
		NXLA[7]	NXLA[6]	NXLA[5]	NXLA[4]	NXLA[3]	NXLA[2]	NXLA[1]	NXLA[0]
	CRLA_11	CRLA[31]	CRLA[30]	CRLA[29]	CRLA[28]	CRLA[27]	CRLA[26]	CRLA[25]	CRLA[24]
		CRLA[23]	CRLA[22]	CRLA[21]	CRLA[20]	CRLA[19]	CRLA[18]	CRLA[17]	CRLA[16]
		CRLA[15]	CRLA[14]	CRLA[13]	CRLA[12]	CRLA[11]	CRLA[10]	CRLA[9]	CRLA[8]
		CRLA[7]	CRLA[6]	CRLA[5]	CRLA[4]	CRLA[3]	CRLA[2]	CRLA[1]	CRLA[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Direct memory access controller	N0SA_12	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N0DA_12	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N0TB_12	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	N1SA_12	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N1DA_12	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N1TB_12	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	CRSA_12	CRSA[31]	CRSA[30]	CRSA[29]	CRSA[28]	CRSA[27]	CRSA[26]	CRSA[25]	CRSA[24]
		CRSA[23]	CRSA[22]	CRSA[21]	CRSA[20]	CRSA[19]	CRSA[18]	CRSA[17]	CRSA[16]
		CRSA[15]	CRSA[14]	CRSA[13]	CRSA[12]	CRSA[11]	CRSA[10]	CRSA[9]	CRSA[8]
		CRSA[7]	CRSA[6]	CRSA[5]	CRSA[4]	CRSA[3]	CRSA[2]	CRSA[1]	CRSA[0]
	CRDA_12	CRDA[31]	CRDA[30]	CRDA[29]	CRDA[28]	CRDA[27]	CRDA[26]	CRDA[25]	CRDA[24]
		CRDA[23]	CRDA[22]	CRDA[21]	CRDA[20]	CRDA[19]	CRDA[18]	CRDA[17]	CRDA[16]
		CRDA[15]	CRDA[14]	CRDA[13]	CRDA[12]	CRDA[11]	CRDA[10]	CRDA[9]	CRDA[8]
		CRDA[7]	CRDA[6]	CRDA[5]	CRDA[4]	CRDA[3]	CRDA[2]	CRDA[1]	CRDA[0]
	CRTB_12	CRTB[31]	CRTB[30]	CRTB[29]	CRTB[28]	CRTB[27]	CRTB[26]	CRTB[25]	CRTB[24]
		CRTB[23]	CRTB[22]	CRTB[21]	CRTB[20]	CRTB[19]	CRTB[18]	CRTB[17]	CRTB[16]
		CRTB[15]	CRTB[14]	CRTB[13]	CRTB[12]	CRTB[11]	CRTB[10]	CRTB[9]	CRTB[8]
		CRTB[7]	CRTB[6]	CRTB[5]	CRTB[4]	CRTB[3]	CRTB[2]	CRTB[1]	CRTB[0]
	CHSTAT_12	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INTMSK
		-	-	-	-	MODE	DER	DW	DL
		SR	TC	END	ER	SUS	TACT	RQST	EN
	CHCTRL_12	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	CLRINTMSK	SETINTMSK
		-	-	-	-	-	-	CLRSUS	SETSUS
		-	CLRTC	CLREND	CLRRQ	SWRST	STG	CLREN	SETEN
	CHCFG_12	DMS	REN	RSW	RSEL	SBE	-	-	DEM
		-	TM	DAD	SAD	DDS[3]	DDS[2]	DDS[1]	DDS[0]
		SDS[3]	SDS[2]	SDS[1]	SDS[0]	-	AM[2]	AM[1]	AM[0]
		-	LVL	HIEN	LOEN	REQD	SEL[2]	SEL[1]	SEL[0]
	CHITVL_12	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ITVL[15]	ITVL[14]	ITVL[13]	ITVL[12]	ITVL[11]	ITVL[10]	ITVL[9]	ITVL[8]
		ITVL[7]	ITVL[6]	ITVL[5]	ITVL[4]	ITVL[3]	ITVL[2]	ITVL[1]	ITVL[0]
	CHEXT_12	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DCA[3]	DCA[2]	DCA[1]	DCA[0]	-	-	-	-
		SCA[3]	SCA[2]	SCA[1]	SCA[0]	-	-	-	-
	NXLA_12	NXLA[31]	NXLA[30]	NXLA[29]	NXLA[28]	NXLA[27]	NXLA[26]	NXLA[25]	NXLA[24]
		NXLA[23]	NXLA[22]	NXLA[21]	NXLA[20]	NXLA[19]	NXLA[18]	NXLA[17]	NXLA[16]
		NXLA[15]	NXLA[14]	NXLA[13]	NXLA[12]	NXLA[11]	NXLA[10]	NXLA[9]	NXLA[8]
		NXLA[7]	NXLA[6]	NXLA[5]	NXLA[4]	NXLA[3]	NXLA[2]	NXLA[1]	NXLA[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Direct memory access controller	CRLA_12	CRLA[31]	CRLA[30]	CRLA[29]	CRLA[28]	CRLA[27]	CRLA[26]	CRLA[25]	CRLA[24]
		CRLA[23]	CRLA[22]	CRLA[21]	CRLA[20]	CRLA[19]	CRLA[18]	CRLA[17]	CRLA[16]
		CRLA[15]	CRLA[14]	CRLA[13]	CRLA[12]	CRLA[11]	CRLA[10]	CRLA[9]	CRLA[8]
		CRLA[7]	CRLA[6]	CRLA[5]	CRLA[4]	CRLA[3]	CRLA[2]	CRLA[1]	CRLA[0]
	N0SA_13	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N0DA_13	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N0TB_13	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	N1SA_13	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N1DA_13	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N1TB_13	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	CRSA_13	CRSA[31]	CRSA[30]	CRSA[29]	CRSA[28]	CRSA[27]	CRSA[26]	CRSA[25]	CRSA[24]
		CRSA[23]	CRSA[22]	CRSA[21]	CRSA[20]	CRSA[19]	CRSA[18]	CRSA[17]	CRSA[16]
		CRSA[15]	CRSA[14]	CRSA[13]	CRSA[12]	CRSA[11]	CRSA[10]	CRSA[9]	CRSA[8]
		CRSA[7]	CRSA[6]	CRSA[5]	CRSA[4]	CRSA[3]	CRSA[2]	CRSA[1]	CRSA[0]
	CRDA_13	CRDA[31]	CRDA[30]	CRDA[29]	CRDA[28]	CRDA[27]	CRDA[26]	CRDA[25]	CRDA[24]
		CRDA[23]	CRDA[22]	CRDA[21]	CRDA[20]	CRDA[19]	CRDA[18]	CRDA[17]	CRDA[16]
		CRDA[15]	CRDA[14]	CRDA[13]	CRDA[12]	CRDA[11]	CRDA[10]	CRDA[9]	CRDA[8]
		CRDA[7]	CRDA[6]	CRDA[5]	CRDA[4]	CRDA[3]	CRDA[2]	CRDA[1]	CRDA[0]
	CRTB_13	CRTB[31]	CRTB[30]	CRTB[29]	CRTB[28]	CRTB[27]	CRTB[26]	CRTB[25]	CRTB[24]
		CRTB[23]	CRTB[22]	CRTB[21]	CRTB[20]	CRTB[19]	CRTB[18]	CRTB[17]	CRTB[16]
		CRTB[15]	CRTB[14]	CRTB[13]	CRTB[12]	CRTB[11]	CRTB[10]	CRTB[9]	CRTB[8]
		CRTB[7]	CRTB[6]	CRTB[5]	CRTB[4]	CRTB[3]	CRTB[2]	CRTB[1]	CRTB[0]
	CHSTAT_13	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INTMSK
		-	-	-	-	MODE	DER	DW	DL
		SR	TC	END	ER	SUS	TACT	RQST	EN
	CHCTRL_13	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	CLRINTMSK	SETINTMSK
		-	-	-	-	-	-	CLRSUS	SETSUS
		-	CLRTC	CLREND	CLRRQ	SWRST	STG	CLREN	SETEN
	CHCFG_13	DMS	REN	RSW	RSEL	SBE	-	-	DEM
		-	TM	DAD	SAD	DDS[3]	DDS[2]	DDS[1]	DDS[0]
		SDS[3]	SDS[2]	SDS[1]	SDS[0]	-	AM[2]	AM[1]	AM[0]
		-	LVL	HIEN	LOEN	REQD	SEL[2]	SEL[1]	SEL[0]
	CHITVL_13	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ITVL[15]	ITVL[14]	ITVL[13]	ITVL[12]	ITVL[11]	ITVL[10]	ITVL[9]	ITVL[8]
		ITVL[7]	ITVL[6]	ITVL[5]	ITVL[4]	ITVL[3]	ITVL[2]	ITVL[1]	ITVL[0]
	CHEXT_13	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DCA[3]	DCA[2]	DCA[1]	DCA[0]	-	-	-	-
		SCA[3]	SCA[2]	SCA[1]	SCA[0]	-	-	-	-

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Direct memory access controller	NXLA_13	NXLA[31]	NXLA[30]	NXLA[29]	NXLA[28]	NXLA[27]	NXLA[26]	NXLA[25]	NXLA[24]
		NXLA[23]	NXLA[22]	NXLA[21]	NXLA[20]	NXLA[19]	NXLA[18]	NXLA[17]	NXLA[16]
		NXLA[15]	NXLA[14]	NXLA[13]	NXLA[12]	NXLA[11]	NXLA[10]	NXLA[9]	NXLA[8]
		NXLA[7]	NXLA[6]	NXLA[5]	NXLA[4]	NXLA[3]	NXLA[2]	NXLA[1]	NXLA[0]
	CRLA_13	CRLA[31]	CRLA[30]	CRLA[29]	CRLA[28]	CRLA[27]	CRLA[26]	CRLA[25]	CRLA[24]
		CRLA[23]	CRLA[22]	CRLA[21]	CRLA[20]	CRLA[19]	CRLA[18]	CRLA[17]	CRLA[16]
		CRLA[15]	CRLA[14]	CRLA[13]	CRLA[12]	CRLA[11]	CRLA[10]	CRLA[9]	CRLA[8]
		CRLA[7]	CRLA[6]	CRLA[5]	CRLA[4]	CRLA[3]	CRLA[2]	CRLA[1]	CRLA[0]
	N0SA_14	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N0DA_14	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N0TB_14	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	N1SA_14	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N1DA_14	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N1TB_14	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	CRSA_14	CRSA[31]	CRSA[30]	CRSA[29]	CRSA[28]	CRSA[27]	CRSA[26]	CRSA[25]	CRSA[24]
		CRSA[23]	CRSA[22]	CRSA[21]	CRSA[20]	CRSA[19]	CRSA[18]	CRSA[17]	CRSA[16]
		CRSA[15]	CRSA[14]	CRSA[13]	CRSA[12]	CRSA[11]	CRSA[10]	CRSA[9]	CRSA[8]
		CRSA[7]	CRSA[6]	CRSA[5]	CRSA[4]	CRSA[3]	CRSA[2]	CRSA[1]	CRSA[0]
	CRDA_14	CRDA[31]	CRDA[30]	CRDA[29]	CRDA[28]	CRDA[27]	CRDA[26]	CRDA[25]	CRDA[24]
		CRDA[23]	CRDA[22]	CRDA[21]	CRDA[20]	CRDA[19]	CRDA[18]	CRDA[17]	CRDA[16]
		CRDA[15]	CRDA[14]	CRDA[13]	CRDA[12]	CRDA[11]	CRDA[10]	CRDA[9]	CRDA[8]
		CRDA[7]	CRDA[6]	CRDA[5]	CRDA[4]	CRDA[3]	CRDA[2]	CRDA[1]	CRDA[0]
	CRTB_14	CRTB[31]	CRTB[30]	CRTB[29]	CRTB[28]	CRTB[27]	CRTB[26]	CRTB[25]	CRTB[24]
		CRTB[23]	CRTB[22]	CRTB[21]	CRTB[20]	CRTB[19]	CRTB[18]	CRTB[17]	CRTB[16]
		CRTB[15]	CRTB[14]	CRTB[13]	CRTB[12]	CRTB[11]	CRTB[10]	CRTB[9]	CRTB[8]
		CRTB[7]	CRTB[6]	CRTB[5]	CRTB[4]	CRTB[3]	CRTB[2]	CRTB[1]	CRTB[0]
	CHSTAT_14	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INTMSK
		-	-	-	-	MODE	DER	DW	DL
		SR	TC	END	ER	SUS	TACT	RQST	EN
	CHCTRL_14	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	CLRINTMSK	SETINTMSK
		-	-	-	-	-	-	CLRSUS	SETSUS
		-	CLRTC	CLREND	CLRRQ	SWRST	STG	CLREN	SETEN
	CHCFG_14	DMS	REN	RSW	RSEL	SBE	-	-	DEM
		-	TM	DAD	SAD	DDS[3]	DDS[2]	DDS[1]	DDS[0]
		SDS[3]	SDS[2]	SDS[1]	SDS[0]	-	AM[2]	AM[1]	AM[0]
		-	LVL	HIEN	LOEN	REQD	SEL[2]	SEL[1]	SEL[0]
	CHITVL_14	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ITVL[15]	ITVL[14]	ITVL[13]	ITVL[12]	ITVL[11]	ITVL[10]	ITVL[9]	ITVL[8]
		ITVL[7]	ITVL[6]	ITVL[5]	ITVL[4]	ITVL[3]	ITVL[2]	ITVL[1]	ITVL[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Direct memory access controller	CHEXT_14	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DCA[3]	DCA[2]	DCA[1]	DCA[0]	-	-	-	-
		SCA[3]	SCA[2]	SCA[1]	SCA[0]	-	-	-	-
	NXLA_14	NXLA[31]	NXLA[30]	NXLA[29]	NXLA[28]	NXLA[27]	NXLA[26]	NXLA[25]	NXLA[24]
		NXLA[23]	NXLA[22]	NXLA[21]	NXLA[20]	NXLA[19]	NXLA[18]	NXLA[17]	NXLA[16]
		NXLA[15]	NXLA[14]	NXLA[13]	NXLA[12]	NXLA[11]	NXLA[10]	NXLA[9]	NXLA[8]
		NXLA[7]	NXLA[6]	NXLA[5]	NXLA[4]	NXLA[3]	NXLA[2]	NXLA[1]	NXLA[0]
	CRLA_14	CRLA[31]	CRLA[30]	CRLA[29]	CRLA[28]	CRLA[27]	CRLA[26]	CRLA[25]	CRLA[24]
		CRLA[23]	CRLA[22]	CRLA[21]	CRLA[20]	CRLA[19]	CRLA[18]	CRLA[17]	CRLA[16]
		CRLA[15]	CRLA[14]	CRLA[13]	CRLA[12]	CRLA[11]	CRLA[10]	CRLA[9]	CRLA[8]
		CRLA[7]	CRLA[6]	CRLA[5]	CRLA[4]	CRLA[3]	CRLA[2]	CRLA[1]	CRLA[0]
	N0SA_15	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N0DA_15	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N0TB_15	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	N1SA_15	SA[31]	SA[30]	SA[29]	SA[28]	SA[27]	SA[26]	SA[25]	SA[24]
		SA[23]	SA[22]	SA[21]	SA[20]	SA[19]	SA[18]	SA[17]	SA[16]
		SA[15]	SA[14]	SA[13]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	N1DA_15	DA[31]	DA[30]	DA[29]	DA[28]	DA[27]	DA[26]	DA[25]	DA[24]
		DA[23]	DA[22]	DA[21]	DA[20]	DA[19]	DA[18]	DA[17]	DA[16]
		DA[15]	DA[14]	DA[13]	DA[12]	DA[11]	DA[10]	DA[9]	DA[8]
		DA[7]	DA[6]	DA[5]	DA[4]	DA[3]	DA[2]	DA[1]	DA[0]
	N1TB_15	TB[31]	TB[30]	TB[29]	TB[28]	TB[27]	TB[26]	TB[25]	TB[24]
		TB[23]	TB[22]	TB[21]	TB[20]	TB[19]	TB[18]	TB[17]	TB[16]
		TB[15]	TB[14]	TB[13]	TB[12]	TB[11]	TB[10]	TB[9]	TB[8]
		TB[7]	TB[6]	TB[5]	TB[4]	TB[3]	TB[2]	TB[1]	TB[0]
	CRSA_15	CRSA[31]	CRSA[30]	CRSA[29]	CRSA[28]	CRSA[27]	CRSA[26]	CRSA[25]	CRSA[24]
		CRSA[23]	CRSA[22]	CRSA[21]	CRSA[20]	CRSA[19]	CRSA[18]	CRSA[17]	CRSA[16]
		CRSA[15]	CRSA[14]	CRSA[13]	CRSA[12]	CRSA[11]	CRSA[10]	CRSA[9]	CRSA[8]
		CRSA[7]	CRSA[6]	CRSA[5]	CRSA[4]	CRSA[3]	CRSA[2]	CRSA[1]	CRSA[0]
	CRDA_15	CRDA[31]	CRDA[30]	CRDA[29]	CRDA[28]	CRDA[27]	CRDA[26]	CRDA[25]	CRDA[24]
		CRDA[23]	CRDA[22]	CRDA[21]	CRDA[20]	CRDA[19]	CRDA[18]	CRDA[17]	CRDA[16]
		CRDA[15]	CRDA[14]	CRDA[13]	CRDA[12]	CRDA[11]	CRDA[10]	CRDA[9]	CRDA[8]
		CRDA[7]	CRDA[6]	CRDA[5]	CRDA[4]	CRDA[3]	CRDA[2]	CRDA[1]	CRDA[0]
	CRTB_15	CRTB[31]	CRTB[30]	CRTB[29]	CRTB[28]	CRTB[27]	CRTB[26]	CRTB[25]	CRTB[24]
		CRTB[23]	CRTB[22]	CRTB[21]	CRTB[20]	CRTB[19]	CRTB[18]	CRTB[17]	CRTB[16]
		CRTB[15]	CRTB[14]	CRTB[13]	CRTB[12]	CRTB[11]	CRTB[10]	CRTB[9]	CRTB[8]
		CRTB[7]	CRTB[6]	CRTB[5]	CRTB[4]	CRTB[3]	CRTB[2]	CRTB[1]	CRTB[0]
	CHSTAT_15	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INTMSK
		-	-	-	-	MODE	DER	DW	DL
		SR	TC	END	ER	SUS	TACT	RQST	EN
	CHCTRL_15	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	CLRINTMSK	SETINTMSK
		-	-	-	-	-	-	CLRSUS	SETSUS
		-	CLRTC	CLREND	CLRRQ	SWRST	STG	CLREN	SETEN
	CHCFG_15	DMS	REN	RSW	RSEL	SBE	-	-	DEM
		-	TM	DAD	SAD	DDS[3]	DDS[2]	DDS[1]	DDS[0]
		SDS[3]	SDS[2]	SDS[1]	SDS[0]	-	AM[2]	AM[1]	AM[0]
		-	LVL	HIEN	LOEN	REQD	SEL[2]	SEL[1]	SEL[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Direct memory access controller	CHITVL_15	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ITVL[15]	ITVL[14]	ITVL[13]	ITVL[12]	ITVL[11]	ITVL[10]	ITVL[9]	ITVL[8]
		ITVL[7]	ITVL[6]	ITVL[5]	ITVL[4]	ITVL[3]	ITVL[2]	ITVL[1]	ITVL[0]
	CHEXT_15	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DCA[3]	DCA[2]	DCA[1]	DCA[0]	-	-	-	-
		SCA[3]	SCA[2]	SCA[1]	SCA[0]	-	-	-	-
	NXLA_15	NXLA[31]	NXLA[30]	NXLA[29]	NXLA[28]	NXLA[27]	NXLA[26]	NXLA[25]	NXLA[24]
		NXLA[23]	NXLA[22]	NXLA[21]	NXLA[20]	NXLA[19]	NXLA[18]	NXLA[17]	NXLA[16]
		NXLA[15]	NXLA[14]	NXLA[13]	NXLA[12]	NXLA[11]	NXLA[10]	NXLA[9]	NXLA[8]
		NXLA[7]	NXLA[6]	NXLA[5]	NXLA[4]	NXLA[3]	NXLA[2]	NXLA[1]	NXLA[0]
	CRLA_15	CRLA[31]	CRLA[30]	CRLA[29]	CRLA[28]	CRLA[27]	CRLA[26]	CRLA[25]	CRLA[24]
		CRLA[23]	CRLA[22]	CRLA[21]	CRLA[20]	CRLA[19]	CRLA[18]	CRLA[17]	CRLA[16]
		CRLA[15]	CRLA[14]	CRLA[13]	CRLA[12]	CRLA[11]	CRLA[10]	CRLA[9]	CRLA[8]
		CRLA[7]	CRLA[6]	CRLA[5]	CRLA[4]	CRLA[3]	CRLA[2]	CRLA[1]	CRLA[0]
	DCTRL_8_15	LWCA[3]	LWCA[2]	LWCA[1]	LWCA[0]	-	LWPR[2]	LWPR[1]	LWPR[0]
		LDCA[3]	LDCA[2]	LDCA[1]	LDCA[0]	-	LDPR[2]	LDPR[1]	LDPR[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	LVINT	PR
	DSTAT_EN_8_15	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8
	DSTAT_ER_8_15	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ER15	ER14	ER13	ER12	ER11	ER10	ER9	ER8
	DSTAT_END_8_15	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		END15	END14	END13	END12	END11	END10	END9	END8
	DSTAT_TC_8_15	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8
	DSTAT_SUS_8_15	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		SUS15	SUS14	SUS13	SUS12	SUS11	SUS10	SUS9	SUS8
	DMARS0	-	-	-	-	-	-	-	CH1_MID[6]
		CH1_MID[5]	CH1_MID[4]	CH1_MID[3]	CH1_MID[2]	CH1_MID[1]	CH1_MID[0]	CH1_RID[1]	CH1_RID[0]
		-	-	-	-	-	-	-	CH0_MID[6]
		CH0_MID[5]	CH0_MID[4]	CH0_MID[3]	CH0_MID[2]	CH0_MID[1]	CH0_MID[0]	CH0_RID[1]	CH0_RID[0]
	DMARS1	-	-	-	-	-	-	-	CH3_MID[6]
		CH3_MID[5]	CH3_MID[4]	CH3_MID[3]	CH3_MID[2]	CH3_MID[1]	CH3_MID[0]	CH3_RID[1]	CH3_RID[0]
		-	-	-	-	-	-	-	CH2_MID[6]
		CH2_MID[5]	CH2_MID[4]	CH2_MID[3]	CH2_MID[2]	CH2_MID[1]	CH2_MID[0]	CH2_RID[1]	CH2_RID[0]
	DMARS2	-	-	-	-	-	-	-	CH5_MID[6]
		CH5_MID[5]	CH5_MID[4]	CH5_MID[3]	CH5_MID[2]	CH5_MID[1]	CH5_MID[0]	CH5_RID[1]	CH5_RID[0]
		-	-	-	-	-	-	-	CH4_MID[6]
		CH4_MID[5]	CH4_MID[4]	CH4_MID[3]	CH4_MID[2]	CH4_MID[1]	CH4_MID[0]	CH4_RID[1]	CH4_RID[0]
	DMARS3	-	-	-	-	-	-	-	CH7_MID[6]
		CH7_MID[5]	CH7_MID[4]	CH7_MID[3]	CH7_MID[2]	CH7_MID[1]	CH7_MID[0]	CH7_RID[1]	CH7_RID[0]
		-	-	-	-	-	-	-	CH6_MID[6]
		CH6_MID[5]	CH6_MID[4]	CH6_MID[3]	CH6_MID[2]	CH6_MID[1]	CH6_MID[0]	CH6_RID[1]	CH6_RID[0]
	DMARS4	-	-	-	-	-	-	-	CH9_MID[6]
		CH9_MID[5]	CH9_MID[4]	CH9_MID[3]	CH9_MID[2]	CH9_MID[1]	CH9_MID[0]	CH9_RID[1]	CH9_RID[0]
		-	-	-	-	-	-	-	CH8_MID[6]
		CH8_MID[5]	CH8_MID[4]	CH8_MID[3]	CH8_MID[2]	CH8_MID[1]	CH8_MID[0]	CH8_RID[1]	CH8_RID[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Direct memory access controller	DMARS5	-	-	-	-	-	-	-	CH11_MID[6]
		CH11_MID[5]	CH11_MID[4]	CH11_MID[3]	CH11_MID[2]	CH11_MID[1]	CH11_MID[0]	CH11_RID[1]	CH11_RID[0]
		-	-	-	-	-	-	-	CH10_MID[6]
		CH10_MID[5]	CH10_MID[4]	CH10_MID[3]	CH10_MID[2]	CH10_MID[1]	CH10_MID[0]	CH10_RID[1]	CH10_RID[0]
	DMARS6	-	-	-	-	-	-	-	CH13_MID[6]
		CH13_MID[5]	CH13_MID[4]	CH13_MID[3]	CH13_MID[2]	CH13_MID[1]	CH13_MID[0]	CH13_RID[1]	CH13_RID[0]
		-	-	-	-	-	-	-	CH12_MID[6]
		CH12_MID[5]	CH12_MID[4]	CH12_MID[3]	CH12_MID[2]	CH12_MID[1]	CH12_MID[0]	CH12_RID[1]	CH12_RID[0]
	DMARS7	-	-	-	-	-	-	-	CH15_MID[6]
		CH15_MID[5]	CH15_MID[4]	CH15_MID[3]	CH15_MID[2]	CH15_MID[1]	CH15_MID[0]	CH15_RID[1]	CH15_RID[0]
		-	-	-	-	-	-	-	CH14_MID[6]
		CH14_MID[5]	CH14_MID[4]	CH14_MID[3]	CH14_MID[2]	CH14_MID[1]	CH14_MID[0]	CH14_RID[1]	CH14_RID[0]
Multi-function timer pulse unit 2	TCR_0	CCLR[2]	CCLR[1]	CCLR[0]	CKEG[1]	CKEG[0]	TPSC[2]	TPSC[1]	TPSC[0]
	TMDR_0	-	BFE	BFB	BFA	MD[3]	MD[2]	MD[1]	MD[0]
	TIORH_0	IOB[3]	IOB[2]	IOB[1]	IOB[0]	IOA[3]	IOA[2]	IOA[1]	IOA[0]
	TIORL_0	IOD[3]	IOD[2]	IOD[1]	IOD[0]	IOC[3]	IOC[2]	IOC[1]	IOC[0]
	TIER_0	TTGE	-	-	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
	TSR_0	-	-	-	TCFV	TGFD	TGFC	TGFB	TGFA
	TCNT_0								
	TGRA_0								
	TGRB_0								
	TGRC_0								
	TGRD_0								
	TGRE_0								
	TGRF_0								
	TIER2_0	TTGE2	-	-	-	-	-	TGIEF	TGIEE
	TSR2_0	-	-	-	-	-	-	TGFF	TGFE
	TBTM_0	-	-	-	-	-	TTSE	TTSB	TTSA
	TCR_1	-	CCLR[1]	CCLR[0]	CKEG[1]	CKEG[0]	TPSC[2]	TPSC[1]	TPSC[0]
	TMDR_1	-	-	-	-	MD[3]	MD[2]	MD[1]	MD[0]
	TIOR_1	IOB[3]	IOB[2]	IOB[1]	IOB[0]	IOA[3]	IOA[2]	IOA[1]	IOA[0]
	TIER_1	TTGE	-	TCIEU	TCIEV	-	-	TGIEB	TGIEA
	TSR_1	TCFD	-	TCFU	TCFV	-	-	TGFB	TGFA
	TCNT_1								
	TGRA_1								
	TGRB_1								
	TICCR	-	-	-	-	I2BE	I2AE	I1BE	I1AE
	TCR_2	-	CCLR[1]	CCLR[0]	CKEG[1]	CKEG[0]	TPSC[2]	TPSC[1]	TPSC[0]
	TMDR_2	-	-	-	-	MD[3]	MD[2]	MD[1]	MD[0]
	TIOR_2	IOB[3]	IOB[2]	IOB[1]	IOB[0]	IOA[3]	IOA[2]	IOA[1]	IOA[0]
	TIER_2	TTGE	-	TCIEU	TCIEV	-	-	TGIEB	TGIEA
	TSR_2	TCFD	-	TCFU	TCFV	-	-	TGFB	TGFA
	TCNT_2								
	TGRA_2								
	TGRB_2								
	TCR_3	CCLR[2]	CCLR[1]	CCLR[0]	CKEG[1]	CKEG[0]	TPSC[2]	TPSC[1]	TPSC[0]
	TMDR_3	-	-	BFB	BFA	MD[3]	MD[2]	MD[1]	MD[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Multi-function timer pulse unit 2	TIORH_3	IOB[3]	IOB[2]	IOB[1]	IOB[0]	IOA[3]	IOA[2]	IOA[1]	IOA[0]
	TIORL_3	IOD[3]	IOD[2]	IOD[1]	IOD[0]	IOC[3]	IOC[2]	IOC[1]	IOC[0]
	TIER_3	TTGE	-	-	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
	TSR_3	TCFD	-	-	TCFV	TGFD	TGFC	TGFB	TGFA
	TCNT_3								
	TGRA_3								
	TGRB_3								
	TGRC_3								
	TGRD_3								
	TBTM_3	-	-	-	-	-	-	TTSB	TTSA
	TCR_4	CCLR[2]	CCLR[1]	CCLR[0]	CKEG[1]	CKEG[0]	TPSC[2]	TPSC[1]	TPSC[0]
	TMDR_4	-	-	BFB	BFA	MD[3]	MD[2]	MD[1]	MD[0]
	TIORH_4	IOB[3]	IOB[2]	IOB[1]	IOB[0]	IOA[3]	IOA[2]	IOA[1]	IOA[0]
	TIORL_4	IOD[3]	IOD[2]	IOD[1]	IOD[0]	IOC[3]	IOC[2]	IOC[1]	IOC[0]
	TIER_4	TTGE	TTGE2	-	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
	TSR_4	TCFD	-	-	TCFV	TGFD	TGFC	TGFB	TGFA
	TCNT_4								
	TGRA_4								
	TGRB_4								
	TGRC_4								
	TGRD_4								
	TBTM_4	-	-	-	-	-	-	TTSB	TTSA
	TADCR	BF[1]	BF[0]	-	-	-	-	-	-
		UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE
	TADCORA_4								
	TADCORB_4								
	TADCOBRA_4								
	TADCOBRB_4								
	TSTR	CST4	CST3	-	-	-	CST2	CST1	CST0
	TSYR	SYNC4	SYNC3	-	-	-	SYNC2	SYNC1	SYNC0
	TRWER	-	-	-	-	-	-	-	RWE
	TOER	-	-	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
	TOCR1	-	PSYE	-	-	TOCL	TOCS	OLSN	OLSP
	TOCR2	BF[1]	BF[0]	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
	TGCR	-	BDC	N	P	FB	WF	VF	UF
	TCDR								
	TDDR								
	TCNTS								
	TCBR								

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Multi-function timer pulse unit 2	TITCR	T3AEN	3ACOR[2]	3ACOR[1]	3ACOR[0]	T4VEN	4VCOR[2]	4VCOR[1]	4VCOR[0]
	TITCNT	-	3ACNT[2]	3ACNT[1]	3ACNT[0]	-	4VCNT[2]	4VCNT[1]	4VCNT[0]
	TBTER	-	-	-	-	-	-	BTE[1]	BTE[0]
	TDER	-	-	-	-	-	-	-	TDER
	TWCR	CCE	-	-	-	-	-	-	WRE
	TOLBR	-	-	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
OS timer	OSTM0CMP	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	OSTM0CNT	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	OSTM0TE	-	-	-	-	-	-	-	OSTM0TE
	OSTM0TS	-	-	-	-	-	-	-	OSTM0TS
	OSTM0TT	-	-	-	-	-	-	-	OSTM0TT
	OSTM0CTL	-	-	-	-	-	-	OSTM0MD1	OSTM0MD0
	OSTM1CMP	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	OSTM1CNT	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	OSTM1TE	-	-	-	-	-	-	-	OSTM1TE
	OSTM1TS	-	-	-	-	-	-	-	OSTM1TS
	OSTM1TT	-	-	-	-	-	-	-	OSTM1TT
	OSTM1CTL	-	-	-	-	-	-	OSTM1MD1	OSTM1MD0
Watchdog timer	WTCNT								
	WTCSR	IOVF	WT/$\overline{IT}$	TME	-	-	CKS[2]	CKS[1]	CKS[0]
	WRCSR	WOVF	RSTE	-	-	-	-	-	-
Realtime clock	R64CNT	-	1Hz	2Hz	4Hz	8Hz	16Hz	32Hz	64Hz
	RSECCNT	-	10 seconds[2]	10 seconds[1]	10 seconds[0]	1 second[3]	1 second[2]	1 second[1]	1 second[0]
	RMINCNT	-	10 minutes[2]	10 minutes[1]	10 minutes[0]	1 minute[3]	1 minute[2]	1 minute[1]	1 minute[0]
	RHRCNT	-	-	10 hours[1]	10 hours[0]	1 hour[3]	1 hour[2]	1 hour[1]	1 hour[0]
	RWKCNT	-	-	-	-	-	Day[2]	Day[1]	Day[0]
	RDAYCNT	-	-	10 days[1]	10 days[0]	1 day[3]	1 day[2]	1 day[1]	1 day[0]
	RMONCNT	-	-	-	10 months	1 month[3]	1 month[2]	1 month[1]	1 month[0]
	RYRCNT	1000 years[3]	1000 years[2]	1000 years[1]	1000 years[0]	100 years[3]	100 years[2]	100 years[1]	100 years[0]
		10 years[3]	10 years[2]	10 years[1]	10 years[0]	1 year[3]	1 year[2]	1 year[1]	1 year[0]
	RSECAR	ENB	10 seconds[2]	10 seconds[1]	10 seconds[0]	1 second[3]	1 second[2]	1 second[1]	1 second[0]
	RMINAR	ENB	10 minutes[2]	10 minutes[1]	10 minutes[0]	1 minute[3]	1 minute[2]	1 minute[1]	1 minute[0]
	RHRAR	ENB	-	10 hours[1]	10 hours[0]	1 hour[3]	1 hour[2]	1 hour[1]	1 hour[0]
	RWKAR	ENB	-	-	-	-	Day[2]	Day[1]	Day[0]
	RDAYAR	ENB	-	10 days[1]	10 days[0]	1 day[3]	1 day[2]	1 day[1]	1 day[0]
	RMONAR	ENB	-	-	10 months	1 month[3]	1 month[2]	1 month[1]	1 month[0]
	RYRAR	1000 years[3]	1000 years[2]	1000 years[1]	1000 years[0]	100 years[3]	100 years[2]	100 years[1]	100 years[0]
		10 years[3]	10 years[2]	10 years[1]	10 years[0]	1 year[3]	1 year[2]	1 year[1]	1 year[0]
	RCR1	CF	-	-	CIE	AIE	-	-	AF
	RCR2	PEF	PES[2]	PES[1]	PES[0]	RTCEN	ADJ	RESET	START
	RCR3	ENB	-	-	-	-	-	-	-
	RCR5	-	-	-	-	-	-	RCKSEL[1]	RCKSEL[0]
	RFRH	SEL64	-	-	-	-	-	-	-
		-	-	-	-	-	RFC[18]	RFC[17]	RFC[16]
	RFRL	RFC[15]	RFC[14]	RFC[13]	RFC[12]	RFC[11]	RFC[10]	RFC[9]	RFC[8]
		RFC[7]	RFC[6]	RFC[5]	RFC[4]	RFC[3]	RFC[2]	RFC[1]	RFC[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Serial communication interface with FIFO	SCSMR_0	-	-	-	-	-	-	-	-
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	-	CKS[1]	CKS[0]
	SCBRR_0								
	SCSCR_0	-	-	-	-	-	-	-	-
		TIE	RIE	TE	RE	REIE	-	CKE[1]	CKE[0]
	SCFTDR_0								
	SCFSR_0	PER[3]	PER[2]	PER[1]	PER[0]	FER[3]	FER[2]	FER[1]	FER[0]
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_0								
	SCFCR_0	-	-	-	-	-	RSTRG[2]	RSTRG[1]	RSTRG[0]
		RTRG[1]	RTRG[0]	TTRG[1]	TTRG[0]	MCE	TFRST	RFRST	LOOP
	SCFDR_0	-	-	-	T[4]	T[3]	T[2]	T[1]	T[0]
		-	-	-	R[4]	R[3]	R[2]	R[1]	R[0]
	SCSPTR_0	-	-	-	-	-	-	-	-
		RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	ORER
	SCEMR_0	-	-	-	-	-	-	-	-
		BGDM	-	-	-	-	-	-	ABCS
	SCSMR_1	-	-	-	-	-	-	-	-
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	-	CKS[1]	CKS[0]
	SCBRR_1								
	SCSCR_1	-	-	-	-	-	-	-	-
		TIE	RIE	TE	RE	REIE	-	CKE[1]	CKE[0]
	SCFTDR_1								
	SCFSR_1	PER[3]	PER[2]	PER[1]	PER[0]	FER[3]	FER[2]	FER[1]	FER[0]
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_1								
	SCFCR_1	-	-	-	-	-	RSTRG[2]	RSTRG[1]	RSTRG[0]
		RTRG[1]	RTRG[0]	TTRG[1]	TTRG[0]	MCE	TFRST	RFRST	LOOP
	SCFDR_1	-	-	-	T[4]	T[3]	T[2]	T[1]	T[0]
		-	-	-	R[4]	R[3]	R[2]	R[1]	R[0]
	SCSPTR_1	-	-	-	-	-	-	-	-
		RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	ORER
	SCEMR_1	-	-	-	-	-	-	-	-
		BGDM	-	-	-	-	-	-	ABCS
	SCSMR_2	-	-	-	-	-	-	-	-
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	-	CKS[1]	CKS[0]
	SCBRR_2								
	SCSCR_2	-	-	-	-	-	-	-	-
		TIE	RIE	TE	RE	REIE	-	CKE[1]	CKE[0]
	SCFTDR_2								
	SCFSR_2	PER[3]	PER[2]	PER[1]	PER[0]	FER[3]	FER[2]	FER[1]	FER[0]
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_2								
	SCFCR_2	-	-	-	-	-	RSTRG[2]	RSTRG[1]	RSTRG[0]
		RTRG[1]	RTRG[0]	TTRG[1]	TTRG[0]	MCE	TFRST	RFRST	LOOP
	SCFDR_2	-	-	-	T[4]	T[3]	T[2]	T[1]	T[0]
		-	-	-	R[4]	R[3]	R[2]	R[1]	R[0]
	SCSPTR_2	-	-	-	-	-	-	-	-
		RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	ORER

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Serial communication interface with FIFO	SCEMR_2	-	-	-	-	-	-	-	-
		BGDM	-	-	-	-	-	-	ABCS
	SCSMR_3	-	-	-	-	-	-	-	-
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	-	CKS[1]	CKS[0]
	SCBRR_3								
	SCSCR_3	-	-	-	-	-	-	-	-
		TIE	RIE	TE	RE	REIE	-	CKE[1]	CKE[0]
	SCFTDR_3								
	SCFSR_3	PER[3]	PER[2]	PER[1]	PER[0]	FER[3]	FER[2]	FER[1]	FER[0]
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_3								
	SCFCR_3	-	-	-	-	-	RSTRG[2]	RSTRG[1]	RSTRG[0]
		RTRG[1]	RTRG[0]	TTRG[1]	TTRG[0]	MCE	TFRST	RFRST	LOOP
	SCFDR_3	-	-	-	T[4]	T[3]	T[2]	T[1]	T[0]
		-	-	-	R[4]	R[3]	R[2]	R[1]	R[0]
	SCSPTR_3	-	-	-	-	-	-	-	-
		RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	ORER
	SCEMR_3	-	-	-	-	-	-	-	-
		BGDM	-	-	-	-	-	-	ABCS
	SCSMR_4	-	-	-	-	-	-	-	-
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	-	CKS[1]	CKS[0]
	SCBRR_4								
	SCSCR_4	-	-	-	-	-	-	-	-
		TIE	RIE	TE	RE	REIE	-	CKE[1]	CKE[0]
	SCFTDR_4								
	SCFSR_4	PER[3]	PER[2]	PER[1]	PER[0]	FER[3]	FER[2]	FER[1]	FER[0]
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_4								
	SCFCR_4	-	-	-	-	-	RSTRG[2]	RSTRG[1]	RSTRG[0]
		RTRG[1]	RTRG[0]	TTRG[1]	TTRG[0]	MCE	TFRST	RFRST	LOOP
	SCFDR_4	-	-	-	T[4]	T[3]	T[2]	T[1]	T[0]
		-	-	-	R[4]	R[3]	R[2]	R[1]	R[0]
	SCSPTR_4	-	-	-	-	-	-	-	-
		RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_4	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	ORER
	SCEMR_4	-	-	-	-	-	-	-	-
		BGDM	-	-	-	-	-	-	ABCS
Serial communications interface	SMR0	CM	CHR	PE	PM	STOP	MP	CKS[1]	CKS[0]
	SMR0	GM	BLK	PE	PM	BCP[1]	BCP[0]	CKS[1]	CKS[0]
	BRR0	BRR[7]	BRR[6]	BRR[5]	BRR[4]	BRR[3]	BRR[2]	BRR[1]	BRR[0]
	SCR0	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1]	CKE[0]
	TDR0	TDR[7]	TDR[6]	TDR[5]	TDR[4]	TDR[3]	TDR[2]	TDR[1]	TDR[0]
	SSR0	-	-	ORER	FER	PER	TEND	MPB	MPBT
	SSR0	-	-	ORER	ERS	PER	TEND	MPB	MPBT
	RDR0	RDR[7]	RDR[6]	RDR[5]	RDR[4]	RDR[3]	RDR[2]	RDR[1]	RDR[0]
	SCMR0	BCP2	-	-	-	SDIR	SINV	-	SMIF
	SEMR0	-	-	NFEN	ABCS	-	-	-	-
	SNFR0	-	-	-	-	-	NFCS[2]	NFCS[1]	NFCS[0]
	SECR0	-	-	-	-	-	-	CTSE	-
	SMR1	CM	CHR	PE	PM	STOP	MP	CKS[1]	CKS[0]
	SMR1	GM	BLK	PE	PM	BCP[1]	BCP[0]	CKS[1]	CKS[0]
	BRR1	BRR[7]	BRR[6]	BRR[5]	BRR[4]	BRR[3]	BRR[2]	BRR[1]	BRR[0]
	SCR1	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1]	CKE[0]
	TDR1	TDR[7]	TDR[6]	TDR[5]	TDR[4]	TDR[3]	TDR[2]	TDR[1]	TDR[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Serial communications interface	SSR1	-	-	ORER	FER	PER	TEND	MPB	MPBT
	SSR1	-	-	ORER	ERS	PER	TEND	MPB	MPBT
	RDR1	RDR[7]	RDR[6]	RDR[5]	RDR[4]	RDR[3]	RDR[2]	RDR[1]	RDR[0]
	SCMR1	BCP2	-	-	-	SDIR	SINV	-	SMIF
	SEMR1	-	-	NFEN	ABCS	-	-	-	-
	SNFR1	-	-	-	-	-	NFCS[2]	NFCS[1]	NFCS[0]
	SECR1	-	-	-	-	-	-	CTSE	-
	IRCR	IRE	IRCKS[2]	IRCKS[1]	IRCKS[0]	IRTXINV	IRRXINV	-	-
Renesas serial peripheral interface	SPCR_0	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	-	-
	SSLP_0	-	-	-	-	-	-	-	SSL0P
	SPPCR_0	-	-	MOIFE	MOIFV	-	-	-	SPLP
	SPSR_0	SPRF	TEND	SPTEF	-	-	MODF	-	OVRF
	SPDR_0	SPD31	SPD30	SPD29	SPD28	SPD27	SPD26	SPD25	SPD24
		SPD23	SPD22	SPD21	SPD20	SPD19	SPD18	SPD17	SPD16
		SPD15	SPD14	SPD13	SPD12	SPD11	SPD10	SPD9	SPD8
		SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
	SPSCR_0	-	-	-	-	-	-	SPSLN1	SPSLN0
	SPSSR_0	-	-	-	-	-	-	SPCP1	SPCP0
	SPBR_0	SPR7	SPR6	SPR5	SPR4	SPR3	SPR2	SPR1	SPR0
	SPDCR_0	TXDMY	SPLW1	SPLW0	-	-	-	-	-
	SPCKD_0	-	-	-	-	-	SCKDL2	SCKDL1	SCKDL0
	SSLND_0	-	-	-	-	-	SLNDL2	SLNDL1	SLNDL0
	SPND_0	-	-	-	-	-	SPNDL2	SPNDL1	SPNDL0
	SPCMD0_0	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	-	-	-	BRDV1	BRDV0	CPOL	CPHA
	SPCMD1_0	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	-	-	-	BRDV1	BRDV0	CPOL	CPHA
	SPCMD2_0	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	-	-	-	BRDV1	BRDV0	CPOL	CPHA
	SPCMD3_0	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	-	-	-	BRDV1	BRDV0	CPOL	CPHA
	SPBFCR_0	TXRST	RXRST	TXTRG[1]	TXTRG[0]	-	RXTRG[2]	RXTRG[1]	RXTRG[0]
	SPBFDR_0	-	-	-	-	T[3]	T[2]	T[1]	T[0]
		-	-	R[5]	R[4]	R[3]	R[2]	R[1]	R[0]
	SPCR_1	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	-	-
	SSLP_1	-	-	-	-	-	-	-	SSL0P
	SPPCR_1	-	-	MOIFE	MOIFV	-	-	-	SPLP
	SPSR_1	SPRF	TEND	SPTEF	-	-	MODF	-	OVRF
	SPDR_1	SPD31	SPD30	SPD29	SPD28	SPD27	SPD26	SPD25	SPD24
		SPD23	SPD22	SPD21	SPD20	SPD19	SPD18	SPD17	SPD16
		SPD15	SPD14	SPD13	SPD12	SPD11	SPD10	SPD9	SPD8
		SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
	SPSCR_1	-	-	-	-	-	-	SPSLN1	SPSLN0
	SPSSR_1	-	-	-	-	-	-	SPCP1	SPCP0
	SPBR_1	SPR7	SPR6	SPR5	SPR4	SPR3	SPR2	SPR1	SPR0
	SPDCR_1	TXDMY	SPLW1	SPLW0	-	-	-	-	-
	SPCKD_1	-	-	-	-	-	SCKDL2	SCKDL1	SCKDL0
	SSLND_1	-	-	-	-	-	SLNDL2	SLNDL1	SLNDL0
	SPND_1	-	-	-	-	-	SPNDL2	SPNDL1	SPNDL0
	SPCMD0_1	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	-	-	-	BRDV1	BRDV0	CPOL	CPHA
	SPCMD1_1	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	-	-	-	BRDV1	BRDV0	CPOL	CPHA
	SPCMD2_1	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	-	-	-	BRDV1	BRDV0	CPOL	CPHA
	SPCMD3_1	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	-	-	-	BRDV1	BRDV0	CPOL	CPHA
	SPBFCR_1	TXRST	RXRST	TXTRG[1]	TXTRG[0]	-	RXTRG[2]	RXTRG[1]	RXTRG[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Renesas serial peripheral interface	SPBFDR_1	-	-	-	-	T[3]	T[2]	T[1]	T[0]
		-	-	R[5]	R[4]	R[3]	R[2]	R[1]	R[0]
	SPCR_2	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	-	-
	SSLP_2	-	-	-	-	-	-	-	SSL0P
	SPPCR_2	-	-	MOIFE	MOIFV	-	-	-	SPLP
	SPSR_2	SPRF	TEND	SPTEF	-	-	MODF	-	OVRF
	SPDR_2	SPD31	SPD30	SPD29	SPD28	SPD27	SPD26	SPD25	SPD24
		SPD23	SPD22	SPD21	SPD20	SPD19	SPD18	SPD17	SPD16
		SPD15	SPD14	SPD13	SPD12	SPD11	SPD10	SPD9	SPD8
		SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
	SPSCR_2	-	-	-	-	-	-	SPSLN1	SPSLN0
	SPSSR_2	-	-	-	-	-	-	SPCP1	SPCP0
	SPBR_2	SPR7	SPR6	SPR5	SPR4	SPR3	SPR2	SPR1	SPR0
	SPDCR_2	TXDMY	SPLW1	SPLW0	-	-	-	-	-
	SPCKD_2	-	-	-	-	-	SCKDL2	SCKDL1	SCKDL0
	SSLND_2	-	-	-	-	-	SLNDL2	SLNDL1	SLNDL0
	SPND_2	-	-	-	-	-	SPNDL2	SPNDL1	SPNDL0
	SPCMD0_2	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	-	-	-	BRDV1	BRDV0	CPOL	CPHA
	SPCMD1_2	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	-	-	-	BRDV1	BRDV0	CPOL	CPHA
	SPCMD2_2	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	-	-	-	BRDV1	BRDV0	CPOL	CPHA
	SPCMD3_2	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	-	-	-	BRDV1	BRDV0	CPOL	CPHA
	SPBFCR_2	TXRST	RXRST	TXTRG[1]	TXTRG[0]	-	RXTRG[2]	RXTRG[1]	RXTRG[0]
	SPBFDR_2	-	-	-	-	T[3]	T[2]	T[1]	T[0]
		-	-	R[5]	R[4]	R[3]	R[2]	R[1]	R[0]
SPI multi I/O bus controller	CMNCR_0	MD	-	-	-	-	-	-	SFDE
		MOIIO3[1]	MOIIO3[0]	MOIIO2[1]	MOIIO2[0]	MOIIO1[1]	MOIIO1[0]	MOIIO0[1]	MOIIO0[0]
		IO3FV[1]	IO3FV[0]	IO2FV[1]	IO2FV[0]	-	-	IO0FV[1]	IO0FV[0]
		-	CPHAT	CPHAR	SSLP	CPOL	-	BSZ[1]	BSZ[0]
	SSLDR_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	SPNDL[2]	SPNDL[1]	SPNDL[0]
		-	-	-	-	-	SLNDL[2]	SLNDL[1]	SLNDL[0]
		-	-	-	-	-	SCKDL[2]	SCKDL[1]	SCKDL[0]
	SPBCR_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		SPBR[7]	SPBR[6]	SPBR[5]	SPBR[4]	SPBR[3]	SPBR[2]	SPBR[1]	SPBR[0]
		-	-	-	-	-	-	BRDV[1]	BRDV[0]
	DRCR_0	-	-	-	-	-	-	-	SSLN
		-	-	-	-	RBURST[3]	RBURST[2]	RBURST[1]	RBURST[0]
		-	-	-	-	-	-	RCF	RBE
		-	-	-	-	-	-	-	SSLE
	DRCMR_0	-	-	-	-	-	-	-	-
		CMD[7]	CMD[6]	CMD[5]	CMD[4]	CMD[3]	CMD[2]	CMD[1]	CMD[0]
		-	-	-	-	-	-	-	-
		OCMD[7]	OCMD[6]	OCMD[5]	OCMD[4]	OCMD[3]	OCMD[2]	OCMD[1]	OCMD[0]
	DREAR_0	-	-	-	-	-	-	-	-
		EAV[7]	EAV[6]	EAV[5]	EAV[4]	EAV[3]	EAV[2]	EAV[1]	EAV[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	-	EAC[2]	EAC[1]	EAC[0]
	DROPR_0	OPD3[7]	OPD3[6]	OPD3[5]	OPD3[4]	OPD3[3]	OPD3[2]	OPD3[1]	OPD3[0]
		OPD2[7]	OPD2[6]	OPD2[5]	OPD2[4]	OPD2[3]	OPD2[2]	OPD2[1]	OPD2[0]
		OPD1[7]	OPD1[6]	OPD1[5]	OPD1[4]	OPD1[3]	OPD1[2]	OPD1[1]	OPD1[0]
		OPD0[7]	OPD0[6]	OPD0[5]	OPD0[4]	OPD0[3]	OPD0[2]	OPD0[1]	OPD0[0]
	DRENR_0	CDB[1]	CDB[0]	OCDB[1]	OCDB[0]	-	-	ADB[1]	ADB[0]
		-	-	OPDB[1]	OPDB[0]	-	-	DRDB[1]	DRDB[0]
		DME	CDE	-	OCDE	ADE[3]	ADE[2]	ADE[1]	ADE[0]
		OPDE[3]	OPDE[2]	OPDE[1]	OPDE[0]	-	-	-	-

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
SPI multi I/O bus controller	SMCR_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	SSLKP
		-	-	-	-	-	SPIRE	SPIWE	SPIE
	SMCMR_0	-	-	-	-	-	-	-	-
		CMD[7]	CMD[6]	CMD[5]	CMD[4]	CMD[3]	CMD[2]	CMD[1]	CMD[0]
		-	-	-	-	-	-	-	-
		OCMD[7]	OCMD[6]	OCMD[5]	OCMD[4]	OCMD[3]	OCMD[2]	OCMD[1]	OCMD[0]
	SMADR_0	ADR[31]	ADR[30]	ADR[29]	ADR[28]	ADR[27]	ADR[26]	ADR[25]	ADR[24]
		ADR[23]	ADR[22]	ADR[21]	ADR[20]	ADR[19]	ADR[18]	ADR[17]	ADR[16]
		ADR[15]	ADR[14]	ADR[13]	ADR[12]	ADR[11]	ADR[10]	ADR[9]	ADR[8]
		ADR[7]	ADR[6]	ADR[5]	ADR[4]	ADR[3]	ADR[2]	ADR[1]	ADR[0]
	SMOPR_0	OPD3[7]	OPD3[6]	OPD3[5]	OPD3[4]	OPD3[3]	OPD3[2]	OPD3[1]	OPD3[0]
		OPD2[7]	OPD2[6]	OPD2[5]	OPD2[4]	OPD2[3]	OPD2[2]	OPD2[1]	OPD2[0]
		OPD1[7]	OPD1[6]	OPD1[5]	OPD1[4]	OPD1[3]	OPD1[2]	OPD1[1]	OPD1[0]
		OPD0[7]	OPD0[6]	OPD0[5]	OPD0[4]	OPD0[3]	OPD0[2]	OPD0[1]	OPD0[0]
	SMENR_0	CDB[1]	CDB[0]	OCDB[1]	OCDB[0]	-	-	ADB[1]	ADB[0]
		-	-	OPDB[1]	OPDB[0]	-	-	SPIDB[1]	SPIDB[0]
		DME	CDE	-	OCDE	ADE[3]	ADE[2]	ADE[1]	ADE[0]
		OPDE[3]	OPDE[2]	OPDE[1]	OPDE[0]	SPIDE[3]	SPIDE[2]	SPIDE[1]	SPIDE[0]
	SMRDR0_0	RDATA0[31]	RDATA0[30]	RDATA0[29]	RDATA0[28]	RDATA0[27]	RDATA0[26]	RDATA0[25]	RDATA0[24]
		RDATA0[23]	RDATA0[22]	RDATA0[21]	RDATA0[20]	RDATA0[19]	RDATA0[18]	RDATA0[17]	RDATA0[16]
		RDATA0[15]	RDATA0[14]	RDATA0[13]	RDATA0[12]	RDATA0[11]	RDATA0[10]	RDATA0[9]	RDATA0[8]
		RDATA0[7]	RDATA0[6]	RDATA0[5]	RDATA0[4]	RDATA0[3]	RDATA0[2]	RDATA0[1]	RDATA0[0]
	SMRDR1_0	RDATA1[31]	RDATA1[30]	RDATA1[29]	RDATA1[28]	RDATA1[27]	RDATA1[26]	RDATA1[25]	RDATA1[24]
		RDATA1[23]	RDATA1[22]	RDATA1[21]	RDATA1[20]	RDATA1[19]	RDATA1[18]	RDATA1[17]	RDATA1[16]
		RDATA1[15]	RDATA1[14]	RDATA1[13]	RDATA1[12]	RDATA1[11]	RDATA1[10]	RDATA1[9]	RDATA1[8]
		RDATA1[7]	RDATA1[6]	RDATA1[5]	RDATA1[4]	RDATA1[3]	RDATA1[2]	RDATA1[1]	RDATA1[0]
	SMWDR0_0	WDATA0[31]	WDATA0[30]	WDATA0[29]	WDATA0[28]	WDATA0[27]	WDATA0[26]	WDATA0[25]	WDATA0[24]
		WDATA0[23]	WDATA0[22]	WDATA0[21]	WDATA0[20]	WDATA0[19]	WDATA0[18]	WDATA0[17]	WDATA0[16]
		WDATA0[15]	WDATA0[14]	WDATA0[13]	WDATA0[12]	WDATA0[11]	WDATA0[10]	WDATA0[9]	WDATA0[8]
		WDATA0[7]	WDATA0[6]	WDATA0[5]	WDATA0[4]	WDATA0[3]	WDATA0[2]	WDATA0[1]	WDATA0[0]
	SMWDR1_0	WDATA1[31]	WDATA1[30]	WDATA1[29]	WDATA1[28]	WDATA1[27]	WDATA1[26]	WDATA1[25]	WDATA1[24]
		WDATA1[23]	WDATA1[22]	WDATA1[21]	WDATA1[20]	WDATA1[19]	WDATA1[18]	WDATA1[17]	WDATA1[16]
		WDATA1[15]	WDATA1[14]	WDATA1[13]	WDATA1[12]	WDATA1[11]	WDATA1[10]	WDATA1[9]	WDATA1[8]
		WDATA1[7]	WDATA1[6]	WDATA1[5]	WDATA1[4]	WDATA1[3]	WDATA1[2]	WDATA1[1]	WDATA1[0]
	CMNSR_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	SSLF	TEND
	CKDLY_0*2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		GB[7]	GB[6]	GB[5]	GB[4]	GB[3]	GB[2]	GB[1]	GB[0]
						CKDLY[3]	CKDLY[2]	CKDLY[1]	CKDLY[0]
	DRDMCR_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	DMDB[1]	DMDB[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	-	DMCYC[2]	DMCYC[1]	DMCYC[0]
	DRDRENR_0*2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	ADDRE
		-	-	-	OPDRE	-	-	-	DRDRE
	SMDMCR_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	DMDB[1]	DMDB[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	-	DMCYC[2]	DMCYC[1]	DMCYC[0]
	SMDRENR_0*2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	ADDRE
		-	-	-	OPDRE	-	-	-	SPIDRE

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
SPI multi I/O bus controller	SPODLY_0*2	GB[7]	GB[6]	GB[5]	GB[4]	GB[3]	GB[2]	GB[1]	GB[0]
		-	-	-	-	-	-	-	-
		SPODLY[15]	SPODLY[14]	SPODLY[13]	SPODLY[12]	SPODLY[11]	SPODLY[10]	SPODLY[9]	SPODLY[8]
		SPODLY[7]	SPODLY[6]	SPODLY[5]	SPODLY[4]	SPODLY[3]	SPODLY[2]	SPODLY[1]	SPODLY[0]
I²C bus interface	RIIC0CR1	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
	RIIC0CR2	BBSY	MST	TRS	-	SP	RS	ST	-
	RIIC0MR1	-	CKS[2]	CKS[1]	CKS[0]	BCWP	BC[2]	BC[1]	BC[0]
	RIIC0MR2	DLCS	SDDL[2]	SDDL[1]	SDDL[0]	-	TMOH	TMOL	TMOS
	RIIC0MR3	SMBE	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1]	NF[0]
	RIIC0FER	-	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
	RIIC0SER	HOAE	-	DIDE	-	GCE	SAR2	SAR1	SAR0
	RIIC0IER	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
	RIIC0SR1	HOA	-	DID	-	GCA	AAS2	AAS1	AAS0
	RIIC0SR2	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
	RIIC0SAR0	FS0	-	-	-	-	-	SVA[9]	SVA[8]
		SVA[7]	SVA[6]	SVA[5]	SVA[4]	SVA[3]	SVA[2]	SVA[1]	SVA0

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
I²C bus interface	RIIC0SAR1	FS1	-	-	-	-	-	SVA[9]	SVA[8]
		SVA[7]	SVA[6]	SVA[5]	SVA[4]	SVA[3]	SVA[2]	SVA[1]	SVA0
	RIIC0SAR2	FS2	-	-	-	-	-	SVA[9]	SVA[8]
		SVA[7]	SVA[6]	SVA[5]	SVA[4]	SVA[3]	SVA[2]	SVA[1]	SVA0
	RIIC0BRL	-	-	-	BRL[4]	BRL[3]	BRL[2]	BRL[1]	BRL[0]
	RIIC0BRH	-	-	-	BRH[4]	BRH[3]	BRH[2]	BRH[1]	BRH[0]
	RIIC0DRT	DRT[7]	DRT[6]	DRT[5]	DRT[4]	DRT[3]	DRT[2]	DRT[1]	DRT[0]
	RIIC0DRR	DRR[7]	DRR[6]	DRR[5]	DRR[4]	DRR[3]	DRR[2]	DRR[1]	DRR[0]
	RIIC1CR1	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
	RIIC1CR2	BBSY	MST	TRS	-	SP	RS	ST	-
	RIIC1MR1	-	CKS[2]	CKS[1]	CKS[0]	BCWP	BC[2]	BC[1]	BC[0]
	RIIC1MR2	DLCS	SDDL[2]	SDDL[1]	SDDL[0]	-	TMOH	TMOL	TMOS
	RIIC1MR3	SMBE	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1]	NF[0]
	RIIC1FER	-	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
	RIIC1SER	HOAE	-	DIDE	-	GCE	SAR2	SAR1	SAR0
	RIIC1IER	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
	RIIC1SR1	HOA	-	DID	-	GCA	AAS2	AAS1	AAS0
	RIIC1SR2	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
	RIIC1SAR0	FS0	-	-	-	-	-	SVA[9]	SVA[8]
		SVA[7]	SVA[6]	SVA[5]	SVA[4]	SVA[3]	SVA[2]	SVA[1]	SVA0
	RIIC1SAR1	FS1	-	-	-	-	-	SVA[9]	SVA[8]
		SVA[7]	SVA[6]	SVA[5]	SVA[4]	SVA[3]	SVA[2]	SVA[1]	SVA0
	RIIC1SAR2	FS2	-	-	-	-	-	SVA[9]	SVA[8]
		SVA[7]	SVA[6]	SVA[5]	SVA[4]	SVA[3]	SVA[2]	SVA[1]	SVA0
	RIIC1BRL	-	-	-	BRL[4]	BRL[3]	BRL[2]	BRL[1]	BRL[0]
	RIIC1BRH	-	-	-	BRH[4]	BRH[3]	BRH[2]	BRH[1]	BRH[0]
	RIIC1DRT	DRT[7]	DRT[6]	DRT[5]	DRT[4]	DRT[3]	DRT[2]	DRT[1]	DRT[0]
	RIIC1DRR	DRR[7]	DRR[6]	DRR[5]	DRR[4]	DRR[3]	DRR[2]	DRR[1]	DRR[0]
	RIIC2CR1	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
	RIIC2CR2	BBSY	MST	TRS	-	SP	RS	ST	-
	RIIC2MR1	-	CKS[2]	CKS[1]	CKS[0]	BCWP	BC[2]	BC[1]	BC[0]
	RIIC2MR2	DLCS	SDDL[2]	SDDL[1]	SDDL[0]	-	TMOH	TMOL	TMOS
	RIIC2MR3	SMBE	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1]	NF[0]
	RIIC2FER	-	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
	RIIC2SER	HOAE	-	DIDE	-	GCE	SAR2	SAR1	SAR0
	RIIC2IER	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
	RIIC2SR1	HOA	-	DID	-	GCA	AAS2	AAS1	AAS0
	RIIC2SR2	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
	RIIC2SAR0	FS0	-	-	-	-	-	SVA[9]	SVA[8]
		SVA[7]	SVA[6]	SVA[5]	SVA[4]	SVA[3]	SVA[2]	SVA[1]	SVA0
	RIIC2SAR1	FS1	-	-	-	-	-	SVA[9]	SVA[8]
		SVA[7]	SVA[6]	SVA[5]	SVA[4]	SVA[3]	SVA[2]	SVA[1]	SVA0
	RIIC2SAR2	FS2	-	-	-	-	-	SVA[9]	SVA[8]
		SVA[7]	SVA[6]	SVA[5]	SVA[4]	SVA[3]	SVA[2]	SVA[1]	SVA0
	RIIC2BRL	-	-	-	BRL[4]	BRL[3]	BRL[2]	BRL[1]	BRL[0]
	RIIC2BRH	-	-	-	BRH[4]	BRH[3]	BRH[2]	BRH[1]	BRH[0]
	RIIC2DRT	DRT[7]	DRT[6]	DRT[5]	DRT[4]	DRT[3]	DRT[2]	DRT[1]	DRT[0]
	RIIC2DRR	DRR[7]	DRR[6]	DRR[5]	DRR[4]	DRR[3]	DRR[2]	DRR[1]	DRR[0]
	RIIC3CR1	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
	RIIC3CR2	BBSY	MST	TRS	-	SP	RS	ST	-
	RIIC3MR1	-	CKS[2]	CKS[1]	CKS[0]	BCWP	BC[2]	BC[1]	BC[0]
	RIIC3MR2	DLCS	SDDL[2]	SDDL[1]	SDDL[0]	-	TMOH	TMOL	TMOS
	RIIC3MR3	SMBE	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1]	NF[0]
	RIIC3FER	-	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
	RIIC3SER	HOAE	-	DIDE	-	GCE	SAR2	SAR1	SAR0
	RIIC3IER	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
	RIIC3SR1	HOA	-	DID	-	GCA	AAS2	AAS1	AAS0
	RIIC3SR2	TDRE	TEND	RDRF	NACKF	STOP	START	3AL	TMOF
	RIIC3SAR0	FS0	-	-	-	-	-	SVA[9]	SVA[8]
		SVA[7]	SVA[6]	SVA[5]	SVA[4]	SVA[3]	SVA[2]	SVA[1]	SVA0

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
I²C bus interface	RIIC3SAR1	FS1	-	-	-	-	-	SVA[9]	SVA[8]
		SVA[7]	SVA[6]	SVA[5]	SVA[4]	SVA[3]	SVA[2]	SVA[1]	SVA0
	RIIC3SAR2	FS2	-	-	-	-	-	SVA[9]	SVA[8]
		SVA[7]	SVA[6]	SVA[5]	SVA[4]	SVA[3]	SVA[2]	SVA[1]	SVA0
	RIIC3BRL	-	-	-	BRL[4]	BRL[3]	BRL[2]	BRL[1]	BRL[0]
	RIIC3BRH	-	-	-	BRH[4]	BRH[3]	BRH[2]	BRH[1]	BRH[0]
	RIIC3DRT	DRT[7]	DRT[6]	DRT[5]	DRT[4]	DRT[3]	DRT[2]	DRT[1]	DRT[0]
	RIIC3DRR	DRR[7]	DRR[6]	DRR[5]	DRR[4]	DRR[3]	DRR[2]	DRR[1]	DRR[0]
Serial sound interface	SSICR_0	-	CKS	TUIEN	TOIEN	RUIEN	ROIEN	IIEN	-
		CHNL[1]	CHNL[0]	DWL[2]	DWL[1]	DWL[0]	SWL[2]	SWL[1]	SWL[0]
		SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL
		CKDV[3]	CKDV[2]	CKDV[1]	CKDV[0]	MUEN	-	TEN	REN
	SSISR_0	-	-	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	TCHNO[1]	TCHNO[0]	TSWNO	RCHNO[1]	RCHNO[0]	RSWNO	IDST
	SSIFCR_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		TTRG[1]	TTRG[0]	RTRG[1]	RTRG[0]	TIE	RIE	TFRST	RFRST
	SSIFSR_0	-	-	-	-	TDC[3]	TDC[2]	TDC[1]	TDC[0]
		-	-	-	-	-	-	-	TDE
		-	-	-	-	RDC[3]	RDC[2]	RDC[1]	RDC[0]
		-	-	-	-	-	-	-	RDF
	SSIFTDR_0								
	SSIFRDR_0								
	SSITDMR_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	RXDMUTE	-
		-	-	-	-	-	-	-	CONT
		-	-	-	-	-	-	-	TDM
	SSIFCCR_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	FIEN
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	FCEN
	SSIFCMR_0	-	-	MAXV[13]	MAXV[12]	MAXV[11]	MAXV[10]	MAXV[9]	MAXV[8]
		MAXV[7]	MAXV[6]	MAXV[5]	MAXV[4]	MAXV[3]	MAXV[2]	MAXV[1]	MAXV[0]
		-	-	MINV[13]	MINV[12]	MINV[11]	MINV[10]	MINV[9]	MINV[8]
		MINV[7]	MINV[6]	MINV[5]	MINV[4]	MINV[3]	MINV[2]	MINV[1]	MINV[0]
	SSIFCSR_0	-	-	-	-	-	-	-	FCIRQ
		-	-	-	-	-	-	-	-
		-	-	VALUE[13]	VALUE[12]	VALUE[11]	VALUE[10]	VALUE[9]	VALUE[8]
		VALUE[7]	VALUE[6]	VALUE[5]	VALUE[4]	VALUE[3]	VALUE[2]	VALUE[1]	VALUE[0]
	SSICR_1	-	CKS	TUIEN	TOIEN	RUIEN	ROIEN	IIEN	-
		CHNL[1]	CHNL[0]	DWL[2]	DWL[1]	DWL[0]	SWL[2]	SWL[1]	SWL[0]
		SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL
		CKDV[3]	CKDV[2]	CKDV[1]	CKDV[0]	MUEN	-	TEN	REN
	SSISR_1	-	-	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	TCHNO[1]	TCHNO[0]	TSWNO	RCHNO[1]	RCHNO[0]	RSWNO	IDST
	SSIFCR_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		TTRG[1]	TTRG[0]	RTRG[1]	RTRG[0]	TIE	RIE	TFRST	RFRST

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Serial sound interface	SSIFSR_1	-	-	-	-	TDC[3]	TDC[2]	TDC[1]	TDC[0]
		-	-	-	-	-	-	-	TDE
		-	-	-	-	RDC[3]	RDC[2]	RDC[1]	RDC[0]
		-	-	-	-	-	-	-	RDF
	SSIFTDR_1								
	SSIFRDR_1								
	SSITDMR_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	RXDMUTE	-
		-	-	-	-	-	-	-	CONT
		-	-	-	-	-	-	-	TDM
	SSIFCCR_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	FIEN
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	FCEN
	SSIFCMR_1	-	-	MAXV[13]	MAXV[12]	MAXV[11]	MAXV[10]	MAXV[9]	MAXV[8]
		MAXV[7]	MAXV[6]	MAXV[5]	MAXV[4]	MAXV[3]	MAXV[2]	MAXV[1]	MAXV[0]
		-	-	MINV[13]	MINV[12]	MINV[11]	MINV[10]	MINV[9]	MINV[8]
		MINV[7]	MINV[6]	MINV[5]	MINV[4]	MINV[3]	MINV[2]	MINV[1]	MINV[0]
	SSIFCSR_1	-	-	-	-	-	-	-	FCIRQ
		-	-	-	-	-	-	-	-
		-	-	VALUE[13]	VALUE[12]	VALUE[11]	VALUE[10]	VALUE[9]	VALUE[8]
		VALUE[7]	VALUE[6]	VALUE[5]	VALUE[4]	VALUE[3]	VALUE[2]	VALUE[1]	VALUE[0]
	SSICR_2	-	CKS	TUIEN	TOIEN	RUIEN	ROIEN	IIEN	-
		CHNL[1]	CHNL[0]	DWL[2]	DWL[1]	DWL[0]	SWL[2]	SWL[1]	SWL[0]
		SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL
		CKDV[3]	CKDV[2]	CKDV[1]	CKDV[0]	MUEN	-	TEN	REN
	SSISR_2	-	-	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	TCHNO[1]	TCHNO[0]	TSWNO	RCHNO[1]	RCHNO[0]	RSWNO	IDST
	SSIFCR_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		TTRG[1]	TTRG[0]	RTRG[1]	RTRG[0]	TIE	RIE	TFRST	RFRST
	SSIFSR_2	-	-	-	-	TDC[3]	TDC[2]	TDC[1]	TDC[0]
		-	-	-	-	-	-	-	TDE
		-	-	-	-	RDC[3]	RDC[2]	RDC[1]	RDC[0]
		-	-	-	-	-	-	-	RDF
	SSIFTDR_2								
	SSIFRDR_2								
	SSITDMR_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	RXDMUTE	-
		-	-	-	-	-	-	-	CONT
		-	-	-	-	-	-	-	TDM
	SSIFCCR_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	FIEN
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	FCEN

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Serial sound interface	SSIFCMR_2	-	-	MAXV[13]	MAXV[12]	MAXV[11]	MAXV[10]	MAXV[9]	MAXV[8]
		MAXV[7]	MAXV[6]	MAXV[5]	MAXV[4]	MAXV[3]	MAXV[2]	MAXV[1]	MAXV[0]
		-	-	MINV[13]	MINV[12]	MINV[11]	MINV[10]	MINV[9]	MINV[8]
		MINV[7]	MINV[6]	MINV[5]	MINV[4]	MINV[3]	MINV[2]	MINV[1]	MINV[0]
	SSIFCSR_2	-	-	-	-	-	-	-	FCIRQ
		-	-	-	-	-	-	-	-
		-	-	VALUE[13]	VALUE[12]	VALUE[11]	VALUE[10]	VALUE[9]	VALUE[8]
		VALUE[7]	VALUE[6]	VALUE[5]	VALUE[4]	VALUE[3]	VALUE[2]	VALUE[1]	VALUE[0]
	SSICR_3	-	CKS	TUIEN	TOIEN	RUIEN	ROIEN	IIEN	-
		CHNL[1]	CHNL[0]	DWL[2]	DWL[1]	DWL[0]	SWL[2]	SWL[1]	SWL[0]
		SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL
		CKDV[3]	CKDV[2]	CKDV[1]	CKDV[0]	MUEN	-	TEN	REN
	SSISR_3	-	-	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	TCHNO[1]	TCHNO[0]	TSWNO	RCHNO[1]	RCHNO[0]	RSWNO	IDST
	SSIFCR_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		TTRG[1]	TTRG[0]	RTRG[1]	RTRG[0]	TIE	RIE	TFRST	RFRST
	SSIFSR_3	-	-	-	-	TDC[3]	TDC[2]	TDC[1]	TDC[0]
		-	-	-	-	-	-	-	TDE
		-	-	-	-	RDC[3]	RDC[2]	RDC[1]	RDC[0]
		-	-	-	-	-	-	-	RDF
	SSIFTDR_3								
	SSIFRDR_3								
	SSITDMR_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	RXDMUTE	-
		-	-	-	-	-	-	-	CONT
		-	-	-	-	-	-	-	TDM
	SSIFCCR_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	FIEN
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	FCEN
	SSIFCMR_3	-	-	MAXV[13]	MAXV[12]	MAXV[11]	MAXV[10]	MAXV[9]	MAXV[8]
		MAXV[7]	MAXV[6]	MAXV[5]	MAXV[4]	MAXV[3]	MAXV[2]	MAXV[1]	MAXV[0]
		-	-	MINV[13]	MINV[12]	MINV[11]	MINV[10]	MINV[9]	MINV[8]
		MINV[7]	MINV[6]	MINV[5]	MINV[4]	MINV[3]	MINV[2]	MINV[1]	MINV[0]
	SSIFCSR_3	-	-	-	-	-	-	-	FCIRQ
		-	-	-	-	-	-	-	-
		-	-	VALUE[13]	VALUE[12]	VALUE[11]	VALUE[10]	VALUE[9]	VALUE[8]
		VALUE[7]	VALUE[6]	VALUE[5]	VALUE[4]	VALUE[3]	VALUE[2]	VALUE[1]	VALUE[0]
Media local bus*1	DCCR	MDE	LBM	MCS[1]	MCS[0]	M5PS	MLK	MLE	MHRE
		MRS	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		MDA[8]	MDA[7]	MDA[6]	MDA[5]	MDA[4]	MDA[3]	MDA[2]	MDA[1]
	SSCR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		SSRE	SDMU	SDML	SDSC	SDCS	SDNU	SDNL	SDR
	SDCR	MSD[31]	MSD[30]	MSD[29]	MSD[28]	MSD[27]	MSD[26]	MSD[25]	MSD[24]
		MSD[23]	MSD[22]	MSD[21]	MSD[20]	MSD[19]	MSD[18]	MSD[17]	MSD[16]
		MSD[15]	MSD[14]	MSD[13]	MSD[12]	MSD[11]	MSD[10]	MSD[9]	MSD[8]
		MSD[7]	MSD[6]	MSD[5]	MSD[4]	MSD[3]	MSD[2]	MSD[1]	MSD[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Media local bus*1	SMCR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	SMMU	SMML	SMSC	SMCS	SMNU	SMNL	SMR
	VCCR	UMA[7]	UMA[6]	UMA[5]	UMA[4]	UMA[3]	UMA[2]	UMA[1]	UMA[0]
		UMI[7]	UMI[6]	UMI[5]	UMI[4]	UMI[3]	UMI[2]	UMI[1]	UMI[0]
		MMA[7]	MMA[6]	MMA[5]	MMA[4]	MMA[3]	MMA[2]	MMA[1]	MMA[0]
		MMI[7]	MMI[6]	MMI[5]	MMI[4]	MMI[3]	MMI[2]	MMI[1]	MMI[0]
	SBCR	SRBA[31]	SRBA[30]	SRBA[29]	SRBA[28]	SRBA[27]	SRBA[26]	SRBA[25]	SRBA[24]
		SRBA[23]	SRBA[22]	SRBA[21]	SRBA[20]	SRBA[19]	SRBA[18]	SRBA[17]	SRBA[16]
		STBA[31]	STBA[30]	STBA[29]	STBA[28]	STBA[27]	STBA[26]	STBA[25]	STBA[24]
		STBA[23]	STBA[22]	STBA[21]	STBA[20]	STBA[19]	STBA[18]	STBA[17]	STBA[16]
	ABCR	ARBA[31]	ARBA[30]	ARBA[29]	ARBA[28]	ARBA[27]	ARBA[26]	ARBA[25]	ARBA[24]
		ARBA[23]	ARBA[22]	ARBA[21]	ARBA[20]	ARBA[19]	ARBA[18]	ARBA[17]	ARBA[16]
		ATBA[31]	ATBA[30]	ATBA[29]	ATBA[28]	ATBA[27]	ATBA[26]	ATBA[25]	ATBA[24]
		ATBA[23]	ATBA[22]	ATBA[21]	ATBA[20]	ATBA[19]	ATBA[18]	ATBA[17]	ATBA[16]
	CBCR	CRBA[31]	CRBA[30]	CRBA[29]	CRBA[28]	CRBA[27]	CRBA[26]	CRBA[25]	CRBA[24]
		CRBA[23]	CRBA[22]	CRBA[21]	CRBA[20]	CRBA[19]	CRBA[18]	CRBA[17]	CRBA[16]
		CTBA[31]	CTBA[30]	CTBA[29]	CTBA[28]	CTBA[27]	CTBA[26]	CTBA[25]	CTBA[24]
		CTBA[23]	CTBA[22]	CTBA[21]	CTBA[20]	CTBA[19]	CTBA[18]	CTBA[17]	CTBA[16]
	IBCR	IRBA[31]	IRBA[30]	IRBA[29]	IRBA[28]	IRBA[27]	IRBA[26]	IRBA[25]	IRBA[24]
		IRBA[23]	IRBA[22]	IRBA[21]	IRBA[20]	IRBA[19]	IRBA[18]	IRBA[17]	IRBA[16]
		ITBA[31]	ITBA[30]	ITBA[29]	ITBA[28]	ITBA[27]	ITBA[26]	ITBA[25]	ITBA[24]
		ITBA[23]	ITBA[22]	ITBA[21]	ITBA[20]	ITBA[19]	ITBA[18]	ITBA[17]	ITBA[16]
	CICR	-	CnSU[30]	CnSU[29]	CnSU[28]	CnSU[27]	CnSU[26]	CnSU[25]	CnSU[24]
		CnSU[23]	CnSU[22]	CnSU[21]	CnSU[20]	CnSU[19]	CnSU[18]	CnSU[17]	CnSU[16]
		CnSU[15]	CnSU[14]	CnSU[13]	CnSU[12]	CnSU[11]	CnSU[10]	CnSU[9]	CnSU[8]
		CnSU[7]	CnSU[6]	CnSU[5]	CnSU[4]	CnSU[3]	CnSU[2]	CnSU[1]	CnSU[0]
	CECR0	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR0	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR0	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR0	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR1	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR1	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR1	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR1	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Media local bus*1	CECR2	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR2	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR2	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR2	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR3	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR3	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR3	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR3	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR4	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR4	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR4	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR4	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR5	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR5	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR5	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Media local bus*1	CNBCR5	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR6	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR6	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR6	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR6	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR7	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR7	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR7	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR7	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR8	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR8	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR8	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR8	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR9	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR9	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Media local bus*1	CCBCR9	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR9	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR10	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR10	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR10	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR10	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR11	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR11	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR11	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR11	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR12	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR12	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR12	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR12	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR13	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Media local bus*1	CSCR13	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR13	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR13	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR14	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR14	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR14	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR14	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR15	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR15	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR15	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR15	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR16	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR16	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR16	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR16	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Media local bus*1	CECR17	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR17	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR17	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR17	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR18	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR18	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR18	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR18	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR19	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR19	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR19	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR19	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR20	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR20	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR20	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Media local bus*1	CNBCR20	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR21	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR21	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR21	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR21	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR22	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR22	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR22	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR22	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR23	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR23	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR23	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR23	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR24	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR24	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Media local bus*1	CCBCR24	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR24	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR25	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR25	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR25	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR25	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR26	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR26	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR26	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR26	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR27	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR27	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR27	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR27	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Media local bus*1	CECR28	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR28	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR28	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR28	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR29	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR29	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR29	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR29	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	CECR30	CE	TR	CT[1]	CT[0]	PCE_FSE	MDS[1]	MDS[0]	-
		MASK[7]	MASK[6]	MASK[5]	MASK[4]	MASK[3]	MASK[2]	MASK[1]	MASK[0]
		FSCD	-	-	PCTH_FSPC[4]	PCTH_FSPC[3]	PCTH_FSPC[2]	PCTH_FSPC[1]	PCTH_FSPC[0]
		CA[8]	CA[7]	CA[6]	CA[5]	CA[4]	CA[3]	CA[2]	CA[1]
	CSCR30	BM	BF	-	-	-	-	-	-
		-	-	-	-	-	-	GB	RDY
		STS[15]	STS[14]	STS[13]	STS[12]	STS[11]	STS[10]	STS[9]	STS[8]
		STS[7]	STS[6]	STS[5]	STS[4]	STS[3]	STS[2]	STS[1]	STS[0]
	CCBCR30	BCA[15]	BCA[14]	BCA[13]	BCA[12]	BCA[11]	BCA[10]	BCA[9]	BCA[8]
		BCA[7]	BCA[6]	BCA[5]	BCA[4]	BCA[3]	BCA[2]	BCA[1]	BCA[0]
		BFA[15]	BFA[14]	BFA[13]	BFA[12]	BFA[11]	BFA[10]	BFA[9]	BFA[8]
		BFA[7]	BFA[6]	BFA[5]	BFA[4]	BFA[3]	BFA[2]	BFA[1]	BFA[0]
	CNBCR30	BSA[15]	BSA[14]	BSA[13]	BSA[12]	BSA[11]	BSA[10]	BSA[9]	BSA[8]
		BSA[7]	BSA[6]	BSA[5]	BSA[4]	BSA[3]	BSA[2]	-	-
		BEA[15]	BEA[14]	BEA[13]	BEA[12]	BEA[11]	BEA[10]	BEA[9]	BEA[8]
		BEA[7]	BEA[6]	BEA[5]	BEA[4]	BEA[3]	BEA[2]	BEA[1]	BEA[0]
	LCBCR0	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR1	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Media local bus*1	LCBCR2	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR3	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR4	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR5	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR6	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR7	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR8	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR9	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR10	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR11	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR12	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR13	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR14	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR15	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Media local bus*1	LCBCR16	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR17	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR18	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR19	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR20	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR21	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR22	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR23	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR24	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR25	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR26	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR27	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR28	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR29	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]
	LCBCR30	TH[9]	TH[8]	TH[7]	TH[6]	TH[5]	TH[4]	TH[3]	TH[2]
		TH[1]	TH[0]	BD[8]	BD[7]	BD[6]	BD[5]	BD[4]	BD[3]
		BD[2]	BD[1]	BD[0]	SA[12]	SA[11]	SA[10]	SA[9]	SA[8]
		SA[7]	SA[6]	SA[5]	SA[4]	SA[3]	SA[2]	SA[1]	SA[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
CAN interface	RSCAN0CmCFG (m = 0, 1)	-	-	-	-	-	-	SJW[1]	SJW[0]
		-	TSEG2[2]	TSEG2[1]	TSEG2[0]	TSEG1[3]	TSEG1[2]	TSEG1[1]	TSEG1[0]
		-	-	-	-	-	-	BRP[9]	BRP[8]
		BRP[7]	BRP[6]	BRP[5]	BRP[4]	BRP[3]	BRP[2]	BRP[1]	BRP[0]
	RSCAN0CmCTR (m = 0, 1)	-	-	-	-	-	CTMS[1]	CTMS[0]	CTME
		ERRD	BOM[1]	BOM[0]	-	-	-	-	TAIE
		ALIE	BLIE	OLIE	BORIE	BOEIE	EPIE	EWIE	BEIE
		-	-	-	-	RTBO	CSLPR	CHMDC[1]	CHMDC[0]
	RSCAN0CmSTS (m = 0, 1)	TEC[7]	TEC[6]	TEC[5]	TEC[4]	TEC[3]	TEC[2]	TEC[1]	TEC[0]
		REC[7]	REC[6]	REC[5]	REC[4]	REC[3]	REC[2]	REC[1]	REC[0]
		-	-	-	-	-	-	-	-
		COMSTS	RECSTS	TRMSTS	BOSTS	EPSTS	CSLPSTS	CHLTSTS	CRSTSTS
	RSCAN0CmERFL (m = 0, 1)	-	CRCREG[14]	CRCREG[13]	CRCREG[12]	CRCREG[11]	CRCREG[10]	CRCREG[9]	CRCREG[8]
		CRCREG[7]	CRCREG[6]	CRCREG[5]	CRCREG[4]	CRCREG[3]	CRCREG[2]	CRCREG[1]	CRCREG[0]
		-	ADERR	B0ERR	B1ERR	CERR	AERR	FERR	SERR
		ALF	BLF	OVLF	BORF	BOEF	EPF	EWF	BEF
	RSCAN0GCFG	ITRCP[15]	ITRCP[14]	ITRCP[13]	ITRCP[12]	ITRCP[11]	ITRCP[10]	ITRCP[9]	ITRCP[8]
		ITRCP[7]	ITRCP[6]	ITRCP[5]	ITRCP[4]	ITRCP[3]	ITRCP[2]	ITRCP[1]	ITRCP[0]
		TSBTCS[2]	TSBTCS[1]	TSBTCS[0]	TSSS	TSP[3]	TSP[2]	TSP[1]	TSP[0]
		-	-	-	DCS	MME	DRE	DCE	TPRI
	RSCAN0GCTR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	TSRST
		-	-	-	-	-	THLEIE	MEIE	DEIE
		-	-	-	-	-	GSLPR	GMDC[1]	GMDC[0]
	RSCAN0GSTS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	GRAMINIT	GSLPSTS	GHLTSTS	GRSTSTS
	RSCAN0GERFL	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	THLES	MES	DEF
	RSCAN0GTSC	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		TS[15]	TS[14]	TS[13]	TS[12]	TS[11]	TS[10]	TS[9]	TS[8]
		TS[7]	TS[6]	TS[5]	TS[4]	TS[3]	TS[2]	TS[1]	TS[0]
	RSCAN0GAFLECTR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	AFLDAE
		-	-	-	AFLPN[4]	AFLPN[3]	AFLPN[2]	AFLPN[1]	AFLPN[0]
	RSCAN0GAFLCFG0	RNC0[7]	RNC0[6]	RNC0[5]	RNC0[4]	RNC0[3]	RNC0[2]	RNC0[1]	RNC0[0]
		RNC1[7]	RNC1[6]	RNC1[5]	RNC1[4]	RNC1[3]	RNC1[2]	RNC1[1]	RNC1[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	RSCAN0RMNB	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		NRXMB[7]	NRXMB[6]	NRXMB[5]	NRXMB[4]	NRXMB[3]	NRXMB[2]	NRXMB[1]	NRXMB[0]
	RSCAN0RMND0	RMNS[31]	RMNS[30]	RMNS[29]	RMNS[28]	RMNS[27]	RMNS[26]	RMNS[25]	RMNS[24]
		RMNS[23]	RMNS[22]	RMNS[21]	RMNS[20]	RMNS[19]	RMNS[18]	RMNS[17]	RMNS[16]
		RMNS[15]	RMNS[14]	RMNS[13]	RMNS[12]	RMNS[11]	RMNS[10]	RMNS[9]	RMNS[8]
		RMNS[7]	RMNS[6]	RMNS[5]	RMNS[4]	RMNS[3]	RMNS[2]	RMNS[1]	RMNS[0]
	RSCAN0RFCCx (x = 0 to 7)	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		RFIGCV[2]	RFIGCV[1]	RFIGCV[0]	RFIM	-	RFDC[2]	RFDC[1]	RFDC[0]
		-	-	-	-	-	-	RFIE	RFE
	RSCAN0RFSTSx (x = 0 to 7)	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		RFMC[7]	RFMC[6]	RFMC[5]	RFMC[4]	RFMC[3]	RFMC[2]	RFMC[1]	RFMC[0]
		-	-	-	-	RFIF	RFMLT	RFFLL	RFEMP

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
CAN interface	RSCAN0RFPCTRx (x = 0 to 7)	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		RFPC[7]	RFPC[6]	RFPC[5]	RFPC[4]	RFPC[3]	RFPC[2]	RFPC[1]	RFPC[0]
	RSCAN0CFCCk (k = 0 to 5)	CFITT[7]	CFITT[6]	CFITT[5]	CFITT[4]	CFITT[3]	CFITT[2]	CFITT[1]	CFITT[0]
		CFTML[3]	CFTML[2]	CFTML[1]	CFTML[0]	CFITR	CFITSS	CFM[1]	CFM[0]
		CFIGCV[2]	CFIGCV[1]	CFIGCV[0]	CFIM	-	CFDC[2]	CFDC[1]	CFDC[0]
		-	-	-	-	-	CFTXIE	CFRXIE	CFE
	RSCAN0CFSTSk (k = 0 to 5)	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		CFMC[7]	CFMC[6]	CFMC[5]	CFMC[4]	CFMC[3]	CFMC[2]	CFMC[1]	CFMC[0]
		-	-	-	CFTXIF	CFRXIF	CFMLT	CFFLL	CFEMP
	RSCAN0CFPCTRk (k = 0 to 5)	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		CFPC[7]	CFPC[6]	CFPC[5]	CFPC[4]	CFPC[3]	CFPC[2]	CFPC[1]	CFPC[0]
	RSCAN0FESTS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	CF5EMP	CF4EMP	CF3EMP	CF2EMP	CF1EMP	CF0EMP
		RF7EMP	RF6EMP	RF5EMP	RF4EMP	RF3EMP	RF2EMP	RF1EMP	RF0EMP
	RSCAN0FFSTS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	CF5FLL	CF4FLL	CF3FLL	CF2FLL	CF1FLL	CF0FLL
		RF7FLL	RF6FLL	RF5FLL	RF4FLL	RF3FLL	RF2FLL	RF1FLL	RF0FLL
	RSCAN0FMSTS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	CF5MLT	CF4MLT	CF3MLT	CF2MLT	CF1MLT	CF0MLT
		RF7MLT	RF6MLT	RF5MLT	RF4MLT	RF3MLT	RF2MLT	RF1MLT	RF0MLT
	RSCAN0RFISTS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		RF7IF	RF6IF	RF5IF	RF4IF	RF3IF	RF2IF	RF1IF	RF0IF
	RSCAN0CFRISTS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	CF5RXIF	CF4RXIF	CF3RXIF	CF2RXIF	CF1RXIF	CF0RXIF
	RSCAN0CFTISIS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	CF5TXIF	CF4TXIF	CF3TXIF	CF2TXIF	CF1TXIF	CF0TXIF
	RSCAN0TMCp (p = 0 to 31)								
		-	-	-	-	-	TMOM	TMTAR	TMTR
	RSCAN0TMSTSp (p = 0 to 31)								
		-	-	-	TMTARM	TMTRM	TMTRF[1]	TMTRF[0]	TMTSTS
	RSCAN0TMTRSTS0	TMTRSTS[31]	TMTRSTS[30]	TMTRSTS[29]	TMTRSTS[28]	TMTRSTS[27]	TMTRSTS[26]	TMTRSTS[25]	TMTRSTS[24]
		TMTRSTS[23]	TMTRSTS[22]	TMTRSTS[21]	TMTRSTS[20]	TMTRSTS[19]	TMTRSTS[18]	TMTRSTS[17]	TMTRSTS[16]
		TMTRSTS[15]	TMTRSTS[14]	TMTRSTS[13]	TMTRSTS[12]	TMTRSTS[11]	TMTRSTS[10]	TMTRSTS[9]	TMTRSTS[8]
		TMTRSTS[7]	TMTRSTS[6]	TMTRSTS[5]	TMTRSTS[4]	TMTRSTS[3]	TMTRSTS[2]	TMTRSTS[1]	TMTRSTS[0]
	RSCAN0TMTARSTS0	TMTARSTS[31]	TMTARSTS[30]	TMTARSTS[29]	TMTARSTS[28]	TMTARSTS[27]	TMTARSTS[26]	TMTARSTS[25]	TMTARSTS[24]
		TMTARSTS[23]	TMTARSTS[22]	TMTARSTS[21]	TMTARSTS[20]	TMTARSTS[19]	TMTARSTS[18]	TMTARSTS[17]	TMTARSTS[16]
		TMTARSTS[15]	TMTARSTS[14]	TMTARSTS[13]	TMTARSTS[12]	TMTARSTS[11]	TMTARSTS[10]	TMTARSTS[9]	TMTARSTS[8]
		TMTARSTS[7]	TMTARSTS[6]	TMTARSTS[5]	TMTARSTS[4]	TMTARSTS[3]	TMTARSTS[2]	TMTARSTS[1]	TMTARSTS[0]
	RSCAN0TMTCSTS0	TMTCSTS[31]	TMTCSTS[30]	TMTCSTS[29]	TMTCSTS[28]	TMTCSTS[27]	TMTCSTS[26]	TMTCSTS[25]	TMTCSTS[24]
		TMTCSTS[23]	TMTCSTS[22]	TMTCSTS[21]	TMTCSTS[20]	TMTCSTS[19]	TMTCSTS[18]	TMTCSTS[17]	TMTCSTS[16]
		TMTCSTS[15]	TMTCSTS[14]	TMTCSTS[13]	TMTCSTS[12]	TMTCSTS[11]	TMTCSTS[10]	TMTCSTS[9]	TMTCSTS[8]
		TMTCSTS[7]	TMTCSTS[6]	TMTCSTS[5]	TMTCSTS[4]	TMTCSTS[3]	TMTCSTS[2]	TMTCSTS[1]	TMTCSTS[0]
	RSCAN0TMTASTS0	TMTASTS[31]	TMTASTS[30]	TMTASTS[29]	TMTASTS[28]	TMTASTS[27]	TMTASTS[26]	TMTASTS[25]	TMTASTS[24]
		TMTASTS[23]	TMTASTS[22]	TMTASTS[21]	TMTASTS[20]	TMTASTS[19]	TMTASTS[18]	TMTASTS[17]	TMTASTS[16]
		TMTASTS[15]	TMTASTS[14]	TMTASTS[13]	TMTASTS[12]	TMTASTS[11]	TMTASTS[10]	TMTASTS[9]	TMTASTS[8]
		TMTASTS[7]	TMTASTS[6]	TMTASTS[5]	TMTASTS[4]	TMTASTS[3]	TMTASTS[2]	TMTASTS[1]	TMTASTS[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
CAN interface	RSCAN0TMIEC0	TMIE[31]	TMIE[30]	TMIE[29]	TMIE[28]	TMIE[27]	TMIE[26]	TMIE[25]	TMIE[24]
		TMIE[23]	TMIE[22]	TMIE[21]	TMIE[20]	TMIE[19]	TMIE[18]	TMIE[17]	TMIE[16]
		TMIE[15]	TMIE[14]	TMIE[13]	TMIE[12]	TMIE[11]	TMIE[10]	TMIE[9]	TMIE[8]
		TMIE[7]	TMIE[6]	TMIE[5]	TMIE[4]	TMIE[3]	TMIE[2]	TMIE[1]	TMIE[0]
	RSCAN0TXQCCm (m = 0, 1)	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	TXQIM	TXQIE	TXQDC[3]	TXQDC[2]	TXQDC[1]	TXQDC[0]
		-	-	-	-	-	-	-	TXQE
	RSCAN0TXQSTSm (m = 0, 1)	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	TXQIF	TXQFLL	TXQEMP
	RSCAN0TXQPCTRm (m = 0, 1)	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		TXQPC[7]	TXQPC[6]	TXQPC[5]	TXQPC[4]	TXQPC[3]	TXQPC[2]	TXQPC[1]	TXQPC[0]
	RSCAN0THLCCm (m = 0, 1)	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	THLDTE	THLIM	THLIE
		-	-	-	-	-	-	-	THLE
	RSCAN0THLSTSm (m = 0, 1)	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	THLMC[4]	THLMC[3]	THLMC[2]	THLMC[1]	THLMC[0]
		-	-	-	-	THLIF	THLELT	THLFLL	THLEMP
	RSCAN0THLPCTRm (m = 0, 1)	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		THLPC[7]	THLPC[6]	THLPC[5]	THLPC[4]	THLPC[3]	THLPC[2]	THLPC[1]	THLPC[0]
	RSCAN0GTINTSTS0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	THIF1	CFTIF1	TQIF1	TAIF1	TSIF1
		-	-	-	THIF0	CFTIF0	TQIF0	TAIF0	TSIF0
	RSCAN0GTSTCFG	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	C1ICBCE	C0ICBCE
	RSCAN0GTSTCTR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	ICBCTME
	RSCAN0GLOCKK	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		LOCK[15]	LOCK[14]	LOCK[13]	LOCK[12]	LOCK[11]	LOCK[10]	LOCK[9]	LOCK[8]
		LOCK[7]	LOCK[6]	LOCK[5]	LOCK[4]	LOCK[3]	LOCK[2]	LOCK[1]	LOCK[0]
	RSCAN0GAFLIDj (j = 0 to 15)	GAFLIDE	GAFLRTR	GAFLLB	GAFLID[28]	GAFLID[27]	GAFLID[26]	GAFLID[25]	GAFLID[24]
		GAFLID[23]	GAFLID[22]	GAFLID[21]	GAFLID[20]	GAFLID[19]	GAFLID[18]	GAFLID[17]	GAFLID[16]
		GAFLID[15]	GAFLID[14]	GAFLID[13]	GAFLID[12]	GAFLID[11]	GAFLID[10]	GAFLID[9]	GAFLID[8]
		GAFLID[7]	GAFLID[6]	GAFLID[5]	GAFLID[4]	GAFLID[3]	GAFLID[2]	GAFLID[1]	GAFLID[0]
	RSCAN0GAFLMj (j = 0 to 15)	GAFLIDEM	GAFLRTRM	-	GAFLIDM[28]	GAFLIDM[27]	GAFLIDM[26]	GAFLIDM[25]	GAFLIDM[24]
		GAFLIDM[23]	GAFLIDM[22]	GAFLIDM[21]	GAFLIDM[20]	GAFLIDM[19]	GAFLIDM[18]	GAFLIDM[17]	GAFLIDM[16]
		GAFLIDM[15]	GAFLIDM[14]	GAFLIDM[13]	GAFLIDM[12]	GAFLIDM[11]	GAFLIDM[10]	GAFLIDM[9]	GAFLIDM[8]
		GAFLIDM[7]	GAFLIDM[6]	GAFLIDM[5]	GAFLIDM[4]	GAFLIDM[3]	GAFLIDM[2]	GAFLIDM[1]	GAFLIDM[0]
	RSCAN0GAFLP0j (j = 0 to 15)	GAFLDLC[3]	GAFLDLC[2]	GAFLDLC[1]	GAFLDLC[0]	GAFLPTR[11]	GAFLPTR[10]	GAFLPTR[9]	GAFLPTR[8]
		GAFLPTR[7]	GAFLPTR[6]	GAFLPTR[5]	GAFLPTR[4]	GAFLPTR[3]	GAFLPTR[2]	GAFLPTR[1]	GAFLPTR[0]
		GAFLRMV	GAFLRMDP[6]	GAFLRMDP[5]	GAFLRMDP[4]	GAFLRMDP[3]	GAFLRMDP[2]	GAFLRMDP[1]	GAFLRMDP[0]
		-	-	-	-	-	-	-	-
	RSCAN0GAFLP1j (j = 0 to 15)	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	GAFLFDP[13]	GAFLFDP[12]	GAFLFDP[11]	GAFLFDP[10]	GAFLFDP[9]	GAFLFDP[8]
		GAFLFDP[7]	GAFLFDP[6]	GAFLFDP[5]	GAFLFDP[4]	GAFLFDP[3]	GAFLFDP[2]	GAFLFDP[1]	GAFLFDP[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
CAN interface	RSCAN0RMIDq (q = 0 to 31)	PMIDE	PMRTR	-	RMID[28]	RMID[27]	RMID[26]	RMID[25]	RMID[24]
		RMID[23]	RMID[22]	RMID[21]	RMID[20]	RMID[19]	RMID[18]	RMID[17]	RMID[16]
		RMID[15]	RMID[14]	RMID[13]	RMID[12]	RMID[11]	RMID[10]	RMID[9]	RMID[8]
		RMID[7]	RMID[6]	RMID[5]	RMID[4]	RMID[3]	RMID[2]	RMID[1]	RMID[0]
	RSCAN0RMPTRq (q = 0 to 31)	RMDLC[3]	RMDLC[2]	RMDLC[1]	RMDLC[0]	RMPTR[11]	RMPTR[10]	RMPTR[9]	RMPTR[8]
		RMPTR[7]	RMPTR[6]	RMPTR[5]	RMPTR[4]	RMPTR[3]	RMPTR[2]	RMPTR[1]	RMPTR[0]
		RMTS[15]	RMTS[14]	RMTS[13]	RMTS[12]	RMTS[11]	RMTS[10]	RMTS[9]	RMTS[8]
		RMTS[7]	RMTS[6]	RMTS[5]	RMTS[4]	RMTS[3]	RMTS[2]	RMTS[1]	RMTS[0]
	RSCAN0RMDF0q (q = 0 to 31)	RMDB3[7]	RMDB3[6]	RMDB3[5]	RMDB3[4]	RMDB3[3]	RMDB3[2]	RMDB3[1]	RMDB3[0]
		RMDB2[7]	RMDB2[6]	RMDB2[5]	RMDB2[4]	RMDB2[3]	RMDB2[2]	RMDB2[1]	RMDB2[0]
		RMDB1[7]	RMDB1[6]	RMDB1[5]	RMDB1[4]	RMDB1[3]	RMDB1[2]	RMDB1[1]	RMDB1[0]
		RMDB0[7]	RMDB0[6]	RMDB0[5]	RMDB0[4]	RMDB0[3]	RMDB0[2]	RMDB0[1]	RMDB0[0]
	RSCAN0RMDF1q (q = 0 to 31)	RMDB7[7]	RMDB7[6]	RMDB7[5]	RMDB7[4]	RMDB7[3]	RMDB7[2]	RMDB7[1]	RMDB7[0]
		RMDB6[7]	RMDB6[6]	RMDB6[5]	RMDB6[4]	RMDB6[3]	RMDB6[2]	RMDB6[1]	RMDB6[0]
		RMDB5[7]	RMDB5[6]	RMDB5[5]	RMDB5[4]	RMDB5[3]	RMDB5[2]	RMDB5[1]	RMDB5[0]
		RMDB4[7]	RMDB4[6]	RMDB4[5]	RMDB4[4]	RMDB4[3]	RMDB4[2]	RMDB4[1]	RMDB4[0]
	RSCAN0RFIDx (x = 0 to 7)	RFIDE	RFRTR	-	RFID[28]	RFID[27]	RFID[26]	RFID[25]	RFID[24]
		RFID[23]	RFID[22]	RFID[21]	RFID[20]	RFID[19]	RFID[18]	RFID[17]	RFID[16]
		RFID[15]	RFID[14]	RFID[13]	RFID[12]	RFID[11]	RFID[10]	RFID[9]	RFID[8]
		RFID[7]	RFID[6]	RFID[5]	RFID[4]	RFID[3]	RFID[2]	RFID[1]	RFID[0]
	RSCAN0RFPTRx (x = 0 to 7)	RFDLC[3]	RFDLC[2]	RFDLC[1]	RFDLC[0]	RFPTR[11]	RFPTR[10]	RFPTR[9]	RFPTR[8]
		RFPTR[7]	RFPTR[6]	RFPTR[5]	RFPTR[4]	RFPTR[3]	RFPTR[2]	RFPTR[1]	RFPTR[0]
		RFTS[15]	RFTS[14]	RFTS[13]	RFTS[12]	RFTS[11]	RFTS[10]	RFTS[9]	RFTS[8]
		RFTS[7]	RFTS[6]	RFTS[5]	RFTS[4]	RFTS[3]	RFTS[2]	RFTS[1]	RFTS[0]
	RSCAN0RFDF0x (x = 0 to 7)	RFDB3[7]	RFDB3[6]	RFDB3[5]	RFDB3[4]	RFDB3[3]	RFDB3[2]	RFDB3[1]	RFDB3[0]
		RFDB2[7]	RFDB2[6]	RFDB2[5]	RFDB2[4]	RFDB2[3]	RFDB2[2]	RFDB2[1]	RFDB2[0]
		RFDB1[7]	RFDB1[6]	RFDB1[5]	RFDB1[4]	RFDB1[3]	RFDB1[2]	RFDB1[1]	RFDB1[0]
		RFDB0[7]	RFDB0[6]	RFDB0[5]	RFDB0[4]	RFDB0[3]	RFDB0[2]	RFDB0[1]	RFDB0[0]
	RSCAN0RFDF1x (x = 0 to 7)	RFDB7[7]	RFDB7[6]	RFDB7[5]	RFDB7[4]	RFDB7[3]	RFDB7[2]	RFDB7[1]	RFDB7[0]
		RFDB6[7]	RFDB6[6]	RFDB6[5]	RFDB6[4]	RFDB6[3]	RFDB6[2]	RFDB6[1]	RFDB6[0]
		RFDB5[7]	RFDB5[6]	RFDB5[5]	RFDB5[4]	RFDB5[3]	RFDB5[2]	RFDB5[1]	RFDB5[0]
		RFDB4[7]	RFDB4[6]	RFDB4[5]	RFDB4[4]	RFDB4[3]	RFDB4[2]	RFDB4[1]	RFDB4[0]
	RSCAN0CFIDk (k = 0 to 5)	CFIDE	CFRTR	THLEN	CFID[28]	CFID[27]	CFID[26]	CFID[25]	CFID[24]
		CFID[23]	CFID[22]	CFID[21]	CFID[20]	CFID[19]	CFID[18]	CFID[17]	CFID[16]
		CFID[15]	CFID[14]	CFID[13]	CFID[12]	CFID[11]	CFID[10]	CFID[9]	CFID[8]
		CFID[7]	CFID[6]	CFID[5]	CFID[4]	CFID[3]	CFID[2]	CFID[1]	CFID[0]
	RSCAN0CFPTRk (k = 0 to 5)	CFDLC[3]	CFDLC[2]	CFDLC[1]	CFDLC[0]	CFPTR[11]	CFPTR[10]	CFPTR[9]	CFPTR[8]
		CFPTR[7]	CFPTR[6]	CFPTR[5]	CFPTR[4]	CFPTR[3]	CFPTR[2]	CFPTR[1]	CFPTR[0]
		CFTS[15]	CFTS[14]	CFTS[13]	CFTS[12]	CFTS[11]	CFTS[10]	CFTS[9]	CFTS[8]
		CFTS[7]	CFTS[6]	CFTS[5]	CFTS[4]	CFTS[3]	CFTS[2]	CFTS[1]	CFTS[0]
	RSCAN0CFDF0k (k = 0 to 5)	CFDB3[7]	CFDB3[6]	CFDB3[5]	CFDB3[4]	CFDB3[3]	CFDB3[2]	CFDB3[1]	CFDB3[0]
		CFDB2[7]	CFDB2[6]	CFDB2[5]	CFDB2[4]	CFDB2[3]	CFDB2[2]	CFDB2[1]	CFDB2[0]
		CFDB1[7]	CFDB1[6]	CFDB1[5]	CFDB1[4]	CFDB1[3]	CFDB1[2]	CFDB1[1]	CFDB1[0]
		CFDB0[7]	CFDB0[6]	CFDB0[5]	CFDB0[4]	CFDB0[3]	CFDB0[2]	CFDB0[1]	CFDB0[0]
	RSCAN0CFDF1k (k = 0 to 5)	CFDB7[7]	CFDB7[6]	CFDB7[5]	CFDB7[4]	CFDB7[3]	CFDB7[2]	CFDB7[1]	CFDB7[0]
		CFDB6[7]	CFDB6[6]	CFDB6[5]	CFDB6[4]	CFDB6[3]	CFDB6[2]	CFDB6[1]	CFDB6[0]
		CFDB5[7]	CFDB5[6]	CFDB5[5]	CFDB5[4]	CFDB5[3]	CFDB5[2]	CFDB5[1]	CFDB5[0]
		CFDB4[7]	CFDB4[6]	CFDB4[5]	CFDB4[4]	CFDB4[3]	CFDB4[2]	CFDB4[1]	CFDB4[0]
	RSCAN0TMIDp (p = 0 to 31)	TMIDE	TMRTR	THLEN	TMID[28]	TMID[27]	TMID[26]	TMID[25]	TMID[24]
		TMID[23]	TMID[22]	TMID[21]	TMID[20]	TMID[19]	TMID[18]	TMID[17]	TMID[16]
		TMID[15]	TMID[14]	TMID[13]	TMID[12]	TMID[11]	TMID[10]	TMID[9]	TMID[8]
		TMID[7]	TMID[6]	TMID[5]	TMID[4]	TMID[3]	TMID[2]	TMID[1]	TMID[0]
	RSCAN0TMPTRp (p = 0 to 31)	TMDLC[3]	TMDLC[2]	TMDLC[1]	TMDLC[0]	-	-	-	-
		TMPTR[7]	TMPTR[6]	TMPTR[5]	TMPTR[4]	TMPTR[3]	TMPTR[2]	TMPTR[1]	TMPTR[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	RSCAN0TMDF0p (p = 0 to 31)	TMDB3[7]	TMDB3[6]	TMDB3[5]	TMDB3[4]	TMDB3[3]	TMDB3[2]	TMDB3[1]	TMDB3[0]
		TMDB2[7]	TMDB2[6]	TMDB2[5]	TMDB2[4]	TMDB2[3]	TMDB2[2]	TMDB2[1]	TMDB2[0]
		TMDB1[7]	TMDB1[6]	TMDB1[5]	TMDB1[4]	TMDB1[3]	TMDB1[2]	TMDB1[1]	TMDB1[0]
		TMDB0[7]	TMDB0[6]	TMDB0[5]	TMDB0[4]	TMDB0[3]	TMDB0[2]	TMDB0[1]	TMDB0[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
CAN interface	RSCAN0TMDF1p (p = 0 to 31)	TMDB7[7]	TMDB7[6]	TMDB7[5]	TMDB7[4]	TMDB7[3]	TMDB7[2]	TMDB7[1]	TMDB7[0]
		TMDB6[7]	TMDB6[6]	TMDB6[5]	TMDB6[4]	TMDB6[3]	TMDB6[2]	TMDB6[1]	TMDB6[0]
		TMDB5[7]	TMDB5[6]	TMDB5[5]	TMDB5[4]	TMDB5[3]	TMDB5[2]	TMDB5[1]	TMDB5[0]
		TMDB4[7]	TMDB4[6]	TMDB4[5]	TMDB4[4]	TMDB4[3]	TMDB4[2]	TMDB4[1]	TMDB4[0]
	RSCAN0THLACCm (m = 0, 1)	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		TID[7]	TID[6]	TID[5]	TID[4]	TID[3]	TID[2]	TID[1]	TID[0]
		-	BN[3]	BN[2]	BN[1]	BN[0]	BT[2]	BT[1]	BT[0]
IEBus controller*1	IEBB0BCR	IEBB0PW	IEBB0MSRQ	IEBB0ALRQ	IEBB0STXE	IEBB0SRXE	-	-	-
	IEBB0PSR	IEBB0CLKE	IEBB0CMD	-	-	-	-	-	-
	IEBB0UAR	-	-	-	-				
	IEBB0SAR	-	-	-	-				
	IEBB0PAR	-	-	-	-				
	IEBB0RSA	-	-	-	-				
	IEBB0CDR	-	-	-	-	IEBB0SLCD3	IEBB0SLCD2	IEBB0SLCD1	IEBB0SLCD0
	IEBB0TCD	-	-	-	-	IEBB0SLTD3	IEBB0SLTD2	IEBB0SLTD1	IEBB0SLTD0
	IEBB0RCD	-	-	-	-	IEBB0SLRD3	IEBB0SLRD2	IEBB0SLRD1	IEBB0SLRD0
	IEBB0DLR								
	IEBB0TDL								
	IEBB0RDL								
	IEBB0CKS	-	-	-	IEBB0PRS	-	IEBB0BRS2	IEBB0BRS1	IEBB0BRS0
	IEBB0TMS	IEBB0FMDE	IEBB0SLRI1	IEBB0SLRI0	IEBB0SLTI1	IEBB0SLTI0	IEBB0ALC2	IEBB0ALC1	IEBB0ALC0
	IEBB0PCR	IEBB0CRPT	IEBB0CTPT	-	-	-	-	-	-
	IEBB0BSR	IEBB0RFLF	IEBB0FOVR	-	IEBB0SRFP4	IEBB0SRFP3	IEBB0SRFP2	IEBB0SRFP1	IEBB0SRFP0
		IEBB0TFLF	IEBB0FOVW	-	IEBB0STFP4	IEBB0STFP3	IEBB0STFP2	IEBB0STFP1	IEBB0STFP0
	IEBB0SSR	-	-	-	IEBB0SSLF	-	IEBB0STLF	IEBB0SRXF	IEBB0STXF
	IEBB0USR	-	IEBB0SRQF	IEBB0ARBF	IEBB0ALTF	IEBB0ACKF	IEBB0LCKF	-	-
	IEBB0ISR	-	IEBB0IEBE	IEBB0STRF	IEBB0STSF	IEBB0ETRF	IEBB0EFMF	-	IEBB0FOVE
	IEBB0ESR	IEBB0TIME	IEBB0PARE	IEBB0NACE	IEBB0UNRE	IEBB0OVRE	-	IEBB0ABTE	IEBB0TRDE
	IEBB0FSR	IEBB0RTRF	IEBB0TTRF	-	-	-	-	IEBB0SSFS1	IEBB0SSFS0
	IEBB0SCR								
	IEBB0CCR								
	IEBB0STC0	IEBB0CLTM	IEBB0CLPA	IEBB0CLNC	IEBB0CLUR	IEBB0CLOV	-	IEBB0CLAB	IEBB0CLTR
	IEBB0STC1	-	-	-	-	-	-	-	IEBB0CLFF
	IEBB0DR								
Renesas SPDIF interface	TLCA	-	-	-	-	-	-	-	-
	TRCA	-	-	-	-	-	-	-	-
	TLCS	-	-	CLAC[1]	CLAC[0]	FS[3]	FS[2]	FS[1]	FS[0]
		CHNO[3]	CHNO[2]	CHNO[1]	CHNO[0]	SRCNO[3]	SRCNO[2]	SRCNO[1]	SRCNO[0]
		CATCD[7]	CATCD[6]	CATCD[5]	CATCD[4]	CATCD[3]	CATCD[2]	CATCD[1]	CATCD[0]
		–	–	CTL[4]	CTL[3]	CTL[2]	CTL[1]	CTL[0]	–
	TRCS	-	-	CLAC[1]	CLAC[0]	FS[3]	FS[2]	FS[1]	FS[0]
		CHNO[3]	CHNO[2]	CHNO[1]	CHNO[0]	SRCNO[3]	SRCNO[2]	SRCNO[1]	SRCNO[0]
		CATCD[7]	CATCD[6]	CATCD[5]	CATCD[4]	CATCD[3]	CATCD[2]	CATCD[1]	CATCD[0]
		–	–	CTL[4]	CTL[3]	CTL[2]	CTL[1]	CTL[0]	–
	TUI								

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Renesas SPDIF interface	RLCA	-	-	-	-	-	-	-	-
	RRCA	-	-	-	-	-	-	-	-
	RLCS	-	-	CLAC[1]	CLAC[0]	FS[3]	FS[2]	FS[1]	FS[0]
		CHNO[3]	CHNO[2]	CHNO[1]	CHNO[0]	SRCNO[3]	SRCNO[2]	SRCNO[1]	SRCNO[0]
		CATCD[7]	CATCD[6]	CATCD[5]	CATCD[4]	CATCD[3]	CATCD[2]	CATCD[1]	CATCD[0]
		–	–	CTL[4]	CTL[3]	CTL[2]	CTL[1]	CTL[0]	–
	RRCS	-	-	CLAC[1]	CLAC[0]	FS[3]	FS[2]	FS[1]	FS[0]
		CHNO[3]	CHNO[2]	CHNO[1]	CHNO[0]	SRCNO[3]	SRCNO[2]	SRCNO[1]	SRCNO[0]
		CATCD[7]	CATCD[6]	CATCD[5]	CATCD[4]	CATCD[3]	CATCD[2]	CATCD[1]	CATCD[0]
		–	–	CTL[4]	CTL[3]	CTL[2]	CTL[1]	CTL[0]	–
	RUI								
	CTRL	-	-	-	CKS	-	PB	RASS[1]	RASS[0]
		TASS[1]	TASS[0]	RDE	TDE	NCSI	AOS	RME	TME
		REIE	TEIE	UBOI	UBUI	CREI	PAEI	PREI	CSEI
		ABOI	ABUI	RUII	TUII	RCSI	RCBI	TCSI	TCBI
	STAT	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	CMD
		RIS	TIS	UBO	UBU	CE	PARE	PREE	CSE
		ABO	ABU	RUIR	TUIR	CSRX	CBRX	CSTX	CBTX
	TDAD	-	-	-	-	-	-	-	-
	RDAD	-	-	-	-	-	-	-	-
CD-ROM decoder *1	CROMEN	SUBC_EN	CROM_EN	CROM_STP	-	-	-	-	-
	CROMSY0	SY_AUT	SY_IEN	SY_DEN	-	-	-	-	-
	CROMCTL0	MD_DESC	-	MD_AUTO	MD_AUTOS1	MD_AUTOS2	MD_SEC[2]	MD_SEC[1]	MD_SEC[0]
	CROMCTL1	M2F2EDC	MD_DEC[2]	MD_DEC[1]	MD_DEC[0]	-	-	MD_PQREP[1]	MD_PQREP[0]
	CROMCTL3	STP_ECC	STP_EDC	-	STP_MD	STP_MIN	-	-	-
	CROMCTL4	-	LINK2	-	EROSEL	NO_ECC	-	-	-
	CROMCTL5	-	-	-	-	-	-	-	MSF_LBA_SEL
	CROMST0	-	-	ST_SYIL	ST_SYNO	ST_BLKS	ST_BLKL	ST_SECS	ST_SECL
	CROMST1	-	-	-	-	ER2_HEAD0	ER2_HEAD1	ER2_HEAD2	ER2_HEAD3
	CROMST3	ER2_SHEAD0	ER2_SHEAD1	ER2_SHEAD2	ER2_SHEAD3	ER2_SHEAD4	ER2_SHEAD5	ER2_SHEAD6	ER2_SHEAD7
	CROMST4	NG_MD	NG_MDCMP1	NG_MDCMP2	NG_MDCMP3	NG_MDCMP4	NG_MDDEF	NG_MDTIM1	NG_MDTIM2
	CROMST5	ST_AMD[2]	ST_AMD[1]	ST_AMD[0]	ST_MDX	LINK_ON	LINK_DET	LINK_SDET	LINK_OUT1
	CROMST6	ST_ERR	-	ST_ECCABT	ST_ECCNG	ST_ECCP	ST_ECCQ	ST_EDC1	ST_EDC2
	CBUFST0	BUF_REF	BUF_ACT	-	-	-	-	-	-
	CBUFST1	BUF_ECC	BUF_EDC	-	BUF_MD	BUF_MIN	-	-	-
	CBUFST2	BUF_NG	-	-	-	-	-	-	-
	HEAD00	HEAD00[7]	HEAD00[6]	HEAD00[5]	HEAD00[4]	HEAD00[3]	HEAD00[2]	HEAD00[1]	HEAD00[0]
	HEAD01	HEAD01[7]	HEAD01[6]	HEAD01[5]	HEAD01[4]	HEAD01[3]	HEAD01[2]	HEAD01[1]	HEAD01[0]
	HEAD02	HEAD02[7]	HEAD02[6]	HEAD02[5]	HEAD02[4]	HEAD02[3]	HEAD02[2]	HEAD02[1]	HEAD02[0]
	HEAD03	HEAD03[7]	HEAD03[6]	HEAD03[5]	HEAD03[4]	HEAD03[3]	HEAD03[2]	HEAD03[1]	HEAD03[0]
	SHEAD00	SHEAD00[7]	SHEAD00[6]	SHEAD00[5]	SHEAD00[4]	SHEAD00[3]	SHEAD00[2]	SHEAD00[1]	SHEAD00[0]
	SHEAD01	SHEAD01[7]	SHEAD01[6]	SHEAD01[5]	SHEAD01[4]	SHEAD01[3]	SHEAD01[2]	SHEAD01[1]	SHEAD01[0]
	SHEAD02	SHEAD02[7]	SHEAD02[6]	SHEAD02[5]	SHEAD02[4]	SHEAD02[3]	SHEAD02[2]	SHEAD02[1]	SHEAD02[0]
	SHEAD03	SHEAD03[7]	SHEAD03[6]	SHEAD03[5]	SHEAD03[4]	SHEAD03[3]	SHEAD03[2]	SHEAD03[1]	SHEAD03[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
CD-ROM decoder *1	SHEAD04	SHEAD04[7]	SHEAD04[6]	SHEAD04[5]	SHEAD04[4]	SHEAD04[3]	SHEAD04[2]	SHEAD04[1]	SHEAD04[0]
	SHEAD05	SHEAD05[7]	SHEAD05[6]	SHEAD05[5]	SHEAD05[4]	SHEAD05[3]	SHEAD05[2]	SHEAD05[1]	SHEAD05[0]
	SHEAD06	SHEAD06[7]	SHEAD06[6]	SHEAD06[5]	SHEAD06[4]	SHEAD06[3]	SHEAD06[2]	SHEAD06[1]	SHEAD06[0]
	SHEAD07	SHEAD07[7]	SHEAD07[6]	SHEAD07[5]	SHEAD07[4]	SHEAD07[3]	SHEAD07[2]	SHEAD07[1]	SHEAD07[0]
	HEAD20	HEAD20[7]	HEAD20[6]	HEAD20[5]	HEAD20[4]	HEAD20[3]	HEAD20[2]	HEAD20[1]	HEAD20[0]
	HEAD21	HEAD21[7]	HEAD21[6]	HEAD21[5]	HEAD21[4]	HEAD21[3]	HEAD21[2]	HEAD21[1]	HEAD21[0]
	HEAD22	HEAD22[7]	HEAD22[6]	HEAD22[5]	HEAD22[4]	HEAD22[3]	HEAD22[2]	HEAD22[1]	HEAD22[0]
	HEAD23	HEAD23[7]	HEAD23[6]	HEAD23[5]	HEAD23[4]	HEAD23[3]	HEAD23[2]	HEAD23[1]	HEAD23[0]
	SHEAD20	SHEAD20[7]	SHEAD20[6]	SHEAD20[5]	SHEAD20[4]	SHEAD20[3]	SHEAD20[2]	SHEAD20[1]	SHEAD20[0]
	SHEAD21	SHEAD21[7]	SHEAD21[6]	SHEAD21[5]	SHEAD21[4]	SHEAD21[3]	SHEAD21[2]	SHEAD21[1]	SHEAD21[0]
	SHEAD22	SHEAD22[7]	SHEAD22[6]	SHEAD22[5]	SHEAD22[4]	SHEAD22[3]	SHEAD22[2]	SHEAD22[1]	SHEAD22[0]
	SHEAD23	SHEAD23[7]	SHEAD23[6]	SHEAD23[5]	SHEAD23[4]	SHEAD23[3]	SHEAD23[2]	SHEAD23[1]	SHEAD23[0]
	SHEAD24	SHEAD24[7]	SHEAD24[6]	SHEAD24[5]	SHEAD24[4]	SHEAD24[3]	SHEAD24[2]	SHEAD24[1]	SHEAD24[0]
	SHEAD25	SHEAD25[7]	SHEAD25[6]	SHEAD25[5]	SHEAD25[4]	SHEAD25[3]	SHEAD25[2]	SHEAD25[1]	SHEAD25[0]
	SHEAD26	SHEAD26[7]	SHEAD26[6]	SHEAD26[5]	SHEAD26[4]	SHEAD26[3]	SHEAD26[2]	SHEAD26[1]	SHEAD26[0]
	SHEAD27	SHEAD27[7]	SHEAD27[6]	SHEAD27[5]	SHEAD27[4]	SHEAD27[3]	SHEAD27[2]	SHEAD27[1]	SHEAD27[0]
	CBUFCTL0	CBUF_AUT	CBUF_EN	-	CBUF_MD[1]	CBUF_MD[0]	CBUF_TS	CBUF_Q	-
	CBUFCTL1	BS_MIN[7]	BS_MIN[6]	BS_MIN[5]	BS_MIN[4]	BS_MIN[3]	BS_MIN[2]	BS_MIN[1]	BS_MIN[0]
	CBUFCTL2	BS_SEC[7]	BS_SEC[6]	BS_SEC[5]	BS_SEC[4]	BS_SEC[3]	BS_SEC[2]	BS_SEC[1]	BS_SEC[0]
	CBUFCTL3	BS_FRM[7]	BS_FRM[6]	BS_FRM[5]	BS_FRM[4]	BS_FRM[3]	BS_FRM[2]	BS_FRM[1]	BS_FRM[0]
	CROMST0M	-	-	ST_SYILM	ST_SYNOM	ST_BLKSM	ST_BLKLM	ST_SECSM	ST_SECLM
	ROMDECRST	LOGICRST	RAMRST	-	-	-	-	-	-
	RSTSTAT	RAMCLRST	-	-	-	-	-	-	-
	SSI	BYTEND	BITEND	BUFEND0[1]	BUFEND0[0]	BUFEND1[1]	BUFEND1[0]	-	-
	INTHOLD	ISEC	ITARG	ISY	IERR	IBUF	IREADY	-	-
	INHINT	INHISEC	INHITARG	INHISY	INHIERR	INHIBUF	INHIREADY	PREINHREQDM	PREINHIREADY
	STRMDIN0	STRMDIN[31]	STRMDIN[30]	STRMDIN[29]	STRMDIN[28]	STRMDIN[27]	STRMDIN[26]	STRMDIN[25]	STRMDIN[24]
		STRMDIN[23]	STRMDIN[22]	STRMDIN[21]	STRMDIN[20]	STRMDIN[19]	STRMDIN[18]	STRMDIN[17]	STRMDIN[16]
	STRMDIN2	STRMDIN[15]	STRMDIN[14]	STRMDIN[13]	STRMDIN[12]	STRMDIN[11]	STRMDIN[10]	STRMDIN[9]	STRMDIN[8]
		STRMDIN[7]	STRMDIN[6]	STRMDIN[5]	STRMDIN[4]	STRMDIN[3]	STRMDIN[2]	STRMDIN[1]	STRMDIN[0]
	STRMDOUT0	STRMDOUT[15]	STRMDOUT[14]	STRMDOUT[13]	STRMDOUT[12]	STRMDOUT[11]	STRMDOUT[10]	STRMDOUT[9]	STRMDOUT[8]
		STRMDOUT[7]	STRMDOUT[6]	STRMDOUT[5]	STRMDOUT[4]	STRMDOUT[3]	STRMDOUT[2]	STRMDOUT[1]	STRMDOUT[0]
LIN interface LIN master*1	RLN30LWBR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		NSPB[3]	NSPB[2]	NSPB[1]	NSPB[0]	LPRS[2]	LPRS[1]	LPRS[0]	LWBR0
	RLN30LBRP0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		LBRP0[7]	LBRP0[6]	LBRP0[5]	LBRP0[4]	LBRP0[3]	LBRP0[2]	LBRP0[1]	LBRP0[0]
	RLN30LBRP1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		LBRP1[7]	LBRP1[6]	LBRP1[5]	LBRP1[4]	LBRP1[3]	LBRP1[2]	LBRP1[1]	LBRP1[0]
	RLN30LSTC	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	LSTM
	RLN30LMD	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	LRDNFS	LIOS	LCKS[1]	LCKS[0]	LMD[1]	LMD[0]
	RLN30LBFC	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	BDT[1]	BDT[0]	BLT[3]	BLT[2]	BLT[1]	BLT[0]
	RLN30LSC	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	IBS[1]	IBS[0]	-	IBHS[2]	IBHS[1]	IBHS[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
LIN interface LIN master*1	RLN30LWUP	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		WUTL[3]	WUTL[2]	WUTL[1]	WUTL[0]	-	-	-	-
	RLN30LIE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	SHIE	ERRIE	FRCIE	FTCIE
	RLN30LEDE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		LTES	-	-	-	FERE	FTERE	PBERE	BERE
	RLN30LCUC	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	OM1	OM0
	RLN30LTRC	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	RTS	FTS
	RLN30LMST	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	OMM1	OMM0
	RLN30LST	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		HTRC	D1RC	-	-	ERR	-	FRC	FTC
	RLN30LEST	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		RPER	-	CSER	-	FER	FTER	PBER	BER
	RLN30LDFC	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		LSS	FSM	CSM	RFT	RFDL[3]	RFDL[2]	RFDL[1]	RFDL[0]
	RLN30LIDB	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		IDP1	IDP0	ID[5]	ID[4]	ID[3]	ID[2]	ID[1]	ID[0]
	RLN30LCBR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		CKSM[7]	CKSM[6]	CKSM[5]	CKSM[4]	CKSM[3]	CKSM[2]	CKSM[1]	CKSM[0]
	RLN30LDBR1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		LDB[7]	LDB[6]	LDB[5]	LDB[4]	LDB[3]	LDB[2]	LDB[1]	LDB[0]
	RLN30LDBR2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		LDB[7]	LDB[6]	LDB[5]	LDB[4]	LDB[3]	LDB[2]	LDB[1]	LDB[0]
	RLN30LDBR3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		LDB[7]	LDB[6]	LDB[5]	LDB[4]	LDB[3]	LDB[2]	LDB[1]	LDB[0]
	RLN30LDBR4	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		LDB[7]	LDB[6]	LDB[5]	LDB[4]	LDB[3]	LDB[2]	LDB[1]	LDB[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
LIN interface LIN master*1	RLN30LDBR5	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		LDB[7]	LDB[6]	LDB[5]	LDB[4]	LDB[3]	LDB[2]	LDB[1]	LDB[0]
	RLN30LDBR6	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		LDB[7]	LDB[6]	LDB[5]	LDB[4]	LDB[3]	LDB[2]	LDB[1]	LDB[0]
	RLN30LDBR7	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		LDB[7]	LDB[6]	LDB[5]	LDB[4]	LDB[3]	LDB[2]	LDB[1]	LDB[0]
	RLN30LDBR8	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		LDB[7]	LDB[6]	LDB[5]	LDB[4]	LDB[3]	LDB[2]	LDB[1]	LDB[0]
Ethernet controller	ARSTR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	ARST
	ECMR0	-	-	-	-	-	TRCCM	-	-
		RCSC	-	DPAD	RZPF	ZPF	PFR	RXF	TXXF
		-	-	MCT	-	-	-	-	-
		-	RE	TE	-	-	-	DM	PRM
	ECSR0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	PFROI	-	-	-	ICD
	ECSIPR0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	PFROIP	-	-	-	ICDIP
	PIR0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	MDI	MDO	MMD	MDC
	MAHR0	MA[47]	MA[46]	MA[45]	MA[44]	MA[43]	MA[42]	MA[41]	MA[40]
		MA[39]	MA[38]	MA[37]	MA[36]	MA[35]	MA[34]	MA[33]	MA[32]
		MA[31]	MA[30]	MA[29]	MA[28]	MA[27]	MA[26]	MA[25]	MA[24]
		MA[23]	MA[22]	MA[21]	MA[20]	MA[19]	MA[18]	MA[17]	MA[16]
	MALR0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		MA[15]	MA[14]	MA[13]	MA[12]	MA[11]	MA[10]	MA[9]	MA[8]
		MA[7]	MA[6]	MA[5]	MA[4]	MA[3]	MA[2]	MA[1]	MA[0]
	RFLR0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	RFL[17]	RFL[16]
		RFL[15]	RFL[14]	RFL[13]	RFL[12]	RFL[11]	RFL[10]	RFL[9]	RFL[8]
		RFL[7]	RFL[6]	RFL[5]	RFL[4]	RFL[3]	RFL[2]	RFL[1]	RFL[0]
	CEFCR0	CEFC[31]	CEFC[30]	CEFC[29]	CEFC[28]	CEFC[27]	CEFC[26]	CEFC[25]	CEFC[24]
		CEFC[23]	CEFC[22]	CEFC[21]	CEFC[20]	CEFC[19]	CEFC[18]	CEFC[17]	CEFC[16]
		CEFC[15]	CEFC[14]	CEFC[13]	CEFC[12]	CEFC[11]	CEFC[10]	CEFC[9]	CEFC[8]
		CEFC[7]	CEFC[6]	CEFC[5]	CEFC[4]	CEFC[3]	CEFC[2]	CEFC[1]	CEFC[0]
	FRECR0	FREC[31]	FREC[30]	FREC[29]	FREC[28]	FREC[27]	FREC[26]	FREC[25]	FREC[24]
		FREC[23]	FREC[22]	FREC[21]	FREC[20]	FREC[19]	FREC[18]	FREC[17]	FREC[16]
		FREC[15]	FREC[14]	FREC[13]	FREC[12]	FREC[11]	FREC[10]	FREC[9]	FREC[8]
		FREC[7]	FREC[6]	FREC[5]	FREC[4]	FREC[3]	FREC[2]	FREC[1]	FREC[0]
	TSFRCR0	TSFC[31]	TSFC[30]	TSFC[29]	TSFC[28]	TSFC[27]	TSFC[26]	TSFC[25]	TSFC[24]
		TSFC[23]	TSFC[22]	TSFC[21]	TSFC[20]	TSFC[19]	TSFC[18]	TSFC[17]	TSFC[16]
		TSFC[15]	TSFC[14]	TSFC[13]	TSFC[12]	TSFC[11]	TSFC[10]	TSFC[9]	TSFC[8]
		TSFC[7]	TSFC[6]	TSFC[5]	TSFC[4]	TSFC[3]	TSFC[2]	TSFC[1]	TSFC[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Ethernet controller	TLFRCR0	TLFC[31]	TLFC[30]	TLFC[29]	TLFC[28]	TLFC[27]	TLFC[26]	TLFC[25]	TLFC[24]
		TLFC[23]	TLFC[22]	TLFC[21]	TLFC[20]	TLFC[19]	TLFC[18]	TLFC[17]	TLFC[16]
		TLFC[15]	TLFC[14]	TLFC[13]	TLFC[12]	TLFC[11]	TLFC[10]	TLFC[9]	TLFC[8]
		TLFC[7]	TLFC[6]	TLFC[5]	TLFC[4]	TLFC[3]	TLFC[2]	TLFC[1]	TLFC[0]
	RFCR0	RFC[31]	RFC[30]	RFC[29]	RFC[28]	RFC[27]	RFC[26]	RFC[25]	RFC[24]
		RFC[23]	RFC[22]	RFC[21]	RFC[20]	RFC[19]	RFC[18]	RFC[17]	RFC[16]
		RFC[15]	RFC[14]	RFC[13]	RFC[12]	RFC[11]	RFC[10]	RFC[9]	RFC[8]
		RFC[7]	RFC[6]	RFC[5]	RFC[4]	RFC[3]	RFC[2]	RFC[1]	RFC[0]
	MAFCR0	MAFC[31]	MAFC[30]	MAFC[29]	MAFC[28]	MAFC[27]	MAFC[26]	MAFC[25]	MAFC[24]
		MAFC[23]	MAFC[22]	MAFC[21]	MAFC[20]	MAFC[19]	MAFC[18]	MAFC[17]	MAFC[16]
		MAFC[15]	MAFC[14]	MAFC[13]	MAFC[12]	MAFC[11]	MAFC[10]	MAFC[9]	MAFC[8]
		MAFC[7]	MAFC[6]	MAFC[5]	MAFC[4]	MAFC[3]	MAFC[2]	MAFC[1]	MAFC[0]
	APR0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		AP[15]	AP[14]	AP[13]	AP[12]	AP[11]	AP[10]	AP[9]	AP[8]
		AP[7]	AP[6]	AP[5]	AP[4]	AP[3]	AP[2]	AP[1]	AP[0]
	MPR0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		MP[15]	MP[14]	MP[13]	MP[12]	MP[11]	MP[10]	MP[9]	MP[8]
		MP[7]	MP[6]	MP[5]	MP[4]	MP[3]	MP[2]	MP[1]	MP[0]
	TPAUSER0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		TPAUSE[15]	TPAUSE[14]	TPAUSE[13]	TPAUSE[12]	TPAUSE[11]	TPAUSE[10]	TPAUSE[9]	TPAUSE[8]
		TPAUSE[7]	TPAUSE[6]	TPAUSE[5]	TPAUSE[4]	TPAUSE[3]	TPAUSE[2]	TPAUSE[1]	TPAUSE[0]
	PFTCR0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		PFTXC[15]	PFTXC[14]	PFTXC[13]	PFTXC[12]	PFTXC[11]	PFTXC[10]	PFTXC[9]	PFTXC[8]
		PFTXC[7]	PFTXC[6]	PFTXC[5]	PFTXC[4]	PFTXC[3]	PFTXC[2]	PFTXC[1]	PFTXC[0]
	PFRCR0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		PFRXC[15]	PFRXC[14]	PFRXC[13]	PFRXC[12]	PFRXC[11]	PFRXC[10]	PFRXC[9]	PFRXC[8]
		PFRXC[7]	PFRXC[6]	PFRXC[5]	PFRXC[4]	PFRXC[3]	PFRXC[2]	PFRXC[1]	PFRXC[0]
	TSU_CTRST	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	CTRST
		-	-	-	-	-	-	-	-
	TSU_FWSLC	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	POSTENU	POSTENL	-	-	-	-
		-	-	-	-	-	-	-	-
	TSU_VTAG0	VTAG0	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	VID0[11]	VID0[10]	VID0[9]	VID0[8]
		VID0[7]	VID0[6]	VID0[5]	VID0[4]	VID0[3]	VID0[2]	VID0[1]	VID0[0]
	TSU_ADSBSY	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	ADSBSY
	TSU_TEN	TEN0	TEN1	TEN2	TEN3	TEN4	TEN5	TEN6	TEN7
		TEN8	TEN9	TEN10	TEN11	TEN12	TEN13	TEN14	TEN15
		TEN16	TEN17	TEN18	TEN19	TEN20	TEN21	TEN22	TEN23
		TEN24	TEN25	TEN26	TEN27	TEN28	TEN29	TEN30	TEN31
	TSU_POST1	POST0	-	-	-	POST1	-	-	-
		POST2	-	-	-	POST3	-	-	-
		POST4	-	-	-	POST5	-	-	-
		POST6	-	-	-	POST7	-	-	-
	TSU_POST2	POST8	-	-	-	POST9	-	-	-
		POST10	-	-	-	POST11	-	-	-
		POST12	-	-	-	POST13	-	-	-
		POST14	-	-	-	POST15	-	-	-

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Ethernet controller	TSU_POST3	POST16	-	-	-	POST17	-	-	-
		POST18	-	-	-	POST19	-	-	-
		POST20	-	-	-	POST21	-	-	-
		POST22	-	-	-	POST23	-	-	-
	TSU_POST4	POST24	-	-	-	POST25	-	-	-
		POST26	-	-	-	POST27	-	-	-
		POST28	-	-	-	POST29	-	-	-
		POST30	-	-	-	POST31	-	-	-
	TSU_ADRH0	ADRH0[31]	ADRH0[30]	ADRH0[29]	ADRH0[28]	ADRH0[27]	ADRH0[26]	ADRH0[25]	ADRH0[24]
		ADRH0[23]	ADRH0[22]	ADRH0[21]	ADRH0[20]	ADRH0[19]	ADRH0[18]	ADRH0[17]	ADRH0[16]
		ADRH0[15]	ADRH0[14]	ADRH0[13]	ADRH0[12]	ADRH0[11]	ADRH0[10]	ADRH0[9]	ADRH0[8]
		ADRH0[7]	ADRH0[6]	ADRH0[5]	ADRH0[4]	ADRH0[3]	ADRH0[2]	ADRH0[1]	ADRH0[0]
	TSU_ADRH1	ADRH1[31]	ADRH1[30]	ADRH1[29]	ADRH1[28]	ADRH1[27]	ADRH1[26]	ADRH1[25]	ADRH1[24]
		ADRH1[23]	ADRH1[22]	ADRH1[21]	ADRH1[20]	ADRH1[19]	ADRH1[18]	ADRH1[17]	ADRH1[16]
		ADRH1[15]	ADRH1[14]	ADRH1[13]	ADRH1[12]	ADRH1[11]	ADRH1[10]	ADRH1[9]	ADRH1[8]
		ADRH1[7]	ADRH1[6]	ADRH1[5]	ADRH1[4]	ADRH1[3]	ADRH1[2]	ADRH1[1]	ADRH1[0]
	TSU_ADRH2	ADRH2[31]	ADRH2[30]	ADRH2[29]	ADRH2[28]	ADRH2[27]	ADRH2[26]	ADRH2[25]	ADRH2[24]
		ADRH2[23]	ADRH2[22]	ADRH2[21]	ADRH2[20]	ADRH2[19]	ADRH2[18]	ADRH2[17]	ADRH2[16]
		ADRH2[15]	ADRH2[14]	ADRH2[13]	ADRH2[12]	ADRH2[11]	ADRH2[10]	ADRH2[9]	ADRH2[8]
		ADRH2[7]	ADRH2[6]	ADRH2[5]	ADRH2[4]	ADRH2[3]	ADRH2[2]	ADRH2[1]	ADRH2[0]
	TSU_ADRH3	ADRH3[31]	ADRH3[30]	ADRH3[29]	ADRH3[28]	ADRH3[27]	ADRH3[26]	ADRH3[25]	ADRH3[24]
		ADRH3[23]	ADRH3[22]	ADRH3[21]	ADRH3[20]	ADRH3[19]	ADRH3[18]	ADRH3[17]	ADRH3[16]
		ADRH3[15]	ADRH3[14]	ADRH3[13]	ADRH3[12]	ADRH3[11]	ADRH3[10]	ADRH3[9]	ADRH3[8]
		ADRH3[7]	ADRH3[6]	ADRH3[5]	ADRH3[4]	ADRH3[3]	ADRH3[2]	ADRH3[1]	ADRH3[0]
	TSU_ADRH4	ADRH4[31]	ADRH4[30]	ADRH4[29]	ADRH4[28]	ADRH4[27]	ADRH4[26]	ADRH4[25]	ADRH4[24]
		ADRH4[23]	ADRH4[22]	ADRH4[21]	ADRH4[20]	ADRH4[19]	ADRH4[18]	ADRH4[17]	ADRH4[16]
		ADRH4[15]	ADRH4[14]	ADRH4[13]	ADRH4[12]	ADRH4[11]	ADRH4[10]	ADRH4[9]	ADRH4[8]
		ADRH4[7]	ADRH4[6]	ADRH4[5]	ADRH4[4]	ADRH4[3]	ADRH4[2]	ADRH4[1]	ADRH4[0]
	TSU_ADRH5	ADRH5[31]	ADRH5[30]	ADRH5[29]	ADRH5[28]	ADRH5[27]	ADRH5[26]	ADRH5[25]	ADRH5[24]
		ADRH5[23]	ADRH5[22]	ADRH5[21]	ADRH5[20]	ADRH5[19]	ADRH5[18]	ADRH5[17]	ADRH5[16]
		ADRH5[15]	ADRH5[14]	ADRH5[13]	ADRH5[12]	ADRH5[11]	ADRH5[10]	ADRH5[9]	ADRH5[8]
		ADRH5[7]	ADRH5[6]	ADRH5[5]	ADRH5[4]	ADRH5[3]	ADRH5[2]	ADRH5[1]	ADRH5[0]
	TSU_ADRH6	ADRH6[31]	ADRH6[30]	ADRH6[29]	ADRH6[28]	ADRH6[27]	ADRH6[26]	ADRH6[25]	ADRH6[24]
		ADRH6[23]	ADRH6[22]	ADRH6[21]	ADRH6[20]	ADRH6[19]	ADRH6[18]	ADRH6[17]	ADRH6[16]
		ADRH6[15]	ADRH6[14]	ADRH6[13]	ADRH6[12]	ADRH6[11]	ADRH6[10]	ADRH6[9]	ADRH6[8]
		ADRH6[7]	ADRH6[6]	ADRH6[5]	ADRH6[4]	ADRH6[3]	ADRH6[2]	ADRH6[1]	ADRH6[0]
	TSU_ADRH7	ADRH7[31]	ADRH7[30]	ADRH7[29]	ADRH7[28]	ADRH7[27]	ADRH7[26]	ADRH7[25]	ADRH7[24]
		ADRH7[23]	ADRH7[22]	ADRH7[21]	ADRH7[20]	ADRH7[19]	ADRH7[18]	ADRH7[17]	ADRH7[16]
		ADRH7[15]	ADRH7[14]	ADRH7[13]	ADRH7[12]	ADRH7[11]	ADRH7[10]	ADRH7[9]	ADRH7[8]
		ADRH7[7]	ADRH7[6]	ADRH7[5]	ADRH7[4]	ADRH7[3]	ADRH7[2]	ADRH7[1]	ADRH7[0]
	TSU_ADRH8	ADRH8[31]	ADRH8[30]	ADRH8[29]	ADRH8[28]	ADRH8[27]	ADRH8[26]	ADRH8[25]	ADRH8[24]
		ADRH8[23]	ADRH8[22]	ADRH8[21]	ADRH8[20]	ADRH8[19]	ADRH8[18]	ADRH8[17]	ADRH8[16]
		ADRH8[15]	ADRH8[14]	ADRH8[13]	ADRH8[12]	ADRH8[11]	ADRH8[10]	ADRH8[9]	ADRH8[8]
		ADRH8[7]	ADRH8[6]	ADRH8[5]	ADRH8[4]	ADRH8[3]	ADRH8[2]	ADRH8[1]	ADRH8[0]
	TSU_ADRH9	ADRH9[31]	ADRH9[30]	ADRH9[29]	ADRH9[28]	ADRH9[27]	ADRH9[26]	ADRH9[25]	ADRH9[24]
		ADRH9[23]	ADRH9[22]	ADRH9[21]	ADRH9[20]	ADRH9[19]	ADRH9[18]	ADRH9[17]	ADRH9[16]
		ADRH9[15]	ADRH9[14]	ADRH9[13]	ADRH9[12]	ADRH9[11]	ADRH9[10]	ADRH9[9]	ADRH9[8]
		ADRH9[7]	ADRH9[6]	ADRH9[5]	ADRH9[4]	ADRH9[3]	ADRH9[2]	ADRH9[1]	ADRH9[0]
	TSU_ADRH10	ADRH10[31]	ADRH10[30]	ADRH10[29]	ADRH10[28]	ADRH10[27]	ADRH10[26]	ADRH10[25]	ADRH10[24]
		ADRH10[23]	ADRH10[22]	ADRH10[21]	ADRH10[20]	ADRH10[19]	ADRH10[18]	ADRH10[17]	ADRH10[16]
		ADRH10[15]	ADRH10[14]	ADRH10[13]	ADRH10[12]	ADRH10[11]	ADRH10[10]	ADRH10[9]	ADRH10[8]
		ADRH10[7]	ADRH10[6]	ADRH10[5]	ADRH10[4]	ADRH10[3]	ADRH10[2]	ADRH10[1]	ADRH10[0]
	TSU_ADRH11	ADRH11[31]	ADRH11[30]	ADRH11[29]	ADRH11[28]	ADRH11[27]	ADRH11[26]	ADRH11[25]	ADRH11[24]
		ADRH11[23]	ADRH11[22]	ADRH11[21]	ADRH11[20]	ADRH11[19]	ADRH11[18]	ADRH11[17]	ADRH11[16]
		ADRH11[15]	ADRH11[14]	ADRH11[13]	ADRH11[12]	ADRH11[11]	ADRH11[10]	ADRH11[9]	ADRH11[8]
		ADRH11[7]	ADRH11[6]	ADRH11[5]	ADRH11[4]	ADRH11[3]	ADRH11[2]	ADRH11[1]	ADRH11[0]
	TSU_ADRH12	ADRH12[31]	ADRH12[30]	ADRH12[29]	ADRH12[28]	ADRH12[27]	ADRH12[26]	ADRH12[25]	ADRH12[24]
		ADRH12[23]	ADRH12[22]	ADRH12[21]	ADRH12[20]	ADRH12[19]	ADRH12[18]	ADRH12[17]	ADRH12[16]
		ADRH12[15]	ADRH12[14]	ADRH12[13]	ADRH12[12]	ADRH12[11]	ADRH12[10]	ADRH12[9]	ADRH12[8]
		ADRH12[7]	ADRH12[6]	ADRH12[5]	ADRH12[4]	ADRH12[3]	ADRH12[2]	ADRH12[1]	ADRH12[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Ethernet controller	TSU_ADRH13	ADRH13[31]	ADRH13[30]	ADRH13[29]	ADRH13[28]	ADRH13[27]	ADRH13[26]	ADRH13[25]	ADRH13[24]
		ADRH13[23]	ADRH13[22]	ADRH13[21]	ADRH13[20]	ADRH13[19]	ADRH13[18]	ADRH13[17]	ADRH13[16]
		ADRH13[15]	ADRH13[14]	ADRH13[13]	ADRH13[12]	ADRH13[11]	ADRH13[10]	ADRH13[9]	ADRH13[8]
		ADRH13[7]	ADRH13[6]	ADRH13[5]	ADRH13[4]	ADRH13[3]	ADRH13[2]	ADRH13[1]	ADRH13[0]
	TSU_ADRH14	ADRH14[31]	ADRH14[30]	ADRH14[29]	ADRH14[28]	ADRH14[27]	ADRH14[26]	ADRH14[25]	ADRH14[24]
		ADRH14[23]	ADRH14[22]	ADRH14[21]	ADRH14[20]	ADRH14[19]	ADRH14[18]	ADRH14[17]	ADRH14[16]
		ADRH14[15]	ADRH14[14]	ADRH14[13]	ADRH14[12]	ADRH14[11]	ADRH14[10]	ADRH14[9]	ADRH14[8]
		ADRH14[7]	ADRH14[6]	ADRH14[5]	ADRH14[4]	ADRH14[3]	ADRH14[2]	ADRH14[1]	ADRH14[0]
	TSU_ADRH15	ADRH15[31]	ADRH15[30]	ADRH15[29]	ADRH15[28]	ADRH15[27]	ADRH15[26]	ADRH15[25]	ADRH15[24]
		ADRH15[23]	ADRH15[22]	ADRH15[21]	ADRH15[20]	ADRH15[19]	ADRH15[18]	ADRH15[17]	ADRH15[16]
		ADRH15[15]	ADRH15[14]	ADRH15[13]	ADRH15[12]	ADRH15[11]	ADRH15[10]	ADRH15[9]	ADRH15[8]
		ADRH15[7]	ADRH15[6]	ADRH15[5]	ADRH15[4]	ADRH15[3]	ADRH15[2]	ADRH15[1]	ADRH15[0]
	TSU_ADRH16	ADRH16[31]	ADRH16[30]	ADRH16[29]	ADRH16[28]	ADRH16[27]	ADRH16[26]	ADRH16[25]	ADRH16[24]
		ADRH16[23]	ADRH16[22]	ADRH16[21]	ADRH16[20]	ADRH16[19]	ADRH16[18]	ADRH16[17]	ADRH16[16]
		ADRH16[15]	ADRH16[14]	ADRH16[13]	ADRH16[12]	ADRH16[11]	ADRH16[10]	ADRH16[9]	ADRH16[8]
		ADRH16[7]	ADRH16[6]	ADRH16[5]	ADRH16[4]	ADRH16[3]	ADRH16[2]	ADRH16[1]	ADRH16[0]
	TSU_ADRH17	ADRH17[31]	ADRH17[30]	ADRH17[29]	ADRH17[28]	ADRH17[27]	ADRH17[26]	ADRH17[25]	ADRH17[24]
		ADRH17[23]	ADRH17[22]	ADRH17[21]	ADRH17[20]	ADRH17[19]	ADRH17[18]	ADRH17[17]	ADRH17[16]
		ADRH17[15]	ADRH17[14]	ADRH17[13]	ADRH17[12]	ADRH17[11]	ADRH17[10]	ADRH17[9]	ADRH17[8]
		ADRH17[7]	ADRH17[6]	ADRH17[5]	ADRH17[4]	ADRH17[3]	ADRH17[2]	ADRH17[1]	ADRH17[0]
	TSU_ADRH18	ADRH18[31]	ADRH18[30]	ADRH18[29]	ADRH18[28]	ADRH18[27]	ADRH18[26]	ADRH18[25]	ADRH18[24]
		ADRH18[23]	ADRH18[22]	ADRH18[21]	ADRH18[20]	ADRH18[19]	ADRH18[18]	ADRH18[17]	ADRH18[16]
		ADRH18[15]	ADRH18[14]	ADRH18[13]	ADRH18[12]	ADRH18[11]	ADRH18[10]	ADRH18[9]	ADRH18[8]
		ADRH18[7]	ADRH18[6]	ADRH18[5]	ADRH18[4]	ADRH18[3]	ADRH18[2]	ADRH18[1]	ADRH18[0]
	TSU_ADRH19	ADRH19[31]	ADRH19[30]	ADRH19[29]	ADRH19[28]	ADRH19[27]	ADRH19[26]	ADRH19[25]	ADRH19[24]
		ADRH19[23]	ADRH19[22]	ADRH19[21]	ADRH19[20]	ADRH19[19]	ADRH19[18]	ADRH19[17]	ADRH19[16]
		ADRH19[15]	ADRH19[14]	ADRH19[13]	ADRH19[12]	ADRH19[11]	ADRH19[10]	ADRH19[9]	ADRH19[8]
		ADRH19[7]	ADRH19[6]	ADRH19[5]	ADRH19[4]	ADRH19[3]	ADRH19[2]	ADRH19[1]	ADRH19[0]
	TSU_ADRH20	ADRH20[31]	ADRH20[30]	ADRH20[29]	ADRH20[28]	ADRH20[27]	ADRH20[26]	ADRH20[25]	ADRH20[24]
		ADRH20[23]	ADRH20[22]	ADRH20[21]	ADRH20[20]	ADRH20[19]	ADRH20[18]	ADRH20[17]	ADRH20[16]
		ADRH20[15]	ADRH20[14]	ADRH20[13]	ADRH20[12]	ADRH20[11]	ADRH20[10]	ADRH20[9]	ADRH20[8]
		ADRH20[7]	ADRH20[6]	ADRH20[5]	ADRH20[4]	ADRH20[3]	ADRH20[2]	ADRH20[1]	ADRH20[0]
	TSU_ADRH21	ADRH21[31]	ADRH21[30]	ADRH21[29]	ADRH21[28]	ADRH21[27]	ADRH21[26]	ADRH21[25]	ADRH21[24]
		ADRH21[23]	ADRH21[22]	ADRH21[21]	ADRH21[20]	ADRH21[19]	ADRH21[18]	ADRH21[17]	ADRH21[16]
		ADRH21[15]	ADRH21[14]	ADRH21[13]	ADRH21[12]	ADRH21[11]	ADRH21[10]	ADRH21[9]	ADRH21[8]
		ADRH21[7]	ADRH21[6]	ADRH21[5]	ADRH21[4]	ADRH21[3]	ADRH21[2]	ADRH21[1]	ADRH21[0]
	TSU_ADRH22	ADRH22[31]	ADRH22[30]	ADRH22[29]	ADRH22[28]	ADRH22[27]	ADRH22[26]	ADRH22[25]	ADRH22[24]
		ADRH22[23]	ADRH22[22]	ADRH22[21]	ADRH22[20]	ADRH22[19]	ADRH22[18]	ADRH22[17]	ADRH22[16]
		ADRH22[15]	ADRH22[14]	ADRH22[13]	ADRH22[12]	ADRH22[11]	ADRH22[10]	ADRH22[9]	ADRH22[8]
		ADRH22[7]	ADRH22[6]	ADRH22[5]	ADRH22[4]	ADRH22[3]	ADRH22[2]	ADRH22[1]	ADRH22[0]
	TSU_ADRH23	ADRH23[31]	ADRH23[30]	ADRH23[29]	ADRH23[28]	ADRH23[27]	ADRH23[26]	ADRH23[25]	ADRH23[24]
		ADRH23[23]	ADRH23[22]	ADRH23[21]	ADRH23[20]	ADRH23[19]	ADRH23[18]	ADRH23[17]	ADRH23[16]
		ADRH23[15]	ADRH23[14]	ADRH23[13]	ADRH23[12]	ADRH23[11]	ADRH23[10]	ADRH23[9]	ADRH23[8]
		ADRH23[7]	ADRH23[6]	ADRH23[5]	ADRH23[4]	ADRH23[3]	ADRH23[2]	ADRH23[1]	ADRH23[0]
	TSU_ADRH24	ADRH24[31]	ADRH24[30]	ADRH24[29]	ADRH24[28]	ADRH24[27]	ADRH24[26]	ADRH24[25]	ADRH24[24]
		ADRH24[23]	ADRH24[22]	ADRH24[21]	ADRH24[20]	ADRH24[19]	ADRH24[18]	ADRH24[17]	ADRH24[16]
		ADRH24[15]	ADRH24[14]	ADRH24[13]	ADRH24[12]	ADRH24[11]	ADRH24[10]	ADRH24[9]	ADRH24[8]
		ADRH24[7]	ADRH24[6]	ADRH24[5]	ADRH24[4]	ADRH24[3]	ADRH24[2]	ADRH24[1]	ADRH24[0]
	TSU_ADRH25	ADRH25[31]	ADRH25[30]	ADRH25[29]	ADRH25[28]	ADRH25[27]	ADRH25[26]	ADRH25[25]	ADRH25[24]
		ADRH25[23]	ADRH25[22]	ADRH25[21]	ADRH25[20]	ADRH25[19]	ADRH25[18]	ADRH25[17]	ADRH25[16]
		ADRH25[15]	ADRH25[14]	ADRH25[13]	ADRH25[12]	ADRH25[11]	ADRH25[10]	ADRH25[9]	ADRH25[8]
		ADRH25[7]	ADRH25[6]	ADRH25[5]	ADRH25[4]	ADRH25[3]	ADRH25[2]	ADRH25[1]	ADRH25[0]
	TSU_ADRH26	ADRH26[31]	ADRH26[30]	ADRH26[29]	ADRH26[28]	ADRH26[27]	ADRH26[26]	ADRH26[25]	ADRH26[24]
		ADRH26[23]	ADRH26[22]	ADRH26[21]	ADRH26[20]	ADRH26[19]	ADRH26[18]	ADRH26[17]	ADRH26[16]
		ADRH26[15]	ADRH26[14]	ADRH26[13]	ADRH26[12]	ADRH26[11]	ADRH26[10]	ADRH26[9]	ADRH26[8]
		ADRH26[7]	ADRH26[6]	ADRH26[5]	ADRH26[4]	ADRH26[3]	ADRH26[2]	ADRH26[1]	ADRH26[0]
	TSU_ADRH27	ADRH27[31]	ADRH27[30]	ADRH27[29]	ADRH27[28]	ADRH27[27]	ADRH27[26]	ADRH27[25]	ADRH27[24]
		ADRH27[23]	ADRH27[22]	ADRH27[21]	ADRH27[20]	ADRH27[19]	ADRH27[18]	ADRH27[17]	ADRH27[16]
		ADRH27[15]	ADRH27[14]	ADRH27[13]	ADRH27[12]	ADRH27[11]	ADRH27[10]	ADRH27[9]	ADRH27[8]
		ADRH27[7]	ADRH27[6]	ADRH27[5]	ADRH27[4]	ADRH27[3]	ADRH27[2]	ADRH27[1]	ADRH27[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Ethernet controller	TSU_ADRH28	ADRH28[31]	ADRH28[30]	ADRH28[29]	ADRH28[28]	ADRH28[27]	ADRH28[26]	ADRH28[25]	ADRH28[24]
		ADRH28[23]	ADRH28[22]	ADRH28[21]	ADRH28[20]	ADRH28[19]	ADRH28[18]	ADRH28[17]	ADRH28[16]
		ADRH28[15]	ADRH28[14]	ADRH28[13]	ADRH28[12]	ADRH28[11]	ADRH28[10]	ADRH28[9]	ADRH28[8]
		ADRH28[7]	ADRH28[6]	ADRH28[5]	ADRH28[4]	ADRH28[3]	ADRH28[2]	ADRH28[1]	ADRH28[0]
	TSU_ADRH29	ADRH29[31]	ADRH29[30]	ADRH29[29]	ADRH29[28]	ADRH29[27]	ADRH29[26]	ADRH29[25]	ADRH29[24]
		ADRH29[23]	ADRH29[22]	ADRH29[21]	ADRH29[20]	ADRH29[19]	ADRH29[18]	ADRH29[17]	ADRH29[16]
		ADRH29[15]	ADRH29[14]	ADRH29[13]	ADRH29[12]	ADRH29[11]	ADRH29[10]	ADRH29[9]	ADRH29[8]
		ADRH29[7]	ADRH29[6]	ADRH29[5]	ADRH29[4]	ADRH29[3]	ADRH29[2]	ADRH29[1]	ADRH29[0]
	TSU_ADRH30	ADRH30[31]	ADRH30[30]	ADRH30[29]	ADRH30[28]	ADRH30[27]	ADRH30[26]	ADRH30[25]	ADRH30[24]
		ADRH30[23]	ADRH30[22]	ADRH30[21]	ADRH30[20]	ADRH30[19]	ADRH30[18]	ADRH30[17]	ADRH30[16]
		ADRH30[15]	ADRH30[14]	ADRH30[13]	ADRH30[12]	ADRH30[11]	ADRH30[10]	ADRH30[9]	ADRH30[8]
		ADRH30[7]	ADRH30[6]	ADRH30[5]	ADRH30[4]	ADRH30[3]	ADRH30[2]	ADRH30[1]	ADRH30[0]
	TSU_ADRH31	ADRH31[31]	ADRH31[30]	ADRH31[29]	ADRH31[28]	ADRH31[27]	ADRH31[26]	ADRH31[25]	ADRH31[24]
		ADRH31[23]	ADRH31[22]	ADRH31[21]	ADRH31[20]	ADRH31[19]	ADRH31[18]	ADRH31[17]	ADRH31[16]
		ADRH31[15]	ADRH31[14]	ADRH31[13]	ADRH31[12]	ADRH31[11]	ADRH31[10]	ADRH31[9]	ADRH31[8]
		ADRH31[7]	ADRH31[6]	ADRH31[5]	ADRH31[4]	ADRH31[3]	ADRH31[2]	ADRH31[1]	ADRH31[0]
	TSU_ADRL0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL0[15]	ADRL0[14]	ADRL0[13]	ADRL0[12]	ADRL0[11]	ADRL0[10]	ADRL0[9]	ADRL0[8]
		ADRL0[7]	ADRL0[6]	ADRL0[5]	ADRL0[4]	ADRL0[3]	ADRL0[2]	ADRL0[1]	ADRL0[0]
	TSU_ADRL1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL1[15]	ADRL1[14]	ADRL1[13]	ADRL1[12]	ADRL1[11]	ADRL1[10]	ADRL1[9]	ADRL1[8]
		ADRL1[7]	ADRL1[6]	ADRL1[5]	ADRL1[4]	ADRL1[3]	ADRL1[2]	ADRL1[1]	ADRL1[0]
	TSU_ADRL2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL2[15]	ADRL2[14]	ADRL2[13]	ADRL2[12]	ADRL2[11]	ADRL2[10]	ADRL2[9]	ADRL2[8]
		ADRL2[7]	ADRL2[6]	ADRL2[5]	ADRL2[4]	ADRL2[3]	ADRL2[2]	ADRL2[1]	ADRL2[0]
	TSU_ADRL3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL3[15]	ADRL3[14]	ADRL3[13]	ADRL3[12]	ADRL3[11]	ADRL3[10]	ADRL3[9]	ADRL3[8]
		ADRL3[7]	ADRL3[6]	ADRL3[5]	ADRL3[4]	ADRL3[3]	ADRL3[2]	ADRL3[1]	ADRL3[0]
	TSU_ADRL4	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL4[15]	ADRL4[14]	ADRL4[13]	ADRL4[12]	ADRL4[11]	ADRL4[10]	ADRL4[9]	ADRL4[8]
		ADRL4[7]	ADRL4[6]	ADRL4[5]	ADRL4[4]	ADRL4[3]	ADRL4[2]	ADRL4[1]	ADRL4[0]
	TSU_ADRL5	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL5[15]	ADRL5[14]	ADRL5[13]	ADRL5[12]	ADRL5[11]	ADRL5[10]	ADRL5[9]	ADRL5[8]
		ADRL5[7]	ADRL5[6]	ADRL5[5]	ADRL5[4]	ADRL5[3]	ADRL5[2]	ADRL5[1]	ADRL5[0]
	TSU_ADRL6	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL6[15]	ADRL6[14]	ADRL6[13]	ADRL6[12]	ADRL6[11]	ADRL6[10]	ADRL6[9]	ADRL6[8]
		ADRL6[7]	ADRL6[6]	ADRL6[5]	ADRL6[4]	ADRL6[3]	ADRL6[2]	ADRL6[1]	ADRL6[0]
	TSU_ADRL7	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL7[15]	ADRL7[14]	ADRL7[13]	ADRL7[12]	ADRL7[11]	ADRL7[10]	ADRL7[9]	ADRL7[8]
		ADRL7[7]	ADRL7[6]	ADRL7[5]	ADRL7[4]	ADRL7[3]	ADRL7[2]	ADRL7[1]	ADRL7[0]
	TSU_ADRL8	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL8[15]	ADRL8[14]	ADRL8[13]	ADRL8[12]	ADRL8[11]	ADRL8[10]	ADRL8[9]	ADRL8[8]
		ADRL8[7]	ADRL8[6]	ADRL8[5]	ADRL8[4]	ADRL8[3]	ADRL8[2]	ADRL8[1]	ADRL8[0]
	TSU_ADRL9	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL9[15]	ADRL9[14]	ADRL9[13]	ADRL9[12]	ADRL9[11]	ADRL9[10]	ADRL9[9]	ADRL9[8]
		ADRL9[7]	ADRL9[6]	ADRL9[5]	ADRL9[4]	ADRL9[3]	ADRL9[2]	ADRL9[1]	ADRL9[0]
	TSU_ADRL10	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL10[15]	ADRL10[14]	ADRL10[13]	ADRL10[12]	ADRL10[11]	ADRL10[10]	ADRL10[9]	ADRL10[8]
		ADRL10[7]	ADRL10[6]	ADRL10[5]	ADRL10[4]	ADRL10[3]	ADRL10[2]	ADRL10[1]	ADRL10[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Ethernet controller	TSU_ADRL11	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL11[15]	ADRL11[14]	ADRL11[13]	ADRL11[12]	ADRL11[11]	ADRL11[10]	ADRL11[9]	ADRL11[8]
		ADRL11[7]	ADRL11[6]	ADRL11[5]	ADRL11[4]	ADRL11[3]	ADRL11[2]	ADRL11[1]	ADRL11[0]
	TSU_ADRL12	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL12[15]	ADRL12[14]	ADRL12[13]	ADRL12[12]	ADRL12[11]	ADRL12[10]	ADRL12[9]	ADRL12[8]
		ADRL12[7]	ADRL12[6]	ADRL12[5]	ADRL12[4]	ADRL12[3]	ADRL12[2]	ADRL12[1]	ADRL12[0]
	TSU_ADRL13	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL13[15]	ADRL13[14]	ADRL13[13]	ADRL13[12]	ADRL13[11]	ADRL13[10]	ADRL13[9]	ADRL13[8]
		ADRL13[7]	ADRL13[6]	ADRL13[5]	ADRL13[4]	ADRL13[3]	ADRL13[2]	ADRL13[1]	ADRL13[0]
	TSU_ADRL14	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL14[15]	ADRL14[14]	ADRL14[13]	ADRL14[12]	ADRL14[11]	ADRL14[10]	ADRL14[9]	ADRL14[8]
		ADRL14[7]	ADRL14[6]	ADRL14[5]	ADRL14[4]	ADRL14[3]	ADRL14[2]	ADRL14[1]	ADRL14[0]
	TSU_ADRL15	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL15[15]	ADRL15[14]	ADRL15[13]	ADRL15[12]	ADRL15[11]	ADRL15[10]	ADRL15[9]	ADRL15[8]
		ADRL15[7]	ADRL15[6]	ADRL15[5]	ADRL15[4]	ADRL15[3]	ADRL15[2]	ADRL15[1]	ADRL15[0]
	TSU_ADRL16	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL16[15]	ADRL16[14]	ADRL16[13]	ADRL16[12]	ADRL16[11]	ADRL16[10]	ADRL16[9]	ADRL16[8]
		ADRL16[7]	ADRL16[6]	ADRL16[5]	ADRL16[4]	ADRL16[3]	ADRL16[2]	ADRL16[1]	ADRL16[0]
	TSU_ADRL17	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL17[15]	ADRL17[14]	ADRL17[13]	ADRL17[12]	ADRL17[11]	ADRL17[10]	ADRL17[9]	ADRL17[8]
		ADRL17[7]	ADRL17[6]	ADRL17[5]	ADRL17[4]	ADRL17[3]	ADRL17[2]	ADRL17[1]	ADRL17[0]
	TSU_ADRL18	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL18[15]	ADRL18[14]	ADRL18[13]	ADRL18[12]	ADRL18[11]	ADRL18[10]	ADRL18[9]	ADRL18[8]
		ADRL18[7]	ADRL18[6]	ADRL18[5]	ADRL18[4]	ADRL18[3]	ADRL18[2]	ADRL18[1]	ADRL18[0]
	TSU_ADRL19	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL19[15]	ADRL19[14]	ADRL19[13]	ADRL19[12]	ADRL19[11]	ADRL19[10]	ADRL19[9]	ADRL19[8]
		ADRL19[7]	ADRL19[6]	ADRL19[5]	ADRL19[4]	ADRL19[3]	ADRL19[2]	ADRL19[1]	ADRL19[0]
	TSU_ADRL20	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL20[15]	ADRL20[14]	ADRL20[13]	ADRL20[12]	ADRL20[11]	ADRL20[10]	ADRL20[9]	ADRL20[8]
		ADRL20[7]	ADRL20[6]	ADRL20[5]	ADRL20[4]	ADRL20[3]	ADRL20[2]	ADRL20[1]	ADRL20[0]
	TSU_ADRL21	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL21[15]	ADRL21[14]	ADRL21[13]	ADRL21[12]	ADRL21[11]	ADRL21[10]	ADRL21[9]	ADRL21[8]
		ADRL21[7]	ADRL21[6]	ADRL21[5]	ADRL21[4]	ADRL21[3]	ADRL21[2]	ADRL21[1]	ADRL21[0]
	TSU_ADRL22	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL22[15]	ADRL22[14]	ADRL22[13]	ADRL22[12]	ADRL22[11]	ADRL22[10]	ADRL22[9]	ADRL22[8]
		ADRL22[7]	ADRL22[6]	ADRL22[5]	ADRL22[4]	ADRL22[3]	ADRL22[2]	ADRL22[1]	ADRL22[0]
	TSU_ADRL23	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL23[15]	ADRL23[14]	ADRL23[13]	ADRL23[12]	ADRL23[11]	ADRL23[10]	ADRL23[9]	ADRL23[8]
		ADRL23[7]	ADRL23[6]	ADRL23[5]	ADRL23[4]	ADRL23[3]	ADRL23[2]	ADRL23[1]	ADRL23[0]
	TSU_ADRL24	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL24[15]	ADRL24[14]	ADRL24[13]	ADRL24[12]	ADRL24[11]	ADRL24[10]	ADRL24[9]	ADRL24[8]
		ADRL24[7]	ADRL24[6]	ADRL24[5]	ADRL24[4]	ADRL24[3]	ADRL24[2]	ADRL24[1]	ADRL24[0]
	TSU_ADRL25	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL25[15]	ADRL25[14]	ADRL25[13]	ADRL25[12]	ADRL25[11]	ADRL25[10]	ADRL25[9]	ADRL25[8]
		ADRL25[7]	ADRL25[6]	ADRL25[5]	ADRL25[4]	ADRL25[3]	ADRL25[2]	ADRL25[1]	ADRL25[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Ethernet controller	TSU_ADRL26	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL26[15]	ADRL26[14]	ADRL26[13]	ADRL26[12]	ADRL26[11]	ADRL26[10]	ADRL26[9]	ADRL26[8]
		ADRL26[7]	ADRL26[6]	ADRL26[5]	ADRL26[4]	ADRL26[3]	ADRL26[2]	ADRL26[1]	ADRL26[0]
	TSU_ADRL27	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL27[15]	ADRL27[14]	ADRL27[13]	ADRL27[12]	ADRL27[11]	ADRL27[10]	ADRL27[9]	ADRL27[8]
		ADRL27[7]	ADRL27[6]	ADRL27[5]	ADRL27[4]	ADRL27[3]	ADRL27[2]	ADRL27[1]	ADRL27[0]
	TSU_ADRL28	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL28[15]	ADRL28[14]	ADRL28[13]	ADRL28[12]	ADRL28[11]	ADRL28[10]	ADRL28[9]	ADRL28[8]
		ADRL28[7]	ADRL28[6]	ADRL28[5]	ADRL28[4]	ADRL28[3]	ADRL28[2]	ADRL28[1]	ADRL28[0]
	TSU_ADRL29	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL29[15]	ADRL29[14]	ADRL29[13]	ADRL29[12]	ADRL29[11]	ADRL29[10]	ADRL29[9]	ADRL29[8]
		ADRL29[7]	ADRL29[6]	ADRL29[5]	ADRL29[4]	ADRL29[3]	ADRL29[2]	ADRL29[1]	ADRL29[0]
	TSU_ADRL30	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL30[15]	ADRL30[14]	ADRL30[13]	ADRL30[12]	ADRL30[11]	ADRL30[10]	ADRL30[9]	ADRL30[8]
		ADRL30[7]	ADRL30[6]	ADRL30[5]	ADRL30[4]	ADRL30[3]	ADRL30[2]	ADRL30[1]	ADRL30[0]
	TSU_ADRL31	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ADRL31[15]	ADRL31[14]	ADRL31[13]	ADRL31[12]	ADRL31[11]	ADRL31[10]	ADRL31[9]	ADRL31[8]
		ADRL31[7]	ADRL31[6]	ADRL31[5]	ADRL31[4]	ADRL31[3]	ADRL31[2]	ADRL31[1]	ADRL31[0]
	TXNLCR0	NTC0[31]	NTC0[30]	NTC0[29]	NTC0[28]	NTC0[27]	NTC0[26]	NTC0[25]	NTC0[24]
		NTC0[23]	NTC0[22]	NTC0[21]	NTC0[20]	NTC0[19]	NTC0[18]	NTC0[17]	NTC0[16]
		NTC0[15]	NTC0[14]	NTC0[13]	NTC0[12]	NTC0[11]	NTC0[10]	NTC0[9]	NTC0[8]
		NTC0[7]	NTC0[6]	NTC0[5]	NTC0[4]	NTC0[3]	NTC0[2]	NTC0[1]	NTC0[0]
	TXALCR0	TC0[31]	TC0[30]	TC0[29]	TC0[28]	TC0[27]	TC0[26]	TC0[25]	TC0[24]
		TC0[23]	TC0[22]	TC0[21]	TC0[20]	TC0[19]	TC0[18]	TC0[17]	TC0[16]
		TC0[15]	TC0[14]	TC0[13]	TC0[12]	TC0[11]	TC0[10]	TC0[9]	TC0[8]
		TC0[7]	TC0[6]	TC0[5]	TC0[4]	TC0[3]	TC0[2]	TC0[1]	TC0[0]
	RXNLCR0	NRC0[31]	NRC0[30]	NRC0[29]	NRC0[28]	NRC0[27]	NRC0[26]	NRC0[25]	NRC0[24]
		NRC0[23]	NRC0[22]	NRC0[21]	NRC0[20]	NRC0[19]	NRC0[18]	NRC0[17]	NRC0[16]
		NRC0[15]	NRC0[14]	NRC0[13]	NRC0[12]	NRC0[11]	NRC0[10]	NRC0[9]	NRC0[8]
		NRC0[7]	NRC0[6]	NRC0[5]	NRC0[4]	NRC0[3]	NRC0[2]	NRC0[1]	NRC0[0]
	RXALCR0	RC0[31]	RC0[30]	RC0[29]	RC0[28]	RC0[27]	RC0[26]	RC0[25]	RC0[24]
		RC0[23]	RC0[22]	RC0[21]	RC0[20]	RC0[19]	RC0[18]	RC0[17]	RC0[16]
		RC0[15]	RC0[14]	RC0[13]	RC0[12]	RC0[11]	RC0[10]	RC0[9]	RC0[8]
		RC0[7]	RC0[6]	RC0[5]	RC0[4]	RC0[3]	RC0[2]	RC0[1]	RC0[0]
	EDSR0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	ENT	ENR
	EDMR0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	DE	DL[1]	DL[0]	-	-	SWRT	SWRR
	EDTRR0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	TR[1]	TR[0]
	EDRRR0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	RR
	EESR0	TWB[1]	TWB[0]	TC[1]	TUC	ROC	TABT	RABT	RFCOF
		-	ECI	TC[0]	TDE	TFUF	FR	RDE	RFOF
		-	-	-	-	-	-	-	-
		RMAF	-	-	RRF	RTLF	RTSF	PRE	CERF

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Ethernet controller	EESIPR0	TWB1IP	TWB0IP	TC1IP	TUCIP	ROCIP	TABTIP	RABTIP	RFCOFIP
		-	ECIIP	TC0IP	TDEIP	TFUFIP	FRIP	RDEIP	RFOFIP
		-	-	-	-	-	-	-	-
		RMAFIP	-	-	RRFIP	RTLFIP	RTSFIP	PREIP	CERFIP
	TDLAR0	TDLA[31]	TDLA[30]	TDLA[29]	TDLA[28]	TDLA[27]	TDLA[26]	TDLA[25]	TDLA[24]
		TDLA[23]	TDLA[22]	TDLA[21]	TDLA[20]	TDLA[19]	TDLA[18]	TDLA[17]	TDLA[16]
		TDLA[15]	TDLA[14]	TDLA[13]	TDLA[12]	TDLA[11]	TDLA[10]	TDLA[9]	TDLA[8]
		TDLA[7]	TDLA[6]	TDLA[5]	TDLA[4]	TDLA[3]	TDLA[2]	TDLA[1]	TDLA[0]
	TDFAR0	TDFA[31]	TDFA[30]	TDFA[29]	TDFA[28]	TDFA[27]	TDFA[26]	TDFA[25]	TDFA[24]
		TDFA[23]	TDFA[22]	TDFA[21]	TDFA[20]	TDFA[19]	TDFA[18]	TDFA[17]	TDFA[16]
		TDFA[15]	TDFA[14]	TDFA[13]	TDFA[12]	TDFA[11]	TDFA[10]	TDFA[9]	TDFA[8]
		TDFA[7]	TDFA[6]	TDFA[5]	TDFA[4]	TDFA[3]	TDFA[2]	TDFA[1]	TDFA[0]
	TDFXR0	TDFX[31]	TDFX[30]	TDFX[29]	TDFX[28]	TDFX[27]	TDFX[26]	TDFX[25]	TDFX[24]
		TDFX[23]	TDFX[22]	TDFX[21]	TDFX[20]	TDFX[19]	TDFX[18]	TDFX[17]	TDFX[16]
		TDFX[15]	TDFX[14]	TDFX[13]	TDFX[12]	TDFX[11]	TDFX[10]	TDFX[9]	TDFX[8]
		TDFX[7]	TDFX[6]	TDFX[5]	TDFX[4]	TDFX[3]	TDFX[2]	TDFX[1]	TDFX[0]
	TDFFR0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	TDLF
	RDLAR0	RDLA[31]	RDLA[30]	RDLA[29]	RDLA[28]	RDLA[27]	RDLA[26]	RDLA[25]	RDLA[24]
		RDLA[23]	RDLA[22]	RDLA[21]	RDLA[20]	RDLA[19]	RDLA[18]	RDLA[17]	RDLA[16]
		RDLA[15]	RDLA[14]	RDLA[13]	RDLA[12]	RDLA[11]	RDLA[10]	RDLA[9]	RDLA[8]
		RDLA[7]	RDLA[6]	RDLA[5]	RDLA[4]	RDLA[3]	RDLA[2]	RDLA[1]	RDLA[0]
	RDFAR0	RDFA[31]	RDFA[30]	RDFA[29]	RDFA[28]	RDFA[27]	RDFA[26]	RDFA[25]	RDFA[24]
		RDFA[23]	RDFA[22]	RDFA[21]	RDFA[20]	RDFA[19]	RDFA[18]	RDFA[17]	RDFA[16]
		RDFA[15]	RDFA[14]	RDFA[13]	RDFA[12]	RDFA[11]	RDFA[10]	RDFA[9]	RDFA[8]
		RDFA[7]	RDFA[6]	RDFA[5]	RDFA[4]	RDFA[3]	RDFA[2]	RDFA[1]	RDFA[0]
	RDFXR0	RDFX[31]	RDFX[30]	RDFX[29]	RDFX[28]	RDFX[27]	RDFX[26]	RDFX[25]	RDFX[24]
		RDFX[23]	RDFX[22]	RDFX[21]	RDFX[20]	RDFX[19]	RDFX[18]	RDFX[17]	RDFX[16]
		RDFX[15]	RDFX[14]	RDFX[13]	RDFX[12]	RDFX[11]	RDFX[10]	RDFX[9]	RDFX[8]
		RDFX[7]	RDFX[6]	RDFX[5]	RDFX[4]	RDFX[3]	RDFX[2]	RDFX[1]	RDFX[0]
	RDFFR0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	RDLF
	TRSCER0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	TABTCE	RABTCE
		-	-	-	-	-	-	-	-
		RMAFCE	-	-	RRFCE	RTLFCE	RTSFCE	PRECE	CERFCE
	RMFCR0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		MFC[15]	MFC[14]	MFC[13]	MFC[12]	MFC[11]	MFC[10]	MFC[9]	MFC[8]
		MFC[7]	MFC[6]	MFC[5]	MFC[4]	MFC[3]	MFC[2]	MFC[1]	MFC[0]
	TFTR0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	TFT[10]	TFT[9]	TFT[8]
		TFT[7]	TFT[6]	TFT[5]	TFT[4]	TFT[3]	TFT[2]	TFT[1]	TFT[0]
	FDR0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	TFD[2]	TFD[1]	TFD[0]
		-	-	-	RFD[4]	RFD[3]	RFD[2]	RFD[1]	RFD[0]
	RMCR0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	RNC
	RPADIR0	-	-	-	-	-	-	-	-
		-	-	-	PADS[4]	PADS[3]	PADS[2]	PADS[1]	PADS[0]
		PADR[15]	PADR[14]	PADR[13]	PADR[12]	PADR[11]	PADR[10]	PADR[9]	PADR[8]
		PADR[7]	PADR[6]	PADR[5]	PADR[4]	PADR[3]	PADR[2]	PADR[1]	PADR[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Ethernet controller	FCFTR0	-	-	-	-	-	-	-	-
		-	-	-	RFF[4]	RFF[3]	RFF[2]	RFF[1]	RFF[0]
		-	-	-	-	-	-	-	-
		RFD[7]	RFD[6]	RFD[5]	RFD[4]	RFD[3]	RFD[2]	RFD[1]	RFD[0]
	CSMR	CSEBL	CSMD	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	SB[5]	SB[4]	SB[3]	SB[2]	SB[1]	SB[0]
	CSSBM	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	SBM[5]	SBM[4]	SBM[3]	SBM[2]	SBM[1]	SBM[0]
	CSSMR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		CS[15]	CS[14]	CS[13]	CS[12]	CS[11]	CS[10]	CS[9]	CS[8]
		CS[7]	CS[6]	CS[5]	CS[4]	CS[3]	CS[2]	CS[1]	CS[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
A/D converter	ADDRA	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	ADDRB	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	ADDRC	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	ADDRD	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	ADDRE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	ADDRF	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	ADDRG	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	ADDRH	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	ADCMPHA	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	ADCMPLA	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	ADCMPHB	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	ADCMPLB	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	ADCMPHC	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	ADCMPLC	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	ADCMPHD	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	ADCMPLD	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	ADCMPHE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	ADCMPLE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	ADCMPHF	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	ADCMPLF	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	ADCMPHG	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	ADCMPLG	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	ADCMPHH	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	ADCMPLH	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	ADCSR	ADF	ADIE	ADST	TRGS[3]	TRGS[2]	TRGS[1]	TRGS[0]	CKS[2]
		CKS[1]	CKS[0]	MDS[2]	MDS[1]	MDS[0]	CH[2]	CH[1]	CH[0]
	ADCMPER	HLMENH	HLMENG	HLMENF	HLMENE	HLMEND	HLMENC	HLMENB	HLMENA
		LLMENH	LLMENG	LLMENF	LLMENE	LLMEND	LLMENC	LLMENB	LLMENA
	ADCMPSR	HOVRH	HOVRG	HOVRF	HOVRE	HOVRD	HOVRC	HOVRB	HOVRA
		LUDRH	LUDRG	LUDRF	LUDRE	LUDRD	LUDRC	LUDRB	LUDRA

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
USB2.0 host/ function module	SYSCFG0_0	-	-	-	-	-	SCKE	-	-
		HSE	DCFM	DRPD	DPRPU	-	UCKSEL	UPLLE	USBE
	BUSWAIT_0	-	-	-	-	-	-	-	-
		-	-	BWAIT[5]	BWAIT[4]	BWAIT[3]	BWAIT[2]	BWAIT[1]	BWAIT[0]
	SYSSTS0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	LNST[1]	LNST[0]
	DVSTCTR0_0	-	-	-	-	-	-	-	WKUP
		RWUPE	USBRST	RESUME	UACT	-	RHST[2]	RHST[1]	RHST[0]
	TESTMODE_0	-	-	-	-	-	-	-	-
		-	-	-	-	UTST[3]	UTST[2]	UTST[1]	UTST[0]
	D0FBCFG_0	-	-	DFACC[1]	DFACC[0]	-	-	-	-
		-	-	-	TENDE	-	-	-	-
	D1FBCFG_0	-	-	DFACC[1]	DFACC[0]	-	-	-	-
		-	-	-	TENDE	-	-	-	-
	CFIFO_0	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D0FIFO_0	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D1FIFO_0	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	CFIFOSEL_0	RCNT	REW	-	-	MBW[1]	MBW[0]	-	BIGEND
		-	-	ISEL	-	CURPIPE[3]	CURPIPE[2]	CURPIPE[1]	CURPIPE[0]
	CFIFOCTR_0	BVAL	BCLR	FRDY	-	DTLN[11]	DTLN[10]	DTLN[9]	DTLN[8]
		DTLN[7]	DTLN[6]	DTLN[5]	DTLN[4]	DTLN[3]	DTLN[2]	DTLN[1]	DTLN[0]
	D0FIFOSEL_0	RCNT	REW	DCLRM	DREQE	MBW[1]	MBW[0]	-	BIGEND
		-	-	-	-	CURPIPE[3]	CURPIPE[2]	CURPIPE[1]	CURPIPE[0]
	D0FIFOCTR_0	BVAL	BCLR	FRDY	-	DTLN[11]	DTLN[10]	DTLN[9]	DTLN[8]
		DTLN[7]	DTLN[6]	DTLN[5]	DTLN[4]	DTLN[3]	DTLN[2]	DTLN[1]	DTLN[0]
	D1FIFOSEL_0	RCNT	REW	DCLRM	DREQE	MBW[1]	MBW[0]	-	BIGEND
		-	-	-	-	CURPIPE[3]	CURPIPE[2]	CURPIPE[1]	CURPIPE[0]
	D1FIFOCTR_0	BVAL	BCLR	FRDY	-	DTLN[11]	DTLN[10]	DTLN[9]	DTLN[8]
		DTLN[7]	DTLN[6]	DTLN[5]	DTLN[4]	DTLN[3]	DTLN[2]	DTLN[1]	DTLN[0]
	INTENB0_0	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE
		-	-	-	-	-	-	-	-
	INTENB1_0	-	BCHGE	-	DTCHE	ATTCHE	-	-	-
		-	EOFERRE	SIGNE	SACKE	-	-	-	-
	BRDYENB_0	PIPEBRDYE[15]	PIPEBRDYE[14]	PIPEBRDYE[13]	PIPEBRDYE[12]	PIPEBRDYE[11]	PIPEBRDYE[10]	PIPEBRDYE[9]	PIPEBRDYE[8]
		PIPEBRDYE[7]	PIPEBRDYE[6]	PIPEBRDYE[5]	PIPEBRDYE[4]	PIPEBRDYE[3]	PIPEBRDYE[2]	PIPEBRDYE[1]	PIPEBRDYE[0]
	NRDYENB_0	PIPENRDYE[15]	PIPENRDYE[14]	PIPENRDYE[13]	PIPENRDYE[12]	PIPENRDYE[11]	PIPENRDYE[10]	PIPENRDYE[9]	PIPENRDYE[8]
		PIPENRDYE[7]	PIPENRDYE[6]	PIPENRDYE[5]	PIPENRDYE[4]	PIPENRDYE[3]	PIPENRDYE[2]	PIPENRDYE[1]	PIPENRDYE[0]
	BEMPENB_0	PIPEBEMPE[15]	PIPEBEMPE[14]	PIPEBEMPE[13]	PIPEBEMPE[12]	PIPEBEMPE[11]	PIPEBEMPE[10]	PIPEBEMPE[9]	PIPEBEMPE[8]
		PIPEBEMPE[7]	PIPEBEMPE[6]	PIPEBEMPE[5]	PIPEBEMPE[4]	PIPEBEMPE[3]	PIPEBEMPE[2]	PIPEBEMPE[1]	PIPEBEMPE[0]
	SOFCFG_0	-	-	-	-	-	-	-	TRNENSEL
		-	BRDYM	-	-	-	-	-	-
	INTSTS0_0	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY
		VBSTS	DVSQ[2]	DVSQ[1]	DVSQ[0]	VALID	CTSQ[2]	CTSQ[1]	CTSQ[0]
	INTSTS1_0	-	BCHG	-	DTCH	ATTCH	-	-	-
		-	EOFERR	SIGN	SACK	-	-	-	-
	BRDYSTS_0	PIPEBRDY[15]	PIPEBRDY[14]	PIPEBRDY[13]	PIPEBRDY[12]	PIPEBRDY[11]	PIPEBRDY[10]	PIPEBRDY[9]	PIPEBRDY[8]
		PIPEBRDY[7]	PIPEBRDY[6]	PIPEBRDY[5]	PIPEBRDY[4]	PIPEBRDY[3]	PIPEBRDY[2]	PIPEBRDY[1]	PIPEBRDY[0]
	NRDYSTS_0	PIPENRDY[15]	PIPENRDY[14]	PIPENRDY[13]	PIPENRDY[12]	PIPENRDY[11]	PIPENRDY[10]	PIPENRDY[9]	PIPENRDY[8]
		PIPENRDY[7]	PIPENRDY[6]	PIPENRDY[5]	PIPENRDY[4]	PIPENRDY[3]	PIPENRDY[2]	PIPENRDY[1]	PIPENRDY[0]
	BEMPSTS_0	PIPEBEMP[15]	PIPEBEMP[14]	PIPEBEMP[13]	PIPEBEMP[12]	PIPEBEMP[11]	PIPEBEMP[10]	PIPEBEMP[9]	PIPEBEMP[8]
		PIPEBEMP[7]	PIPEBEMP[6]	PIPEBEMP[5]	PIPEBEMP[4]	PIPEBEMP[3]	PIPEBEMP[2]	PIPEBEMP[1]	PIPEBEMP[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
USB2.0 host/ function module	FRMNUM_0	OVRN	CRCE	-	-	-	FRNM[10]	FRNM[9]	FRNM[8]
		FRNM[7]	FRNM[6]	FRNM[5]	FRNM[4]	FRNM[3]	FRNM[2]	FRNM[1]	FRNM[0]
	UFRMNUM_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	UFRNM[2]	UFRNM[1]	UFRNM[0]
	USBADDR_0	-	-	-	-	-	-	-	-
		-	USBADDR[6]	USBADDR[5]	USBADDR[4]	USBADDR[3]	USBADDR[2]	USBADDR[1]	USBADDR[0]
	USBREQ_0	bRequest[7]	bRequest[6]	bRequest[5]	bRequest[4]	bRequest[3]	bRequest[2]	bRequest[1]	bRequest[0]
		bmRequestType[7]	bmRequestType[6]	bmRequestType[5]	bmRequestType[4]	bmRequestType[3]	bmRequestType[2]	bmRequestType[1]	bmRequestType[0]
	USBVAL_0	wValue[15]	wValue[14]	wValue[13]	wValue[12]	wValue[11]	wValue[10]	wValue[9]	wValue[8]
		wValue[7]	wValue[6]	wValue[5]	wValue[4]	wValue[3]	wValue[2]	wValue[1]	wValue[0]
	USBINDX_0	wIndex[15]	wIndex[14]	wIndex[13]	wIndex[12]	wIndex[11]	wIndex[10]	wIndex[9]	wIndex[8]
		wIndex[7]	wIndex[6]	wIndex[5]	wIndex[4]	wIndex[3]	wIndex[2]	wIndex[1]	wIndex[0]
	USBLENG_0	wLength[15]	wLength[14]	wLength[13]	wLength[12]	wLength[11]	wLength[10]	wLength[9]	wLength[8]
		wLength[7]	wLength[6]	wLength[5]	wLength[4]	wLength[3]	wLength[2]	wLength[1]	wLength[0]
	DCPCFG_0	-	-	-	-	-	-	-	CNTMD
		SHTNAK	-	-	DIR	-	-	-	-
	DCPMAXP_0	DEVSEL[3]	DEVSEL[2]	DEVSEL[1]	DEVSEL[0]	-	-	-	-
		-	MXPS[6]	MXPS[5]	MXPS[4]	MXPS[3]	MXPS[2]	MXPS[1]	MXPS[0]
	DCPCTR_0	BSTS	SUREQ	CSCLR	CSSTS	SUREQCLR	-	-	SQCLR
		SQSET	SQMON	PBUSY	PINGE	-	CCPL	PID[1]	PID[0]
	PIPESEL_0	-	-	-	-	-	-	-	-
		-	-	-	-	PIPESEL[3]	PIPESEL[2]	PIPESEL[1]	PIPESEL[0]
	PIPECFG_0	TYPE[1]	TYPE[0]	-	-	-	BFRE	DBLB	CNTMD
		SHTNAK	-	-	DIR	EPNUM[3]	EPNUM[2]	EPNUM[1]	EPNUM[0]
	PIPEBUF_0	-	BUFSIZE[4]	BUFSIZE[3]	BUFSIZE[2]	BUFSIZE[1]	BUFSIZE[0]	-	-
		BUFNMB[7]	BUFNMB[6]	BUFNMB[5]	BUFNMB[4]	BUFNMB[3]	BUFNMB[2]	BUFNMB[1]	BUFNMB[0]
	PIPEMAXP_0	DEVSEL[3]	DEVSEL[2]	DEVSEL[1]	DEVSEL[0]	-	MXPS[10]	MXPS[9]	MXPS[8]
		MXPS[7]	MXPS[6]	MXPS[5]	MXPS[4]	MXPS[3]	MXPS[2]	MXPS[1]	MXPS[0]
	PIPEPERI_0	-	-	-	IFIS	-	-	-	-
		-	-	-	-	-	IITV[2]	IITV[1]	IITV[0]
	PIPE1CTR_0	BSTS	INBUFM	CSCLR	CSSTS	-	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPE2CTR_0	BSTS	INBUFM	CSCLR	CSSTS	-	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPE3CTR_0	BSTS	INBUFM	CSCLR	CSSTS	-	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPE4CTR_0	BSTS	INBUFM	CSCLR	CSSTS	-	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPE5CTR_0	BSTS	INBUFM	CSCLR	CSSTS	-	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPE6CTR_0	BSTS	-	CSCLR	CSSTS	-	-	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPE7CTR_0	BSTS	-	CSCLR	CSSTS	-	-	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPE8CTR_0	BSTS	-	CSCLR	CSSTS	-	-	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPE9CTR_0	BSTS	INBUFM	CSCLR	CSSTS	-	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPEACTR_0	BSTS	INBUFM	CSCLR	CSSTS	-	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPEBCTR_0	BSTS	INBUFM	CSCLR	CSSTS	-	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPECCTR_0	BSTS	INBUFM	CSCLR	CSSTS	-	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPEDCTR_0	BSTS	INBUFM	CSCLR	CSSTS	-	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPEECTR_0	BSTS	INBUFM	CSCLR	CSSTS	-	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPEFCTR_0	BSTS	INBUFM	CSCLR	CSSTS	-	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
USB2.0 host/ function module	PIPE1TRE_0	-	-	-	-	-	-	TRENB	TRCLR
		-	-	-	-	-	-	-	-
	PIPE1TRN_0	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	PIPE2TRE_0	-	-	-	-	-	-	TRENB	TRCLR
		-	-	-	-	-	-	-	-
	PIPE2TRN_0	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	PIPE3TRE_0	-	-	-	-	-	-	TRENB	TRCLR
		-	-	-	-	-	-	-	-
	PIPE3TRN_0	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	PIPE4TRE_0	-	-	-	-	-	-	TRENB	TRCLR
		-	-	-	-	-	-	-	-
	PIPE4TRN_0	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	PIPE5TRE_0	-	-	-	-	-	-	TRENB	TRCLR
		-	-	-	-	-	-	-	-
	PIPE5TRN_0	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	PIPEBTRE_0	-	-	-	-	-	-	TRENB	TRCLR
		-	-	-	-	-	-	-	-
	PIPEBTRN_0	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	PIPECTRE_0	-	-	-	-	-	-	TRENB	TRCLR
		-	-	-	-	-	-	-	-
	PIPECTRN_0	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	PIPEDTRE_0	-	-	-	-	-	-	TRENB	TRCLR
		-	-	-	-	-	-	-	-
	PIPEDTRN_0	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	PIPEETRE_0	-	-	-	-	-	-	TRENB	TRCLR
		-	-	-	-	-	-	-	-
	PIPEETRN_0	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	PIPEFTRE_0	-	-	-	-	-	-	TRENB	TRCLR
		-	-	-	-	-	-	-	-
	PIPEFTRN_0	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	PIPE9TRE_0	-	-	-	-	-	-	TRENB	TRCLR
		-	-	-	-	-	-	-	-
	PIPE9TRN_0	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	PIPEATRE_0	-	-	-	-	-	-	TRENB	TRCLR
		-	-	-	-	-	-	-	-
	PIPEATRN_0	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	DEVADD0_0	-	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	-	-	-	-	-	-
	DEVADD1_0	-	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	-	-	-	-	-	-
	DEVADD2_0	-	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	-	-	-	-	-	-
	DEVADD3_0	-	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	-	-	-	-	-	-
	DEVADD4_0	-	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	-	-	-	-	-	-
	DEVADD5_0	-	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	-	-	-	-	-	-

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
USB2.0 host/ function module	DEVADD6_0	-	HPPHUB[3]	HPPHUB[2]	HPPHUB[1]	HPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	-	-	-
	DEVADD7_0	-	HPPHUB[3]	HPPHUB[2]	HPPHUB[1]	HPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	-	-	-	-	-	-
	DEVADD8_0	-	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	-	-	-	-	-	-
	DEVADD9_0	-	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	-	-	-	-	-	-
	DEVADDA_0	-	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	-	-	-	-	-	-
	SUSPMODE_0	-	SUSPM	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	D0FIFOB0_0	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D0FIFOB1_0	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D0FIFOB2_0	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D0FIFOB3_0	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D0FIFOB4_0	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D0FIFOB5_0	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D0FIFOB6_0	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D0FIFOB7_0	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D1FIFOB0_0	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D1FIFOB1_0	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D1FIFOB2_0	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D1FIFOB3_0	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
USB2.0 host/ function module	D1FIFOB4_0	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D1FIFOB5_0	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D1FIFOB6_0	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D1FIFOB7_0	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	SYSCFG0_1	-	-	-	-	-	SCKE	-	-
		HSE	DCFM	DRPD	DPRPU	-	-	-	USBE
	BUSWAIT_1	-	-	-	-	-	-	-	-
		-	-	BWAIT[5]	BWAIT[4]	BWAIT[3]	BWAIT[2]	BWAIT[1]	BWAIT[0]
	SYSSTS0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	LNST[1]	LNST[0]
	DVSTCTR0_1	-	-	-	-	-	-	-	WKUP
		RWUPE	USBRST	RESUME	UACT	-	RHST[2]	RHST[1]	RHST[0]
	TESTMODE_1	-	-	-	-	-	-	-	-
		-	-	-	-	UTST[3]	UTST[2]	UTST[1]	UTST[0]
	D0FBCFG_1	-	-	DFACC[1]	DFACC[0]	-	-	-	-
		-	-	-	TENDE	-	-	-	-
	D1FBCFG_1	-	-	DFACC[1]	DFACC[0]	-	-	-	-
		-	-	-	TENDE	-	-	-	-
	CFIFO_1	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D0FIFO_1	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D1FIFO_1	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	CFIFOSEL_1	RCNT	REW	-	-	MBW[1]	MBW[0]	-	BIGEND
		-	-	ISEL	-	CURPIPE[3]	CURPIPE[2]	CURPIPE[1]	CURPIPE[0]
	CFIFOCTR_1	BVAL	BCLR	FRDY	-	DTLN[11]	DTLN[10]	DTLN[9]	DTLN[8]
		DTLN[7]	DTLN[6]	DTLN[5]	DTLN[4]	DTLN[3]	DTLN[2]	DTLN[1]	DTLN[0]
	D0FIFOSEL_1	RCNT	REW	DCLRM	DREQE	MBW[1]	MBW[0]	-	BIGEND
		-	-	-	-	CURPIPE[3]	CURPIPE[2]	CURPIPE[1]	CURPIPE[0]
	D0FIFOCTR_1	BVAL	BCLR	FRDY	-	DTLN[11]	DTLN[10]	DTLN[9]	DTLN[8]
		DTLN[7]	DTLN[6]	DTLN[5]	DTLN[4]	DTLN[3]	DTLN[2]	DTLN[1]	DTLN[0]
	D1FIFOSEL_1	RCNT	REW	DCLRM	DREQE	MBW[1]	MBW[0]	-	BIGEND
		-	-	-	-	CURPIPE[3]	CURPIPE[2]	CURPIPE[1]	CURPIPE[0]
	D1FIFOCTR_1	BVAL	BCLR	FRDY	-	DTLN[11]	DTLN[10]	DTLN[9]	DTLN[8]
		DTLN[7]	DTLN[6]	DTLN[5]	DTLN[4]	DTLN[3]	DTLN[2]	DTLN[1]	DTLN[0]
	INTENB0_1	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE
		-	-	-	-	-	-	-	-
	INTENB1_1	-	BCHGE	-	DTCHE	ATTCHE	-	-	-
		-	EOFERRE	SIGNE	SACKE	-	-	-	-
	BRDYENB_1	PIPEBRDYE[15]	PIPEBRDYE[14]	PIPEBRDYE[13]	PIPEBRDYE[12]	PIPEBRDYE[11]	PIPEBRDYE[10]	PIPEBRDYE[9]	PIPEBRDYE[8]
		PIPEBRDYE[7]	PIPEBRDYE[6]	PIPEBRDYE[5]	PIPEBRDYE[4]	PIPEBRDYE[3]	PIPEBRDYE[2]	PIPEBRDYE[1]	PIPEBRDYE[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
USB2.0 host/ function module	NRDYENB_1	PIPENRDYE[15]	PIPENRDYE[14]	PIPENRDYE[13]	PIPENRDYE[12]	PIPENRDYE[11]	PIPENRDYE[10]	PIPENRDYE[9]	PIPENRDYE[8]
		PIPENRDYE[7]	PIPENRDYE[6]	PIPENRDYE[5]	PIPENRDYE[4]	PIPENRDYE[3]	PIPENRDYE[2]	PIPENRDYE[1]	PIPENRDYE[0]
	BEMPENB_1	PIPEBEMPE[15]	PIPEBEMPE[14]	PIPEBEMPE[13]	PIPEBEMPE[12]	PIPEBEMPE[11]	PIPEBEMPE[10]	PIPEBEMPE[9]	PIPEBEMPE[8]
		PIPEBEMPE[7]	PIPEBEMPE[6]	PIPEBEMPE[5]	PIPEBEMPE[4]	PIPEBEMPE[3]	PIPEBEMPE[2]	PIPEBEMPE[1]	PIPEBEMPE[0]
	SOFCFG_1	-	-	-	-	-	-	-	TRNENSEL
		-	BRDYM	-	-	-	-	-	-
	INTSTS0_1	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY
		VBSTS	DVSQ[2]	DVSQ[1]	DVSQ[0]	VALID	CTSQ[2]	CTSQ[1]	CTSQ[0]
	INTSTS1_1	-	BCHG	-	DTCH	ATTCH	-	-	-
		-	EOFERR	SIGN	SACK	-	-	-	-
	BRDYSTS_1	PIPEBRDY[15]	PIPEBRDY[14]	PIPEBRDY[13]	PIPEBRDY[12]	PIPEBRDY[11]	PIPEBRDY[10]	PIPEBRDY[9]	PIPEBRDY[8]
		PIPEBRDY[7]	PIPEBRDY[6]	PIPEBRDY[5]	PIPEBRDY[4]	PIPEBRDY[3]	PIPEBRDY[2]	PIPEBRDY[1]	PIPEBRDY[0]
	NRDYSTS_1	PIPENRDY[15]	PIPENRDY[14]	PIPENRDY[13]	PIPENRDY[12]	PIPENRDY[11]	PIPENRDY[10]	PIPENRDY[9]	PIPENRDY[8]
		PIPENRDY[7]	PIPENRDY[6]	PIPENRDY[5]	PIPENRDY[4]	PIPENRDY[3]	PIPENRDY[2]	PIPENRDY[1]	PIPENRDY[0]
	BEMPSTS_1	PIPEBEMP[15]	PIPEBEMP[14]	PIPEBEMP[13]	PIPEBEMP[12]	PIPEBEMP[11]	PIPEBEMP[10]	PIPEBEMP[9]	PIPEBEMP[8]
		PIPEBEMP[7]	PIPEBEMP[6]	PIPEBEMP[5]	PIPEBEMP[4]	PIPEBEMP[3]	PIPEBEMP[2]	PIPEBEMP[1]	PIPEBEMP[0]
	FRMNUM_1	OVRN	CRCE	-	-	-	FRNM[10]	FRNM[9]	FRNM[8]
		FRNM[7]	FRNM[6]	FRNM[5]	FRNM[4]	FRNM[3]	FRNM[2]	FRNM[1]	FRNM[0]
	UFRMNUM_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	UFRNM[2]	UFRNM[1]	UFRNM[0]
	USBADDR_1	-	-	-	-	-	-	-	-
		-	USBADDR[6]	USBADDR[5]	USBADDR[4]	USBADDR[3]	USBADDR[2]	USBADDR[1]	USBADDR[0]
	USBREQ_1	bRequest[7]	bRequest[6]	bRequest[5]	bRequest[4]	bRequest[3]	bRequest[2]	bRequest[1]	bRequest[0]
		bmRequestType[7]	bmRequestType[6]	bmRequestType[5]	bmRequestType[4]	bmRequestType[3]	bmRequestType[2]	bmRequestType[1]	bmRequestType[0]
	USBVAL_1	wValue[15]	wValue[14]	wValue[13]	wValue[12]	wValue[11]	wValue[10]	wValue[9]	wValue[8]
		wValue[7]	wValue[6]	wValue[5]	wValue[4]	wValue[3]	wValue[2]	wValue[1]	wValue[0]
	USBINDX_1	wIndex[15]	wIndex[14]	wIndex[13]	wIndex[12]	wIndex[11]	wIndex[10]	wIndex[9]	wIndex[8]
		wIndex[7]	wIndex[6]	wIndex[5]	wIndex[4]	wIndex[3]	wIndex[2]	wIndex[1]	wIndex[0]
	USBLENG_1	wLength[15]	wLength[14]	wLength[13]	wLength[12]	wLength[11]	wLength[10]	wLength[9]	wLength[8]
		wLength[7]	wLength[6]	wLength[5]	wLength[4]	wLength[3]	wLength[2]	wLength[1]	wLength[0]
	DCPCFG_1	-	-	-	-	-	-	-	CNTMD
		SHTNAK	-	-	DIR	-	-	-	-
	DCPMAXP_1	DEVSEL[3]	DEVSEL[2]	DEVSEL[1]	DEVSEL[0]	-	-	-	-
		-	MXPS[6]	MXPS[5]	MXPS[4]	MXPS[3]	MXPS[2]	MXPS[1]	MXPS[0]
	DCPCTR_1	BSTS	SUREQ	CSCLR	CSSTS	SUREQCLR	-	-	SQCLR
		SQSET	SQMON	PBUSY	PINGE	-	CCPL	PID[1]	PID[0]
	PIPESEL_1	-	-	-	-	-	-	-	-
		-	-	-	-	PIPESEL[3]	PIPESEL[2]	PIPESEL[1]	PIPESEL[0]
	PIPECFG_1	TYPE[1]	TYPE[0]	-	-	-	BFRE	DBLB	CNTMD
		SHTNAK	-	-	DIR	EPNUM[3]	EPNUM[2]	EPNUM[1]	EPNUM[0]
	PIPEBUF_1	-	BUFSIZE[4]	BUFSIZE[3]	BUFSIZE[2]	BUFSIZE[1]	BUFSIZE[0]	-	-
		BUFNMB[7]	BUFNMB[6]	BUFNMB[5]	BUFNMB[4]	BUFNMB[3]	BUFNMB[2]	BUFNMB[1]	BUFNMB[0]
	PIPEMAXP_1	DEVSEL[3]	DEVSEL[2]	DEVSEL[1]	DEVSEL[0]	-	MXPS[10]	MXPS[9]	MXPS[8]
		MXPS[7]	MXPS[6]	MXPS[5]	MXPS[4]	MXPS[3]	MXPS[2]	MXPS[1]	MXPS[0]
	PIPEPERI_1	-	-	-	IFIS	-	-	-	-
		-	-	-	-	-	IITV[2]	IITV[1]	IITV[0]
	PIPE1CTR_1	BSTS	INBUFM	CSCLR	CSSTS	-	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPE2CTR_1	BSTS	INBUFM	CSCLR	CSSTS	-	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPE3CTR_1	BSTS	INBUFM	CSCLR	CSSTS	-	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPE4CTR_1	BSTS	INBUFM	CSCLR	CSSTS	-	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPE5CTR_1	BSTS	INBUFM	CSCLR	CSSTS	-	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPE6CTR_1	BSTS	-	CSCLR	CSSTS	-	-	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPE7CTR_1	BSTS	-	CSCLR	CSSTS	-	-	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
USB2.0 host/ function module	PIPE8CTR_1	BSTS	-	CSCLR	CSSTS	-	-	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPE9CTR_1	BSTS	INBUFM	CSCLR	CSSTS	-	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPEACTR_1	BSTS	INBUFM	CSCLR	CSSTS	-	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPEBCTR_1	BSTS	INBUFM	CSCLR	CSSTS	-	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPECCTR_1	BSTS	INBUFM	CSCLR	CSSTS	-	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPEDCTR_1	BSTS	INBUFM	CSCLR	CSSTS	-	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPEECTR_1	BSTS	INBUFM	CSCLR	CSSTS	-	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPEFCTR_1	BSTS	INBUFM	CSCLR	CSSTS	-	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	-	-	-	PID[1]	PID[0]
	PIPE1TRE_1	-	-	-	-	-	-	TRENB	TRCLR
		-	-	-	-	-	-	-	-
	PIPE1TRN_1	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	PIPE2TRE_1	-	-	-	-	-	-	TRENB	TRCLR
		-	-	-	-	-	-	-	-
	PIPE2TRN_1	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	PIPE3TRE_1	-	-	-	-	-	-	TRENB	TRCLR
		-	-	-	-	-	-	-	-
	PIPE3TRN_1	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	PIPE4TRE_1	-	-	-	-	-	-	TRENB	TRCLR
		-	-	-	-	-	-	-	-
	PIPE4TRN_1	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	PIPE5TRE_1	-	-	-	-	-	-	TRENB	TRCLR
		-	-	-	-	-	-	-	-
	PIPE5TRN_1	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	PIPEBTRE_1	-	-	-	-	-	-	TRENB	TRCLR
		-	-	-	-	-	-	-	-
	PIPEBTRN_1	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	PIPECTRE_1	-	-	-	-	-	-	TRENB	TRCLR
		-	-	-	-	-	-	-	-
	PIPECTRN_1	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	PIPEDTRE_1	-	-	-	-	-	-	TRENB	TRCLR
		-	-	-	-	-	-	-	-
	PIPEDTRN_1	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	PIPEETRE_1	-	-	-	-	-	-	TRENB	TRCLR
		-	-	-	-	-	-	-	-
	PIPEETRN_1	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	PIPEFTRE_1	-	-	-	-	-	-	TRENB	TRCLR
		-	-	-	-	-	-	-	-
	PIPEFTRN_1	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	PIPE9TRE_1	-	-	-	-	-	-	TRENB	TRCLR
		-	-	-	-	-	-	-	-
	PIPE9TRN_1	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
USB2.0 host/ function module	PIPEATRE_1	-	-	-	-	-	-	TRENB	TRCLR
		-	-	-	-	-	-	-	-
	PIPEATRN_1	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	DEVADD0_1	-	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	-	-	-	-	-	-
	DEVADD1_1	-	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	-	-	-	-	-	-
	DEVADD2_1	-	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	-	-	-	-	-	-
	DEVADD3_1	-	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	-	-	-	-	-	-
	DEVADD4_1	-	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	-	-	-	-	-	-
	DEVADD5_1	-	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	-	-	-	-	-	-
	DEVADD6_1	-	HPPHUB[3]	HPPHUB[2]	HPPHUB[1]	HPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	-	-	-
	DEVADD7_1	-	HPPHUB[3]	HPPHUB[2]	HPPHUB[1]	HPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	-	-	-	-	-	-
	DEVADD8_1	-	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	-	-	-	-	-	-
	DEVADD9_1	-	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	-	-	-	-	-	-
	DEVADDA_1	-	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	-	-	-	-	-	-
	SUSPMODE_1	-	SUSPM	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	D0FIFOB0_1	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D0FIFOB1_1	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D0FIFOB2_1	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D0FIFOB3_1	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D0FIFOB4_1	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D0FIFOB5_1	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D0FIFOB6_1	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D0FIFOB7_1	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
USB2.0 host/ function module	D1FIFOB0_1	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D1FIFOB1_1	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D1FIFOB2_1	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D1FIFOB3_1	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D1FIFOB4_1	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D1FIFOB5_1	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D1FIFOB6_1	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D1FIFOB7_1	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
Video display controller 5	INP_UPDATE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	INP_EXT_UPDATE	-	-	-	INP_IMG_UPDATE
	INP_SEL_CNT	-	-	-	-	-	-	-	-
		-	-	-	INP_SEL	-	-	-	-
		-	INP_FORMAT[2]	INP_FORMAT[1]	INP_FORMAT[0]	-	-	-	INP_PXD_EDGE
		-	-	-	INP_VS_EDGE	-	-	-	INP_HS_EDGE
	INP_EXT_SYNC_ CNT	-	-	-	INP_ENDIAN_ ON	-	-	-	INP_SWAP_ON
		-	-	-	INP_VS_INV	-	-	-	INP_HS_INV
		-	-	-	-	-	-	-	INP_H_EDGE_ SEL
		-	-	-	INP_F525_625	-	-	INP_H_POS[1]	INP_H_POS[0]
	INP_VSYNC_PH_ ADJ	-	-	-	-	-	-	INP_FH50[9]	INP_FH50[8]
		INP_FH50[7]	INP_FH50[6]	INP_FH50[5]	INP_FH50[4]	INP_FH50[3]	INP_FH50[2]	INP_FH50[1]	INP_FH50[0]
		-	-	-	-	-	-	INP_FH25[9]	INP_FH25[8]
		INP_FH25[7]	INP_FH25[6]	INP_FH25[5]	INP_FH25[4]	INP_FH25[3]	INP_FH25[2]	INP_FH25[1]	INP_FH25[0]
	INP_DLY_ADJ	-	-	-	-	-	INP_VS_DLY_L [2]	INP_VS_DLY_L [1]	INP_VS_DLY_L [0]
		INP_FLD_DLY[7]	INP_FLD_DLY[6]	INP_FLD_DLY[5]	INP_FLD_DLY[4]	INP_FLD_DLY[3]	INP_FLD_DLY[2]	INP_FLD_DLY[1]	INP_FLD_DLY[0]
		INP_VS_DLY[7]	INP_VS_DLY[6]	INP_VS_DLY[5]	INP_VS_DLY[4]	INP_VS_DLY[3]	INP_VS_DLY[2]	INP_VS_DLY[1]	INP_VS_DLY[0]
		INP_HS_DLY[7]	INP_HS_DLY[6]	INP_HS_DLY[5]	INP_HS_DLY[4]	INP_HS_DLY[3]	INP_HS_DLY[2]	INP_HS_DLY[1]	INP_HS_DLY[0]
	IMGCNT_UPDATE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	IMGCNT_VEN

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Video display controller 5	IMGCNT_NR_CNT0	-	-	-	-	-	-	-	-
		-	-	-	NR1D_MD	-	-	-	NR1D_ON
		-	NR1D_Y_TH[6]	NR1D_Y_TH[5]	NR1D_Y_TH[4]	NR1D_Y_TH[3]	NR1D_Y_TH[2]	NR1D_Y_TH[1]	NR1D_Y_TH[0]
		-	-	NR1D_Y_TAP[1]	NR1D_Y_TAP[0]	-	-	NR1D_Y_GAIN[1]	NR1D_Y_GAIN[0]
	IMGCNT_NR_CNT1	-	NR1D_CB_TH[6]	NR1D_CB_TH[5]	NR1D_CB_TH[4]	NR1D_CB_TH[3]	NR1D_CB_TH[2]	NR1D_CB_TH[1]	NR1D_CB_TH[0]
		-	-	NR1D_CB_TAP[1]	NR1D_CB_TAP[0]	-	-	NR1D_CB_GAIN [1]	NR1D_CB_GAIN [0]
		-	NR1D_CR_TH[6]	NR1D_CR_TH[5]	NR1D_CR_TH[4]	NR1D_CR_TH[3]	NR1D_CR_TH[2]	NR1D_CR_TH[1]	NR1D_CR_TH[0]
		-	-	NR1D_CR_TAP[1]	NR1D_CR_TAP[0]	-	-	NR1D_CR_GAIN [1]	NR1D_CR_GAIN [0]
	IMGCNT_MTX_ MODE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	IMGCNT_MTX_ MD[1]	IMGCNT_MTX_ MD[0]
	IMGCNT_MTX_YG_ ADJ0	-	-	-	-	-	-	-	-
		IMGCNT_MTX_ YG[7]	IMGCNT_MTX_ YG[6]	IMGCNT_MTX_ YG[5]	IMGCNT_MTX_ YG[4]	IMGCNT_MTX_ YG[3]	IMGCNT_MTX_ YG[2]	IMGCNT_MTX_ YG[1]	IMGCNT_MTX_ YG[0]
		-	-	-	-	-	IMGCNT_MTX_ GG[10]	IMGCNT_MTX_ GG[9]	IMGCNT_MTX_ GG[8]
		IMGCNT_MTX_ GG[7]	IMGCNT_MTX_ GG[6]	IMGCNT_MTX_ GG[5]	IMGCNT_MTX_ GG[4]	IMGCNT_MTX_ GG[3]	IMGCNT_MTX_ GG[2]	IMGCNT_MTX_ GG[1]	IMGCNT_MTX_ GG[0]
	IMGCNT_MTX_YG_ ADJ1	-	-	-	-	-	IMGCNT_MTX_ GB[10]	IMGCNT_MTX_ GB[9]	IMGCNT_MTX_ GB[8]
		IMGCNT_MTX_ GB[7]	IMGCNT_MTX_ GB[6]	IMGCNT_MTX_ GB[5]	IMGCNT_MTX_ GB[4]	IMGCNT_MTX_ GB[3]	IMGCNT_MTX_ GB[2]	IMGCNT_MTX_ GB[1]	IMGCNT_MTX_ GB[0]
		-	-	-	-	-	IMGCNT_MTX_ GR[10]	IMGCNT_MTX_ GR[9]	IMGCNT_MTX_ GR[8]
		IMGCNT_MTX_ GR[7]	IMGCNT_MTX_ GR[6]	IMGCNT_MTX_ GR[5]	IMGCNT_MTX_ GR[4]	IMGCNT_MTX_ GR[3]	IMGCNT_MTX_ GR[2]	IMGCNT_MTX_ GR[1]	IMGCNT_MTX_ GR[0]
	IMGCNT_MTX_ CBB_ADJ0	-	-	-	-	-	-	-	-
		IMGCNT_MTX_ B[7]	IMGCNT_MTX_ B[6]	IMGCNT_MTX_ B[5]	IMGCNT_MTX_ B[4]	IMGCNT_MTX_ B[3]	IMGCNT_MTX_ B[2]	IMGCNT_MTX_ B[1]	IMGCNT_MTX_ B[0]
		-	-	-	-	-	IMGCNT_MTX_ BG[10]	IMGCNT_MTX_ BG[9]	IMGCNT_MTX_ BG[8]
		IMGCNT_MTX_ BG[7]	IMGCNT_MTX_ BG[6]	IMGCNT_MTX_ BG[5]	IMGCNT_MTX_ BG[4]	IMGCNT_MTX_ BG[3]	IMGCNT_MTX_ BG[2]	IMGCNT_MTX_ BG[1]	IMGCNT_MTX_ BG[0]
	IMGCNT_MTX_ CBB_ADJ1	-	-	-	-	-	IMGCNT_MTX_ BB[10]	IMGCNT_MTX_ BB[9]	IMGCNT_MTX_ BB[8]
		IMGCNT_MTX_ BB[7]	IMGCNT_MTX_ BB[6]	IMGCNT_MTX_ BB[5]	IMGCNT_MTX_ BB[4]	IMGCNT_MTX_ BB[3]	IMGCNT_MTX_ BB[2]	IMGCNT_MTX_ BB[1]	IMGCNT_MTX_ BB[0]
		-	-	-	-	-	IMGCNT_MTX_ BR[10]	IMGCNT_MTX_ BR[9]	IMGCNT_MTX_ BR[8]
		IMGCNT_MTX_ BR[7]	IMGCNT_MTX_ BR[6]	IMGCNT_MTX_ BR[5]	IMGCNT_MTX_ BR[4]	IMGCNT_MTX_ BR[3]	IMGCNT_MTX_ BR[2]	IMGCNT_MTX_ BR[1]	IMGCNT_MTX_ BR[0]
	IMGCNT_MTX_ CRR_ADJ0	-	-	-	-	-	-	-	-
		IMGCNT_MTX_ R[7]	IMGCNT_MTX_ R[6]	IMGCNT_MTX_ R[5]	IMGCNT_MTX_ R[4]	IMGCNT_MTX_ R[3]	IMGCNT_MTX_ R[2]	IMGCNT_MTX_ R[1]	IMGCNT_MTX_ R[0]
		-	-	-	-	-	IMGCNT_MTX_ RG[10]	IMGCNT_MTX_ RG[9]	IMGCNT_MTX_ RG[8]
		IMGCNT_MTX_ RG[7]	IMGCNT_MTX_ RG[6]	IMGCNT_MTX_ RG[5]	IMGCNT_MTX_ RG[4]	IMGCNT_MTX_ RG[3]	IMGCNT_MTX_ RG[2]	IMGCNT_MTX_ RG[1]	IMGCNT_MTX_ RG[0]
	IMGCNT_MTX_ CRR_ADJ1	-	-	-	-	-	IMGCNT_MTX_ RB[10]	IMGCNT_MTX_ RB[9]	IMGCNT_MTX_ RB[8]
		IMGCNT_MTX_ RB[7]	IMGCNT_MTX_ RB[6]	IMGCNT_MTX_ RB[5]	IMGCNT_MTX_ RB[4]	IMGCNT_MTX_ RB[3]	IMGCNT_MTX_ RB[2]	IMGCNT_MTX_ RB[1]	IMGCNT_MTX_ RB[0]
		-	-	-	-	-	IMGCNT_MTX_ RR[10]	IMGCNT_MTX_ RR[9]	IMGCNT_MTX_ RR[8]
		IMGCNT_MTX_ RR[7]	IMGCNT_MTX_ RR[6]	IMGCNT_MTX_ RR[5]	IMGCNT_MTX_ RR[4]	IMGCNT_MTX_ RR[3]	IMGCNT_MTX_ RR[2]	IMGCNT_MTX_ RR[1]	IMGCNT_MTX_ RR[0]
	SC0_SCL0_UPDATE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	SC0_SCL0_ VEN_D	SC0_SCL0_ VEN_C	-	-	-	SC0_SCL0_ UPDATE
		-	-	-	SC0_SCL0_ VEN_B	-	-	-	SC0_SCL0_ VEN_A

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Video display controller 5	SC0_SCL0_FRC1	SC0_RES_VMASK[15]	SC0_RES_VMASK[14]	SC0_RES_VMASK[13]	SC0_RES_VMASK[12]	SC0_RES_VMASK[11]	SC0_RES_VMASK[10]	SC0_RES_VMASK[9]	SC0_RES_VMASK[8]
		SC0_RES_VMASK[7]	SC0_RES_VMASK[6]	SC0_RES_VMASK[5]	SC0_RES_VMASK[4]	SC0_RES_VMASK[3]	SC0_RES_VMASK[2]	SC0_RES_VMASK[1]	SC0_RES_VMASK[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	SC0_RES_VMASK_ON
	SC0_SCL0_FRC2	SC0_RES_VLACK[15]	SC0_RES_VLACK[14]	SC0_RES_VLACK[13]	SC0_RES_VLACK[12]	SC0_RES_VLACK[11]	SC0_RES_VLACK[10]	SC0_RES_VLACK[9]	SC0_RES_VLACK[8]
		SC0_RES_VLACK[7]	SC0_RES_VLACK[6]	SC0_RES_VLACK[5]	SC0_RES_VLACK[4]	SC0_RES_VLACK[3]	SC0_RES_VLACK[2]	SC0_RES_VLACK[1]	SC0_RES_VLACK[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	SC0_RES_VLACK_ON
	SC0_SCL0_FRC3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	SC0_RES_VS_SEL
	SC0_SCL0_FRC4	-	-	-	-	-	SC0_RES_FV[10]	SC0_RES_FV[9]	SC0_RES_FV[8]
		SC0_RES_FV[7]	SC0_RES_FV[6]	SC0_RES_FV[5]	SC0_RES_FV[4]	SC0_RES_FV[3]	SC0_RES_FV[2]	SC0_RES_FV[1]	SC0_RES_FV[0]
		-	-	-	-	-	SC0_RES_FH[10]	SC0_RES_FH[9]	SC0_RES_FH[8]
		SC0_RES_FH[7]	SC0_RES_FH[6]	SC0_RES_FH[5]	SC0_RES_FH[4]	SC0_RES_FH[3]	SC0_RES_FH[2]	SC0_RES_FH[1]	SC0_RES_FH[0]
	SC0_SCL0_FRC5	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	SC0_RES_FLD_DLY_SEL
		SC0_RES_VSDLY[7]	SC0_RES_VSDLY[6]	SC0_RES_VSDLY[5]	SC0_RES_VSDLY[4]	SC0_RES_VSDLY[3]	SC0_RES_VSDLY[2]	SC0_RES_VSDLY[1]	SC0_RES_VSDLY[0]
	SC0_SCL0_FRC6	-	-	-	-	-	SC0_RES_F_VS[10]	SC0_RES_F_VS[9]	SC0_RES_F_VS[8]
		SC0_RES_F_VS[7]	SC0_RES_F_VS[6]	SC0_RES_F_VS[5]	SC0_RES_F_VS[4]	SC0_RES_F_VS[3]	SC0_RES_F_VS[2]	SC0_RES_F_VS[1]	SC0_RES_F_VS[0]
		-	-	-	-	-	SC0_RES_F_VW[10]	SC0_RES_F_VW[9]	SC0_RES_F_VW[8]
		SC0_RES_F_VW[7]	SC0_RES_F_VW[6]	SC0_RES_F_VW[5]	SC0_RES_F_VW[4]	SC0_RES_F_VW[3]	SC0_RES_F_VW[2]	SC0_RES_F_VW[1]	SC0_RES_F_VW[0]
	SC0_SCL0_FRC7	-	-	-	-	-	SC0_RES_F_HS[10]	SC0_RES_F_HS[9]	SC0_RES_F_HS[8]
		SC0_RES_F_HS[7]	SC0_RES_F_HS[6]	SC0_RES_F_HS[5]	SC0_RES_F_HS[4]	SC0_RES_F_HS[3]	SC0_RES_F_HS[2]	SC0_RES_F_HS[1]	SC0_RES_F_HS[0]
		-	-	-	-	-	SC0_RES_F_HW[10]	SC0_RES_F_HW[9]	SC0_RES_F_HW[8]
		SC0_RES_F_HW[7]	SC0_RES_F_HW[6]	SC0_RES_F_HW[5]	SC0_RES_F_HW[4]	SC0_RES_F_HW[3]	SC0_RES_F_HW[2]	SC0_RES_F_HW[1]	SC0_RES_F_HW[0]
	SC0_SCL0_FRC9	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	SC0_RES_QVLOCK	-	-	-	SC0_RES_QVLACK
	SC0_SCL0_MON0	-	-	-	-	-	SC0_RES_LIN_STAT[10]	SC0_RES_LIN_STAT[9]	SC0_RES_LIN_STAT[8]
		SC0_RES_LIN_STAT[7]	SC0_RES_LIN_STAT[6]	SC0_RES_LIN_STAT[5]	SC0_RES_LIN_STAT[4]	SC0_RES_LIN_STAT[3]	SC0_RES_LIN_STAT[2]	SC0_RES_LIN_STAT[1]	SC0_RES_LIN_STAT[0]
	SC0_SCL0_INT	-	-	-	-	-	SC0_RES_LINE[10]	SC0_RES_LINE[9]	SC0_RES_LINE[8]
		SC0_RES_LINE[7]	SC0_RES_LINE[6]	SC0_RES_LINE[5]	SC0_RES_LINE[4]	SC0_RES_LINE[3]	SC0_RES_LINE[2]	SC0_RES_LINE[1]	SC0_RES_LINE[0]
	SC0_SCL0_DS1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	SC0_RES_DSV_ON	-	-	-	SC0_RES_DSH_ON
	SC0_SCL0_DS2	-	-	-	-	-	SC0_RES_VS[10]	SC0_RES_VS[9]	SC0_RES_VS[8]
		SC0_RES_VS[7]	SC0_RES_VS[6]	SC0_RES_VS[5]	SC0_RES_VS[4]	SC0_RES_VS[3]	SC0_RES_VS[2]	SC0_RES_VS[1]	SC0_RES_VS[0]
		-	-	-	-	-	SC0_RES_VW[10]	SC0_RES_VW[9]	SC0_RES_VW[8]
		SC0_RES_VW[7]	SC0_RES_VW[6]	SC0_RES_VW[5]	SC0_RES_VW[4]	SC0_RES_VW[3]	SC0_RES_VW[2]	SC0_RES_VW[1]	SC0_RES_VW[0]
	SC0_SCL0_DS3	-	-	-	-	-	SC0_RES_HS[10]	SC0_RES_HS[9]	SC0_RES_HS[8]
		SC0_RES_HS[7]	SC0_RES_HS[6]	SC0_RES_HS[5]	SC0_RES_HS[4]	SC0_RES_HS[3]	SC0_RES_HS[2]	SC0_RES_HS[1]	SC0_RES_HS[0]
		-	-	-	-	-	SC0_RES_HW[10]	SC0_RES_HW[9]	SC0_RES_HW[8]
		SC0_RES_HW[7]	SC0_RES_HW[6]	SC0_RES_HW[5]	SC0_RES_HW[4]	SC0_RES_HW[3]	SC0_RES_HW[2]	SC0_RES_HW[1]	SC0_RES_HW[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Video display controller 5	SC0_SCL0_DS4	-	-	SC0_RES_PFIL_SEL	SC0_RES_DS_H_INTERPOTYP	-	-	-	-
		-	-	-	-	-	-	-	-
		SC0_RES_DS_H_RATIO[15]	SC0_RES_DS_H_RATIO[14]	SC0_RES_DS_H_RATIO[13]	SC0_RES_DS_H_RATIO[12]	SC0_RES_DS_H_RATIO[11]	SC0_RES_DS_H_RATIO[10]	SC0_RES_DS_H_RATIO[9]	SC0_RES_DS_H_RATIO[8]
		SC0_RES_DS_H_RATIO[7]	SC0_RES_DS_H_RATIO[6]	SC0_RES_DS_H_RATIO[5]	SC0_RES_DS_H_RATIO[4]	SC0_RES_DS_H_RATIO[3]	SC0_RES_DS_H_RATIO[2]	SC0_RES_DS_H_RATIO[1]	SC0_RES_DS_H_RATIO[0]
	SC0_SCL0_DS5	-	-	-	SC0_RES_V_INTERPOTYP	SC0_RES_TOP_INIPHASE[11]	SC0_RES_TOP_INIPHASE[10]	SC0_RES_TOP_INIPHASE[9]	SC0_RES_TOP_INIPHASE[8]
		SC0_RES_TOP_INIPHASE[7]	SC0_RES_TOP_INIPHASE[6]	SC0_RES_TOP_INIPHASE[5]	SC0_RES_TOP_INIPHASE[4]	SC0_RES_TOP_INIPHASE[3]	SC0_RES_TOP_INIPHASE[2]	SC0_RES_TOP_INIPHASE[1]	SC0_RES_TOP_INIPHASE[0]
		-	-	-	-	SC0_RES_BTM_INIPHASE[11]	SC0_RES_BTM_INIPHASE[10]	SC0_RES_BTM_INIPHASE[9]	SC0_RES_BTM_INIPHASE[8]
		SC0_RES_BTM_INIPHASE[7]	SC0_RES_BTM_INIPHASE[6]	SC0_RES_BTM_INIPHASE[5]	SC0_RES_BTM_INIPHASE[4]	SC0_RES_BTM_INIPHASE[3]	SC0_RES_BTM_INIPHASE[2]	SC0_RES_BTM_INIPHASE[1]	SC0_RES_BTM_INIPHASE[0]
	SC0_SCL0_DS6	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		SC0_RES_V_RATIO[15]	SC0_RES_V_RATIO[14]	SC0_RES_V_RATIO[13]	SC0_RES_V_RATIO[12]	SC0_RES_V_RATIO[11]	SC0_RES_V_RATIO[10]	SC0_RES_V_RATIO[9]	SC0_RES_V_RATIO[8]
		SC0_RES_V_RATIO[7]	SC0_RES_V_RATIO[6]	SC0_RES_V_RATIO[5]	SC0_RES_V_RATIO[4]	SC0_RES_V_RATIO[3]	SC0_RES_V_RATIO[2]	SC0_RES_V_RATIO[1]	SC0_RES_V_RATIO[0]
	SC0_SCL0_DS7	-	-	-	-	-	SC0_RES_OUT_VW[10]	SC0_RES_OUT_VW[9]	SC0_RES_OUT_VW[8]
		SC0_RES_OUT_VW[7]	SC0_RES_OUT_VW[6]	SC0_RES_OUT_VW[5]	SC0_RES_OUT_VW[4]	SC0_RES_OUT_VW[3]	SC0_RES_OUT_VW[2]	SC0_RES_OUT_VW[1]	SC0_RES_OUT_VW[0]
		-	-	-	-	-	SC0_RES_OUT_HW[10]	SC0_RES_OUT_HW[9]	SC0_RES_OUT_HW[8]
		SC0_RES_OUT_HW[7]	SC0_RES_OUT_HW[6]	SC0_RES_OUT_HW[5]	SC0_RES_OUT_HW[4]	SC0_RES_OUT_HW[3]	SC0_RES_OUT_HW[2]	SC0_RES_OUT_HW[1]	SC0_RES_OUT_HW[0]
	SC0_SCL0_US1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	SC0_RES_US_V_ON	-	-	-	SC0_RES_US_H_ON
	SC0_SCL0_US2	-	-	-	-	-	SC0_RES_P_VS[10]	SC0_RES_P_VS[9]	SC0_RES_P_VS[8]
		SC0_RES_P_VS[7]	SC0_RES_P_VS[6]	SC0_RES_P_VS[5]	SC0_RES_P_VS[4]	SC0_RES_P_VS[3]	SC0_RES_P_VS[2]	SC0_RES_P_VS[1]	SC0_RES_P_VS[0]
		-	-	-	-	-	SC0_RES_P_VW[10]	SC0_RES_P_VW[9]	SC0_RES_P_VW[8]
		SC0_RES_P_VW[7]	SC0_RES_P_VW[6]	SC0_RES_P_VW[5]	SC0_RES_P_VW[4]	SC0_RES_P_VW[3]	SC0_RES_P_VW[2]	SC0_RES_P_VW[1]	SC0_RES_P_VW[0]
	SC0_SCL0_US3	-	-	-	-	-	SC0_RES_P_HS[10]	SC0_RES_P_HS[9]	SC0_RES_P_HS[8]
		SC0_RES_P_HS[7]	SC0_RES_P_HS[6]	SC0_RES_P_HS[5]	SC0_RES_P_HS[4]	SC0_RES_P_HS[3]	SC0_RES_P_HS[2]	SC0_RES_P_HS[1]	SC0_RES_P_HS[0]
		-	-	-	-	-	SC0_RES_P_HW[10]	SC0_RES_P_HW[9]	SC0_RES_P_HW[8]
		SC0_RES_P_HW[7]	SC0_RES_P_HW[6]	SC0_RES_P_HW[5]	SC0_RES_P_HW[4]	SC0_RES_P_HW[3]	SC0_RES_P_HW[2]	SC0_RES_P_HW[1]	SC0_RES_P_HW[0]
	SC0_SCL0_US4	-	-	-	-	-	SC0_RES_IN_VW[10]	SC0_RES_IN_VW[9]	SC0_RES_IN_VW[8]
		SC0_RES_IN_VW[7]	SC0_RES_IN_VW[6]	SC0_RES_IN_VW[5]	SC0_RES_IN_VW[4]	SC0_RES_IN_VW[3]	SC0_RES_IN_VW[2]	SC0_RES_IN_VW[1]	SC0_RES_IN_VW[0]
		-	-	-	-	-	SC0_RES_IN_HW[10]	SC0_RES_IN_HW[9]	SC0_RES_IN_HW[8]
		SC0_RES_IN_HW[7]	SC0_RES_IN_HW[6]	SC0_RES_IN_HW[5]	SC0_RES_IN_HW[4]	SC0_RES_IN_HW[3]	SC0_RES_IN_HW[2]	SC0_RES_IN_HW[1]	SC0_RES_IN_HW[0]
	SC0_SCL0_US5	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		SC0_RES_US_H_RATIO[15]	SC0_RES_US_H_RATIO[14]	SC0_RES_US_H_RATIO[13]	SC0_RES_US_H_RATIO[12]	SC0_RES_US_H_RATIO[11]	SC0_RES_US_H_RATIO[10]	SC0_RES_US_H_RATIO[9]	SC0_RES_US_H_RATIO[8]
		SC0_RES_US_H_RATIO[7]	SC0_RES_US_H_RATIO[6]	SC0_RES_US_H_RATIO[5]	SC0_RES_US_H_RATIO[4]	SC0_RES_US_H_RATIO[3]	SC0_RES_US_H_RATIO[2]	SC0_RES_US_H_RATIO[1]	SC0_RES_US_H_RATIO[0]
	SC0_SCL0_US6	-	-	-	SC0_RES_US_H_INTERPOTYP	SC0_RES_US_HT_INIPHASE[11]	SC0_RES_US_HT_INIPHASE[10]	SC0_RES_US_HT_INIPHASE[9]	SC0_RES_US_HT_INIPHASE[8]
		SC0_RES_US_HT_INIPHASE[7]	SC0_RES_US_HT_INIPHASE[6]	SC0_RES_US_HT_INIPHASE[5]	SC0_RES_US_HT_INIPHASE[4]	SC0_RES_US_HT_INIPHASE[3]	SC0_RES_US_HT_INIPHASE[2]	SC0_RES_US_HT_INIPHASE[1]	SC0_RES_US_HT_INIPHASE[0]
		-	-	-	-	SC0_RES_US_HB_INIPHASE[11]	SC0_RES_US_HB_INIPHASE[10]	SC0_RES_US_HB_INIPHASE[9]	SC0_RES_US_HB_INIPHASE[8]
		SC0_RES_US_HB_INIPHASE[7]	SC0_RES_US_HB_INIPHASE[6]	SC0_RES_US_HB_INIPHASE[5]	SC0_RES_US_HB_INIPHASE[4]	SC0_RES_US_HB_INIPHASE[3]	SC0_RES_US_HB_INIPHASE[2]	SC0_RES_US_HB_INIPHASE[1]	SC0_RES_US_HB_INIPHASE[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Video display controller 5	SC0_SCL0_US7	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		SC0_RES_HCUT [7]	SC0_RES_HCUT [6]	SC0_RES_HCUT [5]	SC0_RES_HCUT [4]	SC0_RES_HCUT [3]	SC0_RES_HCUT [2]	SC0_RES_HCUT [1]	SC0_RES_HCUT [0]
		SC0_RES_VCUT [7]	SC0_RES_VCUT [6]	SC0_RES_VCUT [5]	SC0_RES_VCUT [4]	SC0_RES_VCUT [3]	SC0_RES_VCUT [2]	SC0_RES_VCUT [1]	SC0_RES_VCUT [0]
	SC0_SCL0_US8	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	SC0_RES_IBUS_SYNC_SEL	-	-	-	SC0_RES_DISP_ON
	SC0_SCL0_OVR1	-	-	-	-	-	-	-	-
		SC0_RES_BK_COL_R[7]	SC0_RES_BK_COL_R[6]	SC0_RES_BK_COL_R[5]	SC0_RES_BK_COL_R[4]	SC0_RES_BK_COL_R[3]	SC0_RES_BK_COL_R[2]	SC0_RES_BK_COL_R[1]	SC0_RES_BK_COL_R[0]
		SC0_RES_BK_COL_G[7]	SC0_RES_BK_COL_G[6]	SC0_RES_BK_COL_G[5]	SC0_RES_BK_COL_G[4]	SC0_RES_BK_COL_G[3]	SC0_RES_BK_COL_G[2]	SC0_RES_BK_COL_G[1]	SC0_RES_BK_COL_G[0]
		SC0_RES_BK_COL_B[7]	SC0_RES_BK_COL_B[6]	SC0_RES_BK_COL_B[5]	SC0_RES_BK_COL_B[4]	SC0_RES_BK_COL_B[3]	SC0_RES_BK_COL_B[2]	SC0_RES_BK_COL_B[1]	SC0_RES_BK_COL_B[0]
	SC0_SCL1_UPDATE	-	-	-	-	-	-	-	-
		-	-	-	SC0_SCL1_UPDATE_B	-	-	-	SC0_SCL1_UPDATE_A
		-	-	-	-	-	-	-	-
		-	-	-	SC0_SCL1_VEN_B	-	-	-	SC0_SCL1_VEN_A
	SC0_SCL1_WR1	-	-	-	-	-	-	-	-
		-	-	-	-	-	SC0_RES_WRSWA[2]	SC0_RES_WRSWA[1]	SC0_RES_WRSWA[0]
		-	-	-	-	-	-	-	-
		SC0_RES_TB_ADD_MOD	SC0_RES_DS_WR_MD[2]	SC0_RES_DS_WR_MD[1]	SC0_RES_DS_WR_MD[0]	SC0_RES_MD[1]	SC0_RES_MD[0]	SC0_RES_LOOP	SC0_RES_BST_MD
	SC0_SCL1_WR2	SC0_RES_BASE [31]	SC0_RES_BASE [30]	SC0_RES_BASE [29]	SC0_RES_BASE [28]	SC0_RES_BASE [27]	SC0_RES_BASE [26]	SC0_RES_BASE [25]	SC0_RES_BASE [24]
		SC0_RES_BASE[23]	SC0_RES_BASE [22]	SC0_RES_BASE [21]	SC0_RES_BASE [20]	SC0_RES_BASE [19]	SC0_RES_BASE [18]	SC0_RES_BASE [17]	SC0_RES_BASE [16]
		SC0_RES_BASE [15]	SC0_RES_BASE [14]	SC0_RES_BASE [13]	SC0_RES_BASE [12]	SC0_RES_BASE [11]	SC0_RES_BASE [10]	SC0_RES_BASE [9]	SC0_RES_BASE [8]
		SC0_RES_BASE [7]	SC0_RES_BASE [6]	SC0_RES_BASE [5]	SC0_RES_BASE [4]	SC0_RES_BASE [3]	SC0_RES_BASE [2]	SC0_RES_BASE [1]	SC0_RES_BASE [0]
	SC0_SCL1_WR3	-	SC0_RES_LN_OFF[14]	SC0_RES_LN_OFF[13]	SC0_RES_LN_OFF[12]	SC0_RES_LN_OFF[11]	SC0_RES_LN_OFF[10]	SC0_RES_LN_OFF[9]	SC0_RES_LN_OFF[8]
		SC0_RES_LN_OFF[7]	SC0_RES_LN_OFF[6]	SC0_RES_LN_OFF[5]	SC0_RES_LN_OFF[4]	SC0_RES_LN_OFF[3]	SC0_RES_LN_OFF[2]	SC0_RES_LN_OFF[1]	SC0_RES_LN_OFF[0]
		-	-	-	-	-	-	SC0_RES_FLM_NUM [9]	SC0_RES_FLM_NUM [8]
		SC0_RES_FLM_NUM [7]	SC0_RES_FLM_NUM [6]	SC0_RES_FLM_NUM [5]	SC0_RES_FLM_NUM [4]	SC0_RES_FLM_NUM [3]	SC0_RES_FLM_NUM [2]	SC0_RES_FLM_NUM [1]	SC0_RES_FLM_NUM [0]
	SC0_SCL1_WR4	-	-	-	-	-	-	-	-
		-	SC0_RES_FLM_OFF [22]	SC0_RES_FLM_OFF [21]	SC0_RES_FLM_OFF [20]	SC0_RES_FLM_OFF [19]	SC0_RES_FLM_OFF [18]	SC0_RES_FLM_OFF [17]	SC0_RES_FLM_OFF [16]
		SC0_RES_FLM_OFF [15]	SC0_RES_FLM_OFF [14]	SC0_RES_FLM_OFF [13]	SC0_RES_FLM_OFF [12]	SC0_RES_FLM_OFF [11]	SC0_RES_FLM_OFF [10]	SC0_RES_FLM_OFF [9]	SC0_RES_FLM_OFF [8]
		SC0_RES_FLM_OFF [7]	SC0_RES_FLM_OFF [6]	SC0_RES_FLM_OFF [5]	SC0_RES_FLM_OFF [4]	SC0_RES_FLM_OFF [3]	SC0_RES_FLM_OFF [2]	SC0_RES_FLM_OFF [1]	SC0_RES_FLM_OFF [0]
	SC0_SCL1_WR5	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	SC0_RES_INTER	-	-	SC0_RES_FS_RATE[1]	SC0_RES_FS_RATE[0]
		-	-	-	SC0_RES_FLD_SEL	-	-	-	SC0_RES_WENB
	SC0_SCL1_WR6	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	SC0_RES_DTH_ON	-	-	-	SC0_RES_BITDEC_ON
	SC0_SCL1_WR7	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	SC0_RES_OVERFLOW
		-	-	-	-	-	-	SC0_RES_FLM_CNT [9]	SC0_RES_FLM_CNT [8]
		SC0_RES_FLM_CNT [7]	SC0_RES_FLM_CNT [6]	SC0_RES_FLM_CNT [5]	SC0_RES_FLM_CNT [4]	SC0_RES_FLM_CNT [3]	SC0_RES_FLM_CNT [2]	SC0_RES_FLM_CNT [1]	SC0_RES_FLM_CNT [0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Video display controller 5	SC0_SCL1_WR8	SC0_RES_BASE_B [31]	SC0_RES_BASE_B [30]	SC0_RES_BASE_B [29]	SC0_RES_BASE_B [28]	SC0_RES_BASE_B [27]	SC0_RES_BASE_B [26]	SC0_RES_BASE_B [25]	SC0_RES_BASE_B [24]
		SC0_RES_BASE_B [23]	SC0_RES_BASE_B [22]	SC0_RES_BASE_B [21]	SC0_RES_BASE_B [20]	SC0_RES_BASE_B [19]	SC0_RES_BASE_B [18]	SC0_RES_BASE_B [17]	SC0_RES_BASE_B [16]
		SC0_RES_BASE_B [15]	SC0_RES_BASE_B [14]	SC0_RES_BASE_B [13]	SC0_RES_BASE_B [12]	SC0_RES_BASE_B [11]	SC0_RES_BASE_B [10]	SC0_RES_BASE_B [9]	SC0_RES_BASE_B [8]
		SC0_RES_BASE_B[7]	SC0_RES_BASE_B[6]	SC0_RES_BASE_B[5]	SC0_RES_BASE_B[4]	SC0_RES_BASE_B[3]	SC0_RES_BASE_B[2]	SC0_RES_BASE_B[1]	SC0_RES_BASE_B[0]
	SC0_SCL1_WR9	-	SC0_RES_LN_ OFF_B[14]	SC0_RES_LN_ OFF_B[13]	SC0_RES_LN_ OFF_B[12]	SC0_RES_LN_ OFF_B[11]	SC0_RES_LN_ OFF_B[10]	SC0_RES_LN_ OFF_B[9]	SC0_RES_LN_ OFF_B[8]
		SC0_RES_LN_ OFF_B[7]	SC0_RES_LN_ OFF_B[6]	SC0_RES_LN_ OFF_B[5]	SC0_RES_LN_ OFF_B[4]	SC0_RES_LN_ OFF_B[3]	SC0_RES_LN_ OFF_B[2]	SC0_RES_LN_ OFF_B[1]	SC0_RES_LN_ OFF_B[0]
		-	-	-	-	-	-	SC0_RES_FLM_NUM _B[9]	SC0_RES_FLM_NUM _B[8]
		SC0_RES_FLM_NUM _B[7]	SC0_RES_FLM_NUM _B[6]	SC0_RES_FLM_NUM _B[5]	SC0_RES_FLM_NUM _B[4]	SC0_RES_FLM_NUM _B[3]	SC0_RES_FLM_NUM _B[2]	SC0_RES_FLM_NUM _B[1]	SC0_RES_FLM_NUM _B[0]
	SC0_SCL1_WR10	-	-	-	-	-	-	-	-
		-	SC0_RES_FLM_OFF _B[22]	SC0_RES_FLM_OFF _B[21]	SC0_RES_FLM_OFF _B[20]	SC0_RES_FLM_OFF _B[19]	SC0_RES_FLM_OFF _B[18]	SC0_RES_FLM_OFF _B[17]	SC0_RES_FLM_OFF _B[16]
		SC0_RES_FLM_OFF _B[15]	SC0_RES_FLM_OFF _B[14]	SC0_RES_FLM_OFF _B[13]	SC0_RES_FLM_OFF _B[12]	SC0_RES_FLM_OFF _B[11]	SC0_RES_FLM_OFF _B[10]	SC0_RES_FLM_OFF _B[9]	SC0_RES_FLM_OFF _B[8]
		SC0_RES_FLM_OFF _B[7]	SC0_RES_FLM_OFF _B[6]	SC0_RES_FLM_OFF _B[5]	SC0_RES_FLM_OFF _B[4]	SC0_RES_FLM_OFF _B[3]	SC0_RES_FLM_OFF _B[2]	SC0_RES_FLM_OFF _B[1]	SC0_RES_FLM_OFF _B[0]
	SC0_SCL1_WR11	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	SC0_RES_FLM_CNT _B[9]	SC0_RES_FLM_CNT _B[8]
		SC0_RES_FLM_CNT _B[7]	SC0_RES_FLM_CNT _B[6]	SC0_RES_FLM_CNT _B[5]	SC0_RES_FLM_CNT _B[4]	SC0_RES_FLM_CNT _B[3]	SC0_RES_FLM_CNT _B[2]	SC0_RES_FLM_CNT _B[1]	SC0_RES_FLM_CNT _B[0]
	SC0_SCL1_MON1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	SC0_PBUF_NUM[1]	SC0_PBUF_NUM[0]
		-	-	-	-	-	-	-	-
	SC0_SCL1_PBUF0	SC0_PBUF0_ ADD[31]	SC0_PBUF0_ ADD[30]	SC0_PBUF0_ ADD[29]	SC0_PBUF0_ ADD[28]	SC0_PBUF0_ ADD[27]	SC0_PBUF0_ ADD[26]	SC0_PBUF0_ ADD[25]	SC0_PBUF0_ ADD[24]
		SC0_PBUF0_ ADD[23]	SC0_PBUF0_ ADD[22]	SC0_PBUF0_ ADD[21]	SC0_PBUF0_ ADD[20]	SC0_PBUF0_ ADD[19]	SC0_PBUF0_ ADD[18]	SC0_PBUF0_ ADD[17]	SC0_PBUF0_ ADD[16]
		SC0_PBUF0_ ADD[15]	SC0_PBUF0_ ADD[14]	SC0_PBUF0_ ADD[13]	SC0_PBUF0_ ADD[12]	SC0_PBUF0_ ADD[11]	SC0_PBUF0_ ADD[10]	SC0_PBUF0_ ADD[9]	SC0_PBUF0_ ADD[8]
		SC0_PBUF0_ ADD[7]	SC0_PBUF0_ ADD[6]	SC0_PBUF0_ ADD[5]	SC0_PBUF0_ ADD[4]	SC0_PBUF0_ ADD[3]	SC0_PBUF0_ ADD[2]	SC0_PBUF0_ ADD[1]	SC0_PBUF0_ ADD[0]
	SC0_SCL1_PBUF1	SC0_PBUF1_ ADD[31]	SC0_PBUF1_ ADD[30]	SC0_PBUF1_ ADD[29]	SC0_PBUF1_ ADD[28]	SC0_PBUF1_ ADD[27]	SC0_PBUF1_ ADD[26]	SC0_PBUF1_ ADD[25]	SC0_PBUF1_ ADD[24]
		SC0_PBUF1_ ADD[23]	SC0_PBUF1_ ADD[22]	SC0_PBUF1_ ADD[21]	SC0_PBUF1_ ADD[20]	SC0_PBUF1_ ADD[19]	SC0_PBUF1_ ADD[18]	SC0_PBUF1_ ADD[17]	SC0_PBUF1_ ADD[16]
		SC0_PBUF1_ ADD[15]	SC0_PBUF1_ ADD[14]	SC0_PBUF1_ ADD[13]	SC0_PBUF1_ ADD[12]	SC0_PBUF1_ ADD[11]	SC0_PBUF1_ ADD[10]	SC0_PBUF1_ ADD[9]	SC0_PBUF1_ ADD[8]
		SC0_PBUF1_ ADD[7]	SC0_PBUF1_ ADD[6]	SC0_PBUF1_ ADD[5]	SC0_PBUF1_ ADD[4]	SC0_PBUF1_ ADD[3]	SC0_PBUF1_ ADD[2]	SC0_PBUF1_ ADD[1]	SC0_PBUF1_ ADD[0]
	SC0_SCL1_PBUF2	SC0_PBUF2_ ADD[31]	SC0_PBUF2_ ADD[30]	SC0_PBUF2_ ADD[29]	SC0_PBUF2_ ADD[28]	SC0_PBUF2_ ADD[27]	SC0_PBUF2_ ADD[26]	SC0_PBUF2_ ADD[25]	SC0_PBUF2_ ADD[24]
		SC0_PBUF2_ ADD[23]	SC0_PBUF2_ ADD[22]	SC0_PBUF2_ ADD[21]	SC0_PBUF2_ ADD[20]	SC0_PBUF2_ ADD[19]	SC0_PBUF2_ ADD[18]	SC0_PBUF2_ ADD[17]	SC0_PBUF2_ ADD[16]
		SC0_PBUF2_ ADD[15]	SC0_PBUF2_ ADD[14]	SC0_PBUF2_ ADD[13]	SC0_PBUF2_ ADD[12]	SC0_PBUF2_ ADD[11]	SC0_PBUF2_ ADD[10]	SC0_PBUF2_ ADD[9]	SC0_PBUF2_ ADD[8]
		SC0_PBUF2_ ADD[7]	SC0_PBUF2_ ADD[6]	SC0_PBUF2_ ADD[5]	SC0_PBUF2_ ADD[4]	SC0_PBUF2_ ADD[3]	SC0_PBUF2_ ADD[2]	SC0_PBUF2_ ADD[1]	SC0_PBUF2_ ADD[0]
	SC0_SCL1_PBUF3	SC0_PBUF3_ ADD[31]	SC0_PBUF3_ ADD[30]	SC0_PBUF3_ ADD[29]	SC0_PBUF3_ ADD[28]	SC0_PBUF3_ ADD[27]	SC0_PBUF3_ ADD[26]	SC0_PBUF3_ ADD[25]	SC0_PBUF3_ ADD[24]
		SC0_PBUF3_ ADD[23]	SC0_PBUF3_ ADD[22]	SC0_PBUF3_ ADD[21]	SC0_PBUF3_ ADD[20]	SC0_PBUF3_ ADD[19]	SC0_PBUF3_ ADD[18]	SC0_PBUF3_ ADD[17]	SC0_PBUF3_ ADD[16]
		SC0_PBUF3_ ADD[15]	SC0_PBUF3_ ADD[14]	SC0_PBUF3_ ADD[13]	SC0_PBUF3_ ADD[12]	SC0_PBUF3_ ADD[11]	SC0_PBUF3_ ADD[10]	SC0_PBUF3_ ADD[9]	SC0_PBUF3_ ADD[8]
		SC0_PBUF3_ ADD[7]	SC0_PBUF3_ ADD[6]	SC0_PBUF3_ ADD[5]	SC0_PBUF3_ ADD[4]	SC0_PBUF3_ ADD[3]	SC0_PBUF3_ ADD[2]	SC0_PBUF3_ ADD[1]	SC0_PBUF3_ ADD[0]
	SC0_SCL1_PBUF_ FLD	-	-	-	-	-	-	-	SC0_FLD_INF3
		-	-	-	-	-	-	-	SC0_FLD_INF2
		-	-	-	-	-	-	-	SC0_FLD_INF1
		-	-	-	-	-	-	-	SC0_FLD_INF0
	SC0_SCL1_PBUF_ CNT	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	SC0_PBUF_RST
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Video display controller 5	GR0_UPDATE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	GR0_UPDATE
		-	-	-	GR0_P_VEN	-	-	-	GR0_IBUS_VEN
	GR0_FLM_RD	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	GR0_R_ENB
	GR0_FLM1	GR0_FLD_SEL	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	GR0_LN_OFF_DIR
		-	-	-	-	-	-	GR0_FLM_SEL[1]	GR0_FLM_SEL[0]
		-	-	-	-	-	-	-	GR0_BST_MD
	GR0_FLM2	GR0_BASE[31]	GR0_BASE[30]	GR0_BASE[29]	GR0_BASE[28]	GR0_BASE[27]	GR0_BASE[26]	GR0_BASE[25]	GR0_BASE[24]
		GR0_BASE[23]	GR0_BASE[22]	GR0_BASE[21]	GR0_BASE[20]	GR0_BASE[19]	GR0_BASE[18]	GR0_BASE[17]	GR0_BASE[16]
		GR0_BASE[15]	GR0_BASE[14]	GR0_BASE[13]	GR0_BASE[12]	GR0_BASE[11]	GR0_BASE[10]	GR0_BASE[9]	GR0_BASE[8]
		GR0_BASE[7]	GR0_BASE[6]	GR0_BASE[5]	GR0_BASE[4]	GR0_BASE[3]	GR0_BASE[2]	GR0_BASE[1]	GR0_BASE[0]
	GR0_FLM3	GR0_FLD_NXT	GR0_LN_OFF[14]	GR0_LN_OFF[13]	GR0_LN_OFF[12]	GR0_LN_OFF[11]	GR0_LN_OFF[10]	GR0_LN_OFF[9]	GR0_LN_OFF[8]
		GR0_LN_OFF[7]	GR0_LN_OFF[6]	GR0_LN_OFF[5]	GR0_LN_OFF[4]	GR0_LN_OFF[3]	GR0_LN_OFF[2]	GR0_LN_OFF[1]	GR0_LN_OFF[0]
		-	-	-	-	-	-	GR0_FLM_NUM [9]	GR0_FLM_NUM [8]
		GR0_FLM_NUM [7]	GR0_FLM_NUM [6]	GR0_FLM_NUM [5]	GR0_FLM_NUM [4]	GR0_FLM_NUM [3]	GR0_FLM_NUM [2]	GR0_FLM_NUM [1]	GR0_FLM_NUM [0]
	GR0_FLM4	-	-	-	-	-	-	-	-
		-	GR0_FLM_OFF [22]	GR0_FLM_OFF [21]	GR0_FLM_OFF [20]	GR0_FLM_OFF [19]	GR0_FLM_OFF [18]	GR0_FLM_OFF [17]	GR0_FLM_OFF [16]
		GR0_FLM_OFF [15]	GR0_FLM_OFF [14]	GR0_FLM_OFF [13]	GR0_FLM_OFF [12]	GR0_FLM_OFF [11]	GR0_FLM_OFF [10]	GR0_FLM_OFF[9]	GR0_FLM_OFF[8]
		GR0_FLM_OFF[7]	GR0_FLM_OFF[6]	GR0_FLM_OFF[5]	GR0_FLM_OFF[4]	GR0_FLM_OFF[3]	GR0_FLM_OFF[2]	GR0_FLM_OFF[1]	GR0_FLM_OFF[0]
	GR0_FLM5	-	-	-	-	-	GR0_FLM_LNUM[10]	GR0_FLM_LNUM[9]	GR0_FLM_LNUM[8]
		GR0_FLM_LNUM[7]	GR0_FLM_LNUM[6]	GR0_FLM_LNUM[5]	GR0_FLM_LNUM[4]	GR0_FLM_LNUM[3]	GR0_FLM_LNUM[2]	GR0_FLM_LNUM[1]	GR0_FLM_LNUM[0]
		-	-	-	-	-	GR0_FLM_LOOP [10]	GR0_FLM_LOOP [9]	GR0_FLM_LOOP [8]
		GR0_FLM_LOOP [7]	GR0_FLM_LOOP [6]	GR0_FLM_LOOP [5]	GR0_FLM_LOOP [4]	GR0_FLM_LOOP [3]	GR0_FLM_LOOP [2]	GR0_FLM_LOOP [1]	GR0_FLM_LOOP [0]
	GR0_FLM6	GR0_FORMAT[3]	GR0_FORMAT[2]	GR0_FORMAT[1]	GR0_FORMAT[0]	-	GR0_HW[10]	GR0_HW[9]	GR0_HW[8]
		GR0_HW[7]	GR0_HW[6]	GR0_HW[5]	GR0_HW[4]	GR0_HW[3]	GR0_HW[2]	GR0_HW[1]	GR0_HW[0]
		GR0_YCC_SWAP[2]	GR0_YCC_SWAP[1]	GR0_YCC_SWAP[0]	GR0_RDSWA[2]	GR0_RDSWA[1]	GR0_RDSWA[0]	-	GR0_CNV444_MD
		-	-	GR0_STA_POS[5]	GR0_STA_POS[4]	GR0_STA_POS[3]	GR0_STA_POS[2]	GR0_STA_POS[1]	GR0_STA_POS[0]
	GR0_AB1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	GR0_GRC_DISP_ON	-	-	GR0_DISP_SEL [1]	GR0_DISP_SEL [0]
	GR0_AB2	-	-	-	-	-	GR0_GRC_VS [10]	GR0_GRC_VS[9]	GR0_GRC_VS[8]
		GR0_GRC_VS[7]	GR0_GRC_VS[6]	GR0_GRC_VS[5]	GR0_GRC_VS[4]	GR0_GRC_VS[3]	GR0_GRC_VS[2]	GR0_GRC_VS[1]	GR0_GRC_VS[0]
		-	-	-	-	-	GR0_GRC_VW [10]	GR0_GRC_VW[9]	GR0_GRC_VW[8]
		GR0_GRC_VW[7]	GR0_GRC_VW[6]	GR0_GRC_VW[5]	GR0_GRC_VW[4]	GR0_GRC_VW[3]	GR0_GRC_VW[2]	GR0_GRC_VW[1]	GR0_GRC_VW[0]
	GR0_AB3	-	-	-	-	-	GR0_GRC_HS [10]	GR0_GRC_HS[9]	GR0_GRC_HS[8]
		GR0_GRC_HS[7]	GR0_GRC_HS[6]	GR0_GRC_HS[5]	GR0_GRC_HS[4]	GR0_GRC_HS[3]	GR0_GRC_HS[2]	GR0_GRC_HS[1]	GR0_GRC_HS[0]
		-	-	-	-	-	GR0_GRC_HW [10]	GR0_GRC_HW[9]	GR0_GRC_HW[8]
		GR0_GRC_HW[7]	GR0_GRC_HW[6]	GR0_GRC_HW[5]	GR0_GRC_HW[4]	GR0_GRC_HW[3]	GR0_GRC_HW[2]	GR0_GRC_HW[1]	GR0_GRC_HW[0]
	GR0_AB7	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	GR0_CK_ON
	GR0_AB8	GR0_CK_KCLUT [7]	GR0_CK_KCLUT [6]	GR0_CK_KCLUT [5]	GR0_CK_KCLUT [4]	GR0_CK_KCLUT [3]	GR0_CK_KCLUT [2]	GR0_CK_KCLUT [1]	GR0_CK_KCLUT [0]
		GR0_CK_KG[7]	GR0_CK_KG[6]	GR0_CK_KG[5]	GR0_CK_KG[4]	GR0_CK_KG[3]	GR0_CK_KG[2]	GR0_CK_KG[1]	GR0_CK_KG[0]
		GR0_CK_KB[7]	GR0_CK_KB[6]	GR0_CK_KB[5]	GR0_CK_KB[4]	GR0_CK_KB[3]	GR0_CK_KB[2]	GR0_CK_KB[1]	GR0_CK_KB[0]
		GR0_CK_KR[7]	GR0_CK_KR[6]	GR0_CK_KR[5]	GR0_CK_KR[4]	GR0_CK_KR[3]	GR0_CK_KR[2]	GR0_CK_KR[1]	GR0_CK_KR[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Video display controller 5	GR0_AB9	GR0_CK_A[7]	GR0_CK_A[6]	GR0_CK_A[5]	GR0_CK_A[4]	GR0_CK_A[3]	GR0_CK_A[2]	GR0_CK_A[1]	GR0_CK_A[0]
		GR0_CK_G[7]	GR0_CK_G[6]	GR0_CK_G[5]	GR0_CK_G[4]	GR0_CK_G[3]	GR0_CK_G[2]	GR0_CK_G[1]	GR0_CK_G[0]
		GR0_CK_B[7]	GR0_CK_B[6]	GR0_CK_B[5]	GR0_CK_B[4]	GR0_CK_B[3]	GR0_CK_B[2]	GR0_CK_B[1]	GR0_CK_B[0]
		GR0_CK_R[7]	GR0_CK_R[6]	GR0_CK_R[5]	GR0_CK_R[4]	GR0_CK_R[3]	GR0_CK_R[2]	GR0_CK_R[1]	GR0_CK_R[0]
	GR0_AB10	GR0_A0[7]	GR0_A0[6]	GR0_A0[5]	GR0_A0[4]	GR0_A0[3]	GR0_A0[2]	GR0_A0[1]	GR0_A0[0]
		GR0_G0[7]	GR0_G0[6]	GR0_G0[5]	GR0_G0[4]	GR0_G0[3]	GR0_G0[2]	GR0_G0[1]	GR0_G0[0]
		GR0_B0[7]	GR0_B0[6]	GR0_B0[5]	GR0_B0[4]	GR0_B0[3]	GR0_B0[2]	GR0_B0[1]	GR0_B0[0]
		GR0_R0[7]	GR0_R0[6]	GR0_R0[5]	GR0_R0[4]	GR0_R0[3]	GR0_R0[2]	GR0_R0[1]	GR0_R0[0]
	GR0_AB11	GR0_A1[7]	GR0_A1[6]	GR0_A1[5]	GR0_A1[4]	GR0_A1[3]	GR0_A1[2]	GR0_A1[1]	GR0_A1[0]
		GR0_G1[7]	GR0_G1[6]	GR0_G1[5]	GR0_G1[4]	GR0_G1[3]	GR0_G1[2]	GR0_G1[1]	GR0_G1[0]
		GR0_B1[7]	GR0_B1[6]	GR0_B1[5]	GR0_B1[4]	GR0_B1[3]	GR0_B1[2]	GR0_B1[1]	GR0_B1[0]
		GR0_R1[7]	GR0_R1[6]	GR0_R1[5]	GR0_R1[4]	GR0_R1[3]	GR0_R1[2]	GR0_R1[1]	GR0_R1[0]
	GR0_BASE	-	-	-	-	-	-	-	-
		GR0_BASE_G[7]	GR0_BASE_G[6]	GR0_BASE_G[5]	GR0_BASE_G[4]	GR0_BASE_G[3]	GR0_BASE_G[2]	GR0_BASE_G[1]	GR0_BASE_G[0]
		GR0_BASE_B[7]	GR0_BASE_B[6]	GR0_BASE_B[5]	GR0_BASE_B[4]	GR0_BASE_B[3]	GR0_BASE_B[2]	GR0_BASE_B[1]	GR0_BASE_B[0]
		GR0_BASE_R[7]	GR0_BASE_R[6]	GR0_BASE_R[5]	GR0_BASE_R[4]	GR0_BASE_R[3]	GR0_BASE_R[2]	GR0_BASE_R[1]	GR0_BASE_R[0]
	GR0_CLUT	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	GR0_CLT_SEL
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	ADJ0_UPDATE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	ADJ0_VEN
	ADJ0_BKSTR_SET	-	-	-	-	-	-	-	BKSTR_ON
		BKSTR_ST[3]	BKSTR_ST[2]	BKSTR_ST[1]	BKSTR_ST[0]	BKSTR_D[3]	BKSTR_D[2]	BKSTR_D[1]	BKSTR_D[0]
		-	-	-	BKSTR_T1[4]	BKSTR_T1[3]	BKSTR_T1[2]	BKSTR_T1[1]	BKSTR_T1[0]
		-	-	-	BKSTR_T2[4]	BKSTR_T2[3]	BKSTR_T2[2]	BKSTR_T2[1]	BKSTR_T2[0]
	ADJ0_ENH_TIM1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	ENH_MD	-	-	-	ENH_DISP_ON
	ADJ0_ENH_TIM2	-	-	-	-	-	ENH_VS[10]	ENH_VS[9]	ENH_VS[8]
		ENH_VS[7]	ENH_VS[6]	ENH_VS[5]	ENH_VS[4]	ENH_VS[3]	ENH_VS[2]	ENH_VS[1]	ENH_VS[0]
		-	-	-	-	-	ENH_VW[10]	ENH_VW[9]	ENH_VW[8]
		ENH_VW[7]	ENH_VW[6]	ENH_VW[5]	ENH_VW[4]	ENH_VW[3]	ENH_VW[2]	ENH_VW[1]	ENH_VW[0]
	ADJ0_ENH_TIM3	-	-	-	-	-	ENH_HS[10]	ENH_HS[9]	ENH_HS[8]
		ENH_HS[7]	ENH_HS[6]	ENH_HS[5]	ENH_HS[4]	ENH_HS[3]	ENH_HS[2]	ENH_HS[1]	ENH_HS[0]
		-	-	-	-	-	ENH_HW[10]	ENH_HW[9]	ENH_HW[8]
		ENH_HW[7]	ENH_HW[6]	ENH_HW[5]	ENH_HW[4]	ENH_HW[3]	ENH_HW[2]	ENH_HW[1]	ENH_HW[0]
	ADJ0_ENH_SHP1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	SHP_H_ON
		-	-	-	-	-	-	-	-
		-	SHP_H1_CORE[6]	SHP_H1_CORE[5]	SHP_H1_CORE[4]	SHP_H1_CORE[3]	SHP_H1_CORE[2]	SHP_H1_CORE[1]	SHP_H1_CORE[0]
	ADJ0_ENH_SHP2	SHP_H1_CLIP_O[7]	SHP_H1_CLIP_O[6]	SHP_H1_CLIP_O[5]	SHP_H1_CLIP_O[4]	SHP_H1_CLIP_O[3]	SHP_H1_CLIP_O[2]	SHP_H1_CLIP_O[1]	SHP_H1_CLIP_O[0]
		SHP_H1_CLIP_U[7]	SHP_H1_CLIP_U[6]	SHP_H1_CLIP_U[5]	SHP_H1_CLIP_U[4]	SHP_H1_CLIP_U[3]	SHP_H1_CLIP_U[2]	SHP_H1_CLIP_U[1]	SHP_H1_CLIP_U[0]
		SHP_H1_GAIN_O[7]	SHP_H1_GAIN_O[6]	SHP_H1_GAIN_O[5]	SHP_H1_GAIN_O[4]	SHP_H1_GAIN_O[3]	SHP_H1_GAIN_O[2]	SHP_H1_GAIN_O[1]	SHP_H1_GAIN_O[0]
		SHP_H1_GAIN_U[7]	SHP_H1_GAIN_U[6]	SHP_H1_GAIN_U[5]	SHP_H1_GAIN_U[4]	SHP_H1_GAIN_U[3]	SHP_H1_GAIN_U[2]	SHP_H1_GAIN_U[1]	SHP_H1_GAIN_U[0]
	ADJ0_ENH_SHP3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	SHP_H2_LPF_SEL
		-	-	-	-	-	-	-	-
		-	SHP_H2_CORE[6]	SHP_H2_CORE[5]	SHP_H2_CORE[4]	SHP_H2_CORE[3]	SHP_H2_CORE[2]	SHP_H2_CORE[1]	SHP_H2_CORE[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Video display controller 5	ADJ0_ENH_SHP4	SHP_H2_CLIP_O[7]	SHP_H2_CLIP_O[6]	SHP_H2_CLIP_O[5]	SHP_H2_CLIP_O[4]	SHP_H2_CLIP_O[3]	SHP_H2_CLIP_O[2]	SHP_H2_CLIP_O[1]	SHP_H2_CLIP_O[0]
		SHP_H2_CLIP_U[7]	SHP_H2_CLIP_U[6]	SHP_H2_CLIP_U[5]	SHP_H2_CLIP_U[4]	SHP_H2_CLIP_U[3]	SHP_H2_CLIP_U[2]	SHP_H2_CLIP_U[1]	SHP_H2_CLIP_U[0]
		SHP_H2_GAIN_O[7]	SHP_H2_GAIN_O[6]	SHP_H2_GAIN_O[5]	SHP_H2_GAIN_O[4]	SHP_H2_GAIN_O[3]	SHP_H2_GAIN_O[2]	SHP_H2_GAIN_O[1]	SHP_H2_GAIN_O[0]
		SHP_H2_GAIN_U[7]	SHP_H2_GAIN_U[6]	SHP_H2_GAIN_U[5]	SHP_H2_GAIN_U[4]	SHP_H2_GAIN_U[3]	SHP_H2_GAIN_U[2]	SHP_H2_GAIN_U[1]	SHP_H2_GAIN_U[0]
	ADJ0_ENH_SHP5	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	SHP_H3_CORE[6]	SHP_H3_CORE[5]	SHP_H3_CORE[4]	SHP_H3_CORE[3]	SHP_H3_CORE[2]	SHP_H3_CORE[1]	SHP_H3_CORE[0]
	ADJ0_ENH_SHP6	SHP_H3_CLIP_O[7]	SHP_H3_CLIP_O[6]	SHP_H3_CLIP_O[5]	SHP_H3_CLIP_O[4]	SHP_H3_CLIP_O[3]	SHP_H3_CLIP_O[2]	SHP_H3_CLIP_O[1]	SHP_H3_CLIP_O[0]
		SHP_H3_CLIP_U[7]	SHP_H3_CLIP_U[6]	SHP_H3_CLIP_U[5]	SHP_H3_CLIP_U[4]	SHP_H3_CLIP_U[3]	SHP_H3_CLIP_U[2]	SHP_H3_CLIP_U[1]	SHP_H3_CLIP_U[0]
		SHP_H3_GAIN_O[7]	SHP_H3_GAIN_O[6]	SHP_H3_GAIN_O[5]	SHP_H3_GAIN_O[4]	SHP_H3_GAIN_O[3]	SHP_H3_GAIN_O[2]	SHP_H3_GAIN_O[1]	SHP_H3_GAIN_O[0]
		SHP_H3_GAIN_U[7]	SHP_H3_GAIN_U[6]	SHP_H3_GAIN_U[5]	SHP_H3_GAIN_U[4]	SHP_H3_GAIN_U[3]	SHP_H3_GAIN_U[2]	SHP_H3_GAIN_U[1]	SHP_H3_GAIN_U[0]
	ADJ0_ENH_LTI1	LTI_H_ON	-	-	-	-	-	-	LTI_H2_LPF_SEL
		LTI_H2_INC_ZERO[7]	LTI_H2_INC_ZERO[6]	LTI_H2_INC_ZERO[5]	LTI_H2_INC_ZERO[4]	LTI_H2_INC_ZERO[3]	LTI_H2_INC_ZERO[2]	LTI_H2_INC_ZERO[1]	LTI_H2_INC_ZERO[0]
		LTI_H2_GAIN[7]	LTI_H2_GAIN[6]	LTI_H2_GAIN[5]	LTI_H2_GAIN[4]	LTI_H2_GAIN[3]	LTI_H2_GAIN[2]	LTI_H2_GAIN[1]	LTI_H2_GAIN[0]
		LTI_H2_CORE[7]	LTI_H2_CORE[6]	LTI_H2_CORE[5]	LTI_H2_CORE[4]	LTI_H2_CORE[3]	LTI_H2_CORE[2]	LTI_H2_CORE[1]	LTI_H2_CORE[0]
	ADJ0_ENH_LTI2	-	-	-	-	-	-	-	LTI_H4_MEDIAN_TAP_SEL
		LTI_H4_INC_ZERO[7]	LTI_H4_INC_ZERO[6]	LTI_H4_INC_ZERO[5]	LTI_H4_INC_ZERO[4]	LTI_H4_INC_ZERO[3]	LTI_H4_INC_ZERO[2]	LTI_H4_INC_ZERO[1]	LTI_H4_INC_ZERO[0]
		LTI_H4_GAIN[7]	LTI_H4_GAIN[6]	LTI_H4_GAIN[5]	LTI_H4_GAIN[4]	LTI_H4_GAIN[3]	LTI_H4_GAIN[2]	LTI_H4_GAIN[1]	LTI_H4_GAIN[0]
		LTI_H4_CORE[7]	LTI_H4_CORE[6]	LTI_H4_CORE[5]	LTI_H4_CORE[4]	LTI_H4_CORE[3]	LTI_H4_CORE[2]	LTI_H4_CORE[1]	LTI_H4_CORE[0]
	ADJ0_MTX_MODE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	ADJ0_MTX_MD[1]	ADJ0_MTX_MD[0]
	ADJ0_MTX_YG_ADJ0	-	-	-	-	-	-	-	-
		ADJ0_MTX_YG[7]	ADJ0_MTX_YG[6]	ADJ0_MTX_YG[5]	ADJ0_MTX_YG[4]	ADJ0_MTX_YG[3]	ADJ0_MTX_YG[2]	ADJ0_MTX_YG[1]	ADJ0_MTX_YG[0]
		-	-	-	-	-	ADJ0_MTX_GG[10]	ADJ0_MTX_GG[9]	ADJ0_MTX_GG[8]
		ADJ0_MTX_GG[7]	ADJ0_MTX_GG[6]	ADJ0_MTX_GG[5]	ADJ0_MTX_GG[4]	ADJ0_MTX_GG[3]	ADJ0_MTX_GG[2]	ADJ0_MTX_GG[1]	ADJ0_MTX_GG[0]
	ADJ0_MTX_YG_ADJ1	-	-	-	-	-	ADJ0_MTX_GB[10]	ADJ0_MTX_GB[9]	ADJ0_MTX_GB[8]
		ADJ0_MTX_GB[7]	ADJ0_MTX_GB[6]	ADJ0_MTX_GB[5]	ADJ0_MTX_GB[4]	ADJ0_MTX_GB[3]	ADJ0_MTX_GB[2]	ADJ0_MTX_GB[1]	ADJ0_MTX_GB[0]
		-	-	-	-	-	ADJ0_MTX_GR[10]	ADJ0_MTX_GR[9]	ADJ0_MTX_GR[8]
		ADJ0_MTX_GR[7]	ADJ0_MTX_GR[6]	ADJ0_MTX_GR[5]	ADJ0_MTX_GR[4]	ADJ0_MTX_GR[3]	ADJ0_MTX_GR[2]	ADJ0_MTX_GR[1]	ADJ0_MTX_GR[0]
	ADJ0_MTX_CBB_ADJ0	-	-	-	-	-	-	-	-
		ADJ0_MTX_B[7]	ADJ0_MTX_B[6]	ADJ0_MTX_B[5]	ADJ0_MTX_B[4]	ADJ0_MTX_B[3]	ADJ0_MTX_B[2]	ADJ0_MTX_B[1]	ADJ0_MTX_B[0]
		-	-	-	-	-	ADJ0_MTX_BG[10]	ADJ0_MTX_BG[9]	ADJ0_MTX_BG[8]
		ADJ0_MTX_BG[7]	ADJ0_MTX_BG[6]	ADJ0_MTX_BG[5]	ADJ0_MTX_BG[4]	ADJ0_MTX_BG[3]	ADJ0_MTX_BG[2]	ADJ0_MTX_BG[1]	ADJ0_MTX_BG[0]
	ADJ0_MTX_CBB_ADJ1	-	-	-	-	-	ADJ0_MTX_BB[10]	ADJ0_MTX_BB[9]	ADJ0_MTX_BB[8]
		ADJ0_MTX_BB[7]	ADJ0_MTX_BB[6]	ADJ0_MTX_BB[5]	ADJ0_MTX_BB[4]	ADJ0_MTX_BB[3]	ADJ0_MTX_BB[2]	ADJ0_MTX_BB[1]	ADJ0_MTX_BB[0]
		-	-	-	-	-	ADJ0_MTX_BR[10]	ADJ0_MTX_BR[9]	ADJ0_MTX_BR[8]
		ADJ0_MTX_BR[7]	ADJ0_MTX_BR[6]	ADJ0_MTX_BR[5]	ADJ0_MTX_BR[4]	ADJ0_MTX_BR[3]	ADJ0_MTX_BR[2]	ADJ0_MTX_BR[1]	ADJ0_MTX_BR[0]
	ADJ0_MTX_CRR_ADJ0	-	-	-	-	-	-	-	-
		ADJ0_MTX_R[7]	ADJ0_MTX_R[6]	ADJ0_MTX_R[5]	ADJ0_MTX_R[4]	ADJ0_MTX_R[3]	ADJ0_MTX_R[2]	ADJ0_MTX_R[1]	ADJ0_MTX_R[0]
		-	-	-	-	-	ADJ0_MTX_RG[10]	ADJ0_MTX_RG[9]	ADJ0_MTX_RG[8]
		ADJ0_MTX_RG[7]	ADJ0_MTX_RG[6]	ADJ0_MTX_RG[5]	ADJ0_MTX_RG[4]	ADJ0_MTX_RG[3]	ADJ0_MTX_RG[2]	ADJ0_MTX_RG[1]	ADJ0_MTX_RG[0]
	ADJ0_MTX_CRR_ADJ1	-	-	-	-	-	ADJ0_MTX_RB[10]	ADJ0_MTX_RB[9]	ADJ0_MTX_RB[8]
		ADJ0_MTX_RB[7]	ADJ0_MTX_RB[6]	ADJ0_MTX_RB[5]	ADJ0_MTX_RB[4]	ADJ0_MTX_RB[3]	ADJ0_MTX_RB[2]	ADJ0_MTX_RB[1]	ADJ0_MTX_RB[0]
		-	-	-	-	-	ADJ0_MTX_RR[10]	ADJ0_MTX_RR[9]	ADJ0_MTX_RR[8]
		ADJ0_MTX_RR[7]	ADJ0_MTX_RR[6]	ADJ0_MTX_RR[5]	ADJ0_MTX_RR[4]	ADJ0_MTX_RR[3]	ADJ0_MTX_RR[2]	ADJ0_MTX_RR[1]	ADJ0_MTX_RR[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Video display controller 5	GR2_UPDATE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	GR2_UPDATE
		-	-	-	GR2_P_VEN	-	-	-	GR2_IBUS_VEN
	GR2_FLM_RD	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	GR2_R_ENB
	GR2_FLM1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	GR2_LN_OFF_DIR
		-	-	-	-	-	-	GR2_FLM_SEL[1]	GR2_FLM_SEL[0]
		-	-	-	-	-	-	-	GR2_BST_MD
	GR2_FLM2	GR2_BASE[31]	GR2_BASE[30]	GR2_BASE[29]	GR2_BASE[28]	GR2_BASE[27]	GR2_BASE[26]	GR2_BASE[25]	GR2_BASE[24]
		GR2_BASE[23]	GR2_BASE[22]	GR2_BASE[21]	GR2_BASE[20]	GR2_BASE[19]	GR2_BASE[18]	GR2_BASE[17]	GR2_BASE[16]
		GR2_BASE[15]	GR2_BASE[14]	GR2_BASE[13]	GR2_BASE[12]	GR2_BASE[11]	GR2_BASE[10]	GR2_BASE[9]	GR2_BASE[8]
		GR2_BASE[7]	GR2_BASE[6]	GR2_BASE[5]	GR2_BASE[4]	GR2_BASE[3]	GR2_BASE[2]	GR2_BASE[1]	GR2_BASE[0]
	GR2_FLM3	-	GR2_LN_OFF[14]	GR2_LN_OFF[13]	GR2_LN_OFF[12]	GR2_LN_OFF[11]	GR2_LN_OFF[10]	GR2_LN_OFF[9]	GR2_LN_OFF[8]
		GR2_LN_OFF[7]	GR2_LN_OFF[6]	GR2_LN_OFF[5]	GR2_LN_OFF[4]	GR2_LN_OFF[3]	GR2_LN_OFF[2]	GR2_LN_OFF[1]	GR2_LN_OFF[0]
		-	-	-	-	-	-	GR2_FLM_NUM[9]	GR2_FLM_NUM[8]
		GR2_FLM_NUM[7]	GR2_FLM_NUM[6]	GR2_FLM_NUM[5]	GR2_FLM_NUM[4]	GR2_FLM_NUM[3]	GR2_FLM_NUM[2]	GR2_FLM_NUM[1]	GR2_FLM_NUM[0]
	GR2_FLM4	-	-	-	-	-	-	-	-
		-	GR2_FLM_OFF[22]	GR2_FLM_OFF[21]	GR2_FLM_OFF[20]	GR2_FLM_OFF[19]	GR2_FLM_OFF[18]	GR2_FLM_OFF[17]	GR2_FLM_OFF[16]
		GR2_FLM_OFF[15]	GR2_FLM_OFF[14]	GR2_FLM_OFF[13]	GR2_FLM_OFF[12]	GR2_FLM_OFF[11]	GR2_FLM_OFF[10]	GR2_FLM_OFF[9]	GR2_FLM_OFF[8]
		GR2_FLM_OFF[7]	GR2_FLM_OFF[6]	GR2_FLM_OFF[5]	GR2_FLM_OFF[4]	GR2_FLM_OFF[3]	GR2_FLM_OFF[2]	GR2_FLM_OFF[1]	GR2_FLM_OFF[0]
	GR2_FLM5	-	-	-	-	-	GR2_FLM_LNUM[10]	GR2_FLM_LNUM[9]	GR2_FLM_LNUM[8]
		GR2_FLM_LNUM[7]	GR2_FLM_LNUM[6]	GR2_FLM_LNUM[5]	GR2_FLM_LNUM[4]	GR2_FLM_LNUM[3]	GR2_FLM_LNUM[2]	GR2_FLM_LNUM[1]	GR2_FLM_LNUM[0]
		-	-	-	-	-	GR2_FLM_LOOP[10]	GR2_FLM_LOOP[9]	GR2_FLM_LOOP[8]
		GR2_FLM_LOOP[7]	GR2_FLM_LOOP[6]	GR2_FLM_LOOP[5]	GR2_FLM_LOOP[4]	GR2_FLM_LOOP[3]	GR2_FLM_LOOP[2]	GR2_FLM_LOOP[1]	GR2_FLM_LOOP[0]
	GR2_FLM6	GR2_FORMAT[3]	GR2_FORMAT[2]	GR2_FORMAT[1]	GR2_FORMAT[0]	-	GR2_HW[10]	GR2_HW[9]	GR2_HW[8]
		GR2_HW[7]	GR2_HW[6]	GR2_HW[5]	GR2_HW[4]	GR2_HW[3]	GR2_HW[2]	GR2_HW[1]	GR2_HW[0]
		-	-	-	GR2_RDSWA[2]	GR2_RDSWA[1]	GR2_RDSWA[0]	-	-
		-	-	GR2_STA_POS[5]	GR2_STA_POS[4]	GR2_STA_POS[3]	GR2_STA_POS[2]	GR2_STA_POS[1]	GR2_STA_POS[0]
	GR2_AB1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		GR2_ARC_MUL	GR2_ACALC_MD	-	GR2_ARC_ON	-	-	-	GR2_ARC_DISP_ON
		-	-	-	GR2_GRC_DISP_ON	-	-	GR2_DISP_SEL[1]	GR2_DISP_SEL[0]
	GR2_AB2	-	-	-	-	-	GR2_GRC_VS[10]	GR2_GRC_VS[9]	GR2_GRC_VS[8]
		GR2_GRC_VS[7]	GR2_GRC_VS[6]	GR2_GRC_VS[5]	GR2_GRC_VS[4]	GR2_GRC_VS[3]	GR2_GRC_VS[2]	GR2_GRC_VS[1]	GR2_GRC_VS[0]
		-	-	-	-	-	GR2_GRC_VW[10]	GR2_GRC_VW[9]	GR2_GRC_VW[8]
		GR2_GRC_VW[7]	GR2_GRC_VW[6]	GR2_GRC_VW[5]	GR2_GRC_VW[4]	GR2_GRC_VW[3]	GR2_GRC_VW[2]	GR2_GRC_VW[1]	GR2_GRC_VW[0]
	GR2_AB3	-	-	-	-	-	GR2_GRC_HS[10]	GR2_GRC_HS[9]	GR2_GRC_HS[8]
		GR2_GRC_HS[7]	GR2_GRC_HS[6]	GR2_GRC_HS[5]	GR2_GRC_HS[4]	GR2_GRC_HS[3]	GR2_GRC_HS[2]	GR2_GRC_HS[1]	GR2_GRC_HS[0]
		-	-	-	-	-	GR2_GRC_HW[10]	GR2_GRC_HW[9]	GR2_GRC_HW[8]
		GR2_GRC_HW[7]	GR2_GRC_HW[6]	GR2_GRC_HW[5]	GR2_GRC_HW[4]	GR2_GRC_HW[3]	GR2_GRC_HW[2]	GR2_GRC_HW[1]	GR2_GRC_HW[0]
	GR2_AB4	-	-	-	-	-	GR2_ARC_VS[10]	GR2_ARC_VS[9]	GR2_ARC_VS[8]
		GR2_ARC_VS[7]	GR2_ARC_VS[6]	GR2_ARC_VS[5]	GR2_ARC_VS[4]	GR2_ARC_VS[3]	GR2_ARC_VS[2]	GR2_ARC_VS[1]	GR2_ARC_VS[0]
		-	-	-	-	-	GR2_ARC_VW[10]	GR2_ARC_VW[9]	GR2_ARC_VW[8]
		GR2_ARC_VW[7]	GR2_ARC_VW[6]	GR2_ARC_VW[5]	GR2_ARC_VW[4]	GR2_ARC_VW[3]	GR2_ARC_VW[2]	GR2_ARC_VW[1]	GR2_ARC_VW[0]
	GR2_AB5	-	-	-	-	-	GR2_ARC_HS[10]	GR2_ARC_HS[9]	GR2_ARC_HS[8]
		GR2_ARC_HS[7]	GR2_ARC_HS[6]	GR2_ARC_HS[5]	GR2_ARC_HS[4]	GR2_ARC_HS[3]	GR2_ARC_HS[2]	GR2_ARC_HS[1]	GR2_ARC_HS[0]
		-	-	-	-	-	GR2_ARC_HW[10]	GR2_ARC_HW[9]	GR2_ARC_HW[8]
		GR2_ARC_HW[7]	GR2_ARC_HW[6]	GR2_ARC_HW[5]	GR2_ARC_HW[4]	GR2_ARC_HW[3]	GR2_ARC_HW[2]	GR2_ARC_HW[1]	GR2_ARC_HW[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Video display controller 5	GR2_AB6	-	-	-	-	-	-	-	GR2_ARC_MODE
		GR2_ARC_COEF[7]	GR2_ARC_COEF[6]	GR2_ARC_COEF[5]	GR2_ARC_COEF[4]	GR2_ARC_COEF[3]	GR2_ARC_COEF[2]	GR2_ARC_COEF[1]	GR2_ARC_COEF[0]
		-	-	-	-	-	-	-	-
		GR2_ARC_RATE[7]	GR2_ARC_RATE[6]	GR2_ARC_RATE[5]	GR2_ARC_RATE[4]	GR2_ARC_RATE[3]	GR2_ARC_RATE[2]	GR2_ARC_RATE[1]	GR2_ARC_RATE[0]
	GR2_AB7	-	-	-	-	-	-	-	-
		GR2_ARC_DEF[7]	GR2_ARC_DEF[6]	GR2_ARC_DEF[5]	GR2_ARC_DEF[4]	GR2_ARC_DEF[3]	GR2_ARC_DEF[2]	GR2_ARC_DEF[1]	GR2_ARC_DEF[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	GR2_CK_ON
	GR2_AB8	GR2_CK_KCLUT[7]	GR2_CK_KCLUT[6]	GR2_CK_KCLUT[5]	GR2_CK_KCLUT[4]	GR2_CK_KCLUT[3]	GR2_CK_KCLUT[2]	GR2_CK_KCLUT[1]	GR2_CK_KCLUT[0]
		GR2_CK_KG[7]	GR2_CK_KG[6]	GR2_CK_KG[5]	GR2_CK_KG[4]	GR2_CK_KG[3]	GR2_CK_KG[2]	GR2_CK_KG[1]	GR2_CK_KG[0]
		GR2_CK_KB[7]	GR2_CK_KB[6]	GR2_CK_KB[5]	GR2_CK_KB[4]	GR2_CK_KB[3]	GR2_CK_KB[2]	GR2_CK_KB[1]	GR2_CK_KB[0]
		GR2_CK_KR[7]	GR2_CK_KR[6]	GR2_CK_KR[5]	GR2_CK_KR[4]	GR2_CK_KR[3]	GR2_CK_KR[2]	GR2_CK_KR[1]	GR2_CK_KR[0]
	GR2_AB9	GR2_CK_A[7]	GR2_CK_A[6]	GR2_CK_A[5]	GR2_CK_A[4]	GR2_CK_A[3]	GR2_CK_A[2]	GR2_CK_A[1]	GR2_CK_A[0]
		GR2_CK_G[7]	GR2_CK_G[6]	GR2_CK_G[5]	GR2_CK_G[4]	GR2_CK_G[3]	GR2_CK_G[2]	GR2_CK_G[1]	GR2_CK_G[0]
		GR2_CK_B[7]	GR2_CK_B[6]	GR2_CK_B[5]	GR2_CK_B[4]	GR2_CK_B[3]	GR2_CK_B[2]	GR2_CK_B[1]	GR2_CK_B[0]
		GR2_CK_R[7]	GR2_CK_R[6]	GR2_CK_R[5]	GR2_CK_R[4]	GR2_CK_R[3]	GR2_CK_R[2]	GR2_CK_R[1]	GR2_CK_R[0]
	GR2_AB10	GR2_A0[7]	GR2_A0[6]	GR2_A0[5]	GR2_A0[4]	GR2_A0[3]	GR2_A0[2]	GR2_A0[1]	GR2_A0[0]
		GR2_G0[7]	GR2_G0[6]	GR2_G0[5]	GR2_G0[4]	GR2_G0[3]	GR2_G0[2]	GR2_G0[1]	GR2_G0[0]
		GR2_B0[7]	GR2_B0[6]	GR2_B0[5]	GR2_B0[4]	GR2_B0[3]	GR2_B0[2]	GR2_B0[1]	GR2_B0[0]
		GR2_R0[7]	GR2_R0[6]	GR2_R0[5]	GR2_R0[4]	GR2_R0[3]	GR2_R0[2]	GR2_R0[1]	GR2_R0[0]
	GR2_AB11	GR2_A1[7]	GR2_A1[6]	GR2_A1[5]	GR2_A1[4]	GR2_A1[3]	GR2_A1[2]	GR2_A1[1]	GR2_A1[0]
		GR2_G1[7]	GR2_G1[6]	GR2_G1[5]	GR2_G1[4]	GR2_G1[3]	GR2_G1[2]	GR2_G1[1]	GR2_G1[0]
		GR2_B1[7]	GR2_B1[6]	GR2_B1[5]	GR2_B1[4]	GR2_B1[3]	GR2_B1[2]	GR2_B1[1]	GR2_B1[0]
		GR2_R1[7]	GR2_R1[6]	GR2_R1[5]	GR2_R1[4]	GR2_R1[3]	GR2_R1[2]	GR2_R1[1]	GR2_R1[0]
	GR2_BASE	-	-	-	-	-	-	-	-
		GR2_BASE_G[7]	GR2_BASE_G[6]	GR2_BASE_G[5]	GR2_BASE_G[4]	GR2_BASE_G[3]	GR2_BASE_G[2]	GR2_BASE_G[1]	GR2_BASE_G[0]
		GR2_BASE_B[7]	GR2_BASE_B[6]	GR2_BASE_B[5]	GR2_BASE_B[4]	GR2_BASE_B[3]	GR2_BASE_B[2]	GR2_BASE_B[1]	GR2_BASE_B[0]
		GR2_BASE_R[7]	GR2_BASE_R[6]	GR2_BASE_R[5]	GR2_BASE_R[4]	GR2_BASE_R[3]	GR2_BASE_R[2]	GR2_BASE_R[1]	GR2_BASE_R[0]
	GR2_CLUT	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	GR2_CLT_SEL
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	GR2_MON	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	GR2_ARC_ST
	GR3_UPDATE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	GR3_UPDATE
		-	-	-	GR3_P_VEN	-	-	-	GR3_IBUS_VEN
	GR3_FLM_RD	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	GR3_R_ENB
	GR3_FLM1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	GR3_LN_OFF_DIR
		-	-	-	-	-	-	GR3_FLM_SEL[1]	GR3_FLM_SEL[0]
		-	-	-	-	-	-	-	GR3_BST_MD
	GR3_FLM2	GR3_BASE[31]	GR3_BASE[30]	GR3_BASE[29]	GR3_BASE[28]	GR3_BASE[27]	GR3_BASE[26]	GR3_BASE[25]	GR3_BASE[24]
		GR3_BASE[23]	GR3_BASE[22]	GR3_BASE[21]	GR3_BASE[20]	GR3_BASE[19]	GR3_BASE[18]	GR3_BASE[17]	GR3_BASE[16]
		GR3_BASE[15]	GR3_BASE[14]	GR3_BASE[13]	GR3_BASE[12]	GR3_BASE[11]	GR3_BASE[10]	GR3_BASE[9]	GR3_BASE[8]
		GR3_BASE[7]	GR3_BASE[6]	GR3_BASE[5]	GR3_BASE[4]	GR3_BASE[3]	GR3_BASE[2]	GR3_BASE[1]	GR3_BASE[0]
	GR3_FLM3	-	GR3_LN_OFF[14]	GR3_LN_OFF[13]	GR3_LN_OFF[12]	GR3_LN_OFF[11]	GR3_LN_OFF[10]	GR3_LN_OFF[9]	GR3_LN_OFF[8]
		GR3_LN_OFF[7]	GR3_LN_OFF[6]	GR3_LN_OFF[5]	GR3_LN_OFF[4]	GR3_LN_OFF[3]	GR3_LN_OFF[2]	GR3_LN_OFF[1]	GR3_LN_OFF[0]
		-	-	-	-	-	-	GR3_FLM_NUM[9]	GR3_FLM_NUM[8]
		GR3_FLM_NUM[7]	GR3_FLM_NUM[6]	GR3_FLM_NUM[5]	GR3_FLM_NUM[4]	GR3_FLM_NUM[3]	GR3_FLM_NUM[2]	GR3_FLM_NUM[1]	GR3_FLM_NUM[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Video display controller 5	GR3_FLM4	-	-	-	-	-	-	-	-
		-	GR3_FLM_OFF [22]	GR3_FLM_OFF [21]	GR3_FLM_OFF [20]	GR3_FLM_OFF [19]	GR3_FLM_OFF [18]	GR3_FLM_OFF [17]	GR3_FLM_OFF [16]
		GR3_FLM_OFF [15]	GR3_FLM_OFF [14]	GR3_FLM_OFF [13]	GR3_FLM_OFF [12]	GR3_FLM_OFF [11]	GR3_FLM_OFF [10]	GR3_FLM_OFF[9]	GR3_FLM_OFF[8]
		GR3_FLM_OFF[7]	GR3_FLM_OFF[6]	GR3_FLM_OFF[5]	GR3_FLM_OFF[4]	GR3_FLM_OFF[3]	GR3_FLM_OFF[2]	GR3_FLM_OFF[1]	GR3_FLM_OFF[0]
	GR3_FLM5	-	-	-	-	-	GR3_FLM_LNUM[10]	GR3_FLM_LNUM[9]	GR3_FLM_LNUM[8]
		GR3_FLM_LNUM[7]	GR3_FLM_LNUM[6]	GR3_FLM_LNUM[5]	GR3_FLM_LNUM[4]	GR3_FLM_LNUM[3]	GR3_FLM_LNUM[2]	GR3_FLM_LNUM[1]	GR3_FLM_LNUM[0]
		-	-	-	-	-	GR3_FLM_LOOP [10]	GR3_FLM_LOOP [9]	GR3_FLM_LOOP [8]
		GR3_FLM_LOOP [7]	GR3_FLM_LOOP [6]	GR3_FLM_LOOP [5]	GR3_FLM_LOOP [4]	GR3_FLM_LOOP [3]	GR3_FLM_LOOP [2]	GR3_FLM_LOOP [1]	GR3_FLM_LOOP [0]
	GR3_FLM6	GR3_FORMAT[3]	GR3_FORMAT[2]	GR3_FORMAT[1]	GR3_FORMAT[0]	-	GR3_HW[10]	GR3_HW[9]	GR3_HW[8]
		GR3_HW[7]	GR3_HW[6]	GR3_HW[5]	GR3_HW[4]	GR3_HW[3]	GR3_HW[2]	GR3_HW[1]	GR3_HW[0]
		-	-	-	GR3_RDSWA[2]	GR3_RDSWA[1]	GR3_RDSWA[0]	-	-
		-	-	GR3_STA_POS[5]	GR3_STA_POS[4]	GR3_STA_POS[3]	GR3_STA_POS[2]	GR3_STA_POS[1]	GR3_STA_POS[0]
	GR3_AB1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		GR3_ARC_MUL	GR3_ACALC_MD	-	GR3_ARC_ON	-	-	-	GR3_ARC_DISP_ON
		-	-	-	GR3_GRC_DISP_ON	-	-	GR3_DISP_SEL[1]	GR3_DISP_SEL[0]
	GR3_AB2	-	-	-	-	-	GR3_GRC_VS[10]	GR3_GRC_VS[9]	GR3_GRC_VS[8]
		GR3_GRC_VS[7]	GR3_GRC_VS[6]	GR3_GRC_VS[5]	GR3_GRC_VS[4]	GR3_GRC_VS[3]	GR3_GRC_VS[2]	GR3_GRC_VS[1]	GR3_GRC_VS[0]
		-	-	-	-	-	GR3_GRC_VW [10]	GR3_GRC_VW[9]	GR3_GRC_VW[8]
		GR3_GRC_VW[7]	GR3_GRC_VW[6]	GR3_GRC_VW[5]	GR3_GRC_VW[4]	GR3_GRC_VW[3]	GR3_GRC_VW[2]	GR3_GRC_VW[1]	GR3_GRC_VW[0]
	GR3_AB3	-	-	-	-	-	GR3_GRC_HS[10]	GR3_GRC_HS[9]	GR3_GRC_HS[8]
		GR3_GRC_HS[7]	GR3_GRC_HS[6]	GR3_GRC_HS[5]	GR3_GRC_HS[4]	GR3_GRC_HS[3]	GR3_GRC_HS[2]	GR3_GRC_HS[1]	GR3_GRC_HS[0]
		-	-	-	-	-	GR3_GRC_HW [10]	GR3_GRC_HW[9]	GR3_GRC_HW[8]
		GR3_GRC_HW[7]	GR3_GRC_HW[6]	GR3_GRC_HW[5]	GR3_GRC_HW[4]	GR3_GRC_HW[3]	GR3_GRC_HW[2]	GR3_GRC_HW[1]	GR3_GRC_HW[0]
	GR3_AB4	-	-	-	-	-	GR3_ARC_VS[10]	GR3_ARC_VS[9]	GR3_ARC_VS[8]
		GR3_ARC_VS[7]	GR3_ARC_VS[6]	GR3_ARC_VS[5]	GR3_ARC_VS[4]	GR3_ARC_VS[3]	GR3_ARC_VS[2]	GR3_ARC_VS[1]	GR3_ARC_VS[0]
		-	-	-	-	-	GR3_ARC_VW [10]	GR3_ARC_VW[9]	GR3_ARC_VW[8]
		GR3_ARC_VW[7]	GR3_ARC_VW[6]	GR3_ARC_VW[5]	GR3_ARC_VW[4]	GR3_ARC_VW[3]	GR3_ARC_VW[2]	GR3_ARC_VW[1]	GR3_ARC_VW[0]
	GR3_AB5	-	-	-	-	-	GR3_ARC_HS[10]	GR3_ARC_HS[9]	GR3_ARC_HS[8]
		GR3_ARC_HS[7]	GR3_ARC_HS[6]	GR3_ARC_HS[5]	GR3_ARC_HS[4]	GR3_ARC_HS[3]	GR3_ARC_HS[2]	GR3_ARC_HS[1]	GR3_ARC_HS[0]
		-	-	-	-	-	GR3_ARC_HW [10]	GR3_ARC_HW[9]	GR3_ARC_HW[8]
		GR3_ARC_HW[7]	GR3_ARC_HW[6]	GR3_ARC_HW[5]	GR3_ARC_HW[4]	GR3_ARC_HW[3]	GR3_ARC_HW[2]	GR3_ARC_HW[1]	GR3_ARC_HW[0]
	GR3_AB6	-	-	-	-	-	-	-	GR3_ARC_ MODE
		GR3_ARC_COEF[7]	GR3_ARC_COEF[6]	GR3_ARC_COEF[5]	GR3_ARC_COEF[4]	GR3_ARC_COEF[3]	GR3_ARC_COEF[2]	GR3_ARC_COEF[1]	GR3_ARC_COEF[0]
		-	-	-	-	-	-	-	-
		GR3_ARC_RATE [7]	GR3_ARC_RATE [6]	GR3_ARC_RATE [5]	GR3_ARC_RATE [4]	GR3_ARC_RATE [3]	GR3_ARC_RATE [2]	GR3_ARC_RATE [1]	GR3_ARC_RATE [0]
	GR3_AB7	-	-	-	-	-	-	-	-
		GR3_ARC_DEF[7]	GR3_ARC_DEF[6]	GR3_ARC_DEF[5]	GR3_ARC_DEF[4]	GR3_ARC_DEF[3]	GR3_ARC_DEF[2]	GR3_ARC_DEF[1]	GR3_ARC_DEF[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	GR3_CK_ON
	GR3_AB8	GR3_CK_KCLUT [7]	GR3_CK_KCLUT [6]	GR3_CK_KCLUT [5]	GR3_CK_KCLUT [4]	GR3_CK_KCLUT [3]	GR3_CK_KCLUT [2]	GR3_CK_KCLUT [1]	GR3_CK_KCLUT [0]
		GR3_CK_KG[7]	GR3_CK_KG[6]	GR3_CK_KG[5]	GR3_CK_KG[4]	GR3_CK_KG[3]	GR3_CK_KG[2]	GR3_CK_KG[1]	GR3_CK_KG[0]
		GR3_CK_KB[7]	GR3_CK_KB[6]	GR3_CK_KB[5]	GR3_CK_KB[4]	GR3_CK_KB[3]	GR3_CK_KB[2]	GR3_CK_KB[1]	GR3_CK_KB[0]
		GR3_CK_KR[7]	GR3_CK_KR[6]	GR3_CK_KR[5]	GR3_CK_KR[4]	GR3_CK_KR[3]	GR3_CK_KR[2]	GR3_CK_KR[1]	GR3_CK_KR[0]
	GR3_AB9	GR3_CK_A[7]	GR3_CK_A[6]	GR3_CK_A[5]	GR3_CK_A[4]	GR3_CK_A[3]	GR3_CK_A[2]	GR3_CK_A[1]	GR3_CK_A[0]
		GR3_CK_G[7]	GR3_CK_G[6]	GR3_CK_G[5]	GR3_CK_G[4]	GR3_CK_G[3]	GR3_CK_G[2]	GR3_CK_G[1]	GR3_CK_G[0]
		GR3_CK_B[7]	GR3_CK_B[6]	GR3_CK_B[5]	GR3_CK_B[4]	GR3_CK_B[3]	GR3_CK_B[2]	GR3_CK_B[1]	GR3_CK_B[0]
		GR3_CK_R[7]	GR3_CK_R[6]	GR3_CK_R[5]	GR3_CK_R[4]	GR3_CK_R[3]	GR3_CK_R[2]	GR3_CK_R[1]	GR3_CK_R[0]
	GR3_AB10	GR3_A0[7]	GR3_A0[6]	GR3_A0[5]	GR3_A0[4]	GR3_A0[3]	GR3_A0[2]	GR3_A0[1]	GR3_A0[0]
		GR3_G0[7]	GR3_G0[6]	GR3_G0[5]	GR3_G0[4]	GR3_G0[3]	GR3_G0[2]	GR3_G0[1]	GR3_G0[0]
		GR3_B0[7]	GR3_B0[6]	GR3_B0[5]	GR3_B0[4]	GR3_B0[3]	GR3_B0[2]	GR3_B0[1]	GR3_B0[0]
		GR3_R0[7]	GR3_R0[6]	GR3_R0[5]	GR3_R0[4]	GR3_R0[3]	GR3_R0[2]	GR3_R0[1]	GR3_R0[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Video display controller 5	GR3_AB11	GR3_A1[7]	GR3_A1[6]	GR3_A1[5]	GR3_A1[4]	GR3_A1[3]	GR3_A1[2]	GR3_A1[1]	GR3_A1[0]
		GR3_G1[7]	GR3_G1[6]	GR3_G1[5]	GR3_G1[4]	GR3_G1[3]	GR3_G1[2]	GR3_G1[1]	GR3_G1[0]
		GR3_B1[7]	GR3_B1[6]	GR3_B1[5]	GR3_B1[4]	GR3_B1[3]	GR3_B1[2]	GR3_B1[1]	GR3_B1[0]
		GR3_R1[7]	GR3_R1[6]	GR3_R1[5]	GR3_R1[4]	GR3_R1[3]	GR3_R1[2]	GR3_R1[1]	GR3_R1[0]
	GR3_BASE	-	-	-	-	-	-	-	-
		GR3_BASE_G[7]	GR3_BASE_G[6]	GR3_BASE_G[5]	GR3_BASE_G[4]	GR3_BASE_G[3]	GR3_BASE_G[2]	GR3_BASE_G[1]	GR3_BASE_G[0]
		GR3_BASE_B[7]	GR3_BASE_B[6]	GR3_BASE_B[5]	GR3_BASE_B[4]	GR3_BASE_B[3]	GR3_BASE_B[2]	GR3_BASE_B[1]	GR3_BASE_B[0]
		GR3_BASE_R[7]	GR3_BASE_R[6]	GR3_BASE_R[5]	GR3_BASE_R[4]	GR3_BASE_R[3]	GR3_BASE_R[2]	GR3_BASE_R[1]	GR3_BASE_R[0]
	GR3_CLUT_INT	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	GR3_CLT_SEL
		-	-	-	-	-	GR3_LINE[10]	GR3_LINE[9]	GR3_LINE[8]
		GR3_LINE[7]	GR3_LINE[6]	GR3_LINE[5]	GR3_LINE[4]	GR3_LINE[3]	GR3_LINE[2]	GR3_LINE[1]	GR3_LINE[0]
	GR3_MON	-	-	-	-	-	GR3_LIN_STAT [10]	GR3_LIN_STAT[9]	GR3_LIN_STAT[8]
		GR3_LIN_STAT[7]	GR3_LIN_STAT[6]	GR3_LIN_STAT[5]	GR3_LIN_STAT[4]	GR3_LIN_STAT[3]	GR3_LIN_STAT[2]	GR3_LIN_STAT[1]	GR3_LIN_STAT[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	GR3_ARC_ST
	GR_VIN_UPDATE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	GR_VIN_ UPDATE
		-	-	-	GR_VIN_P_VEN	-	-	-	-
	GR_VIN_AB1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	GR_VIN_SCL_ UND_SEL	GR_VIN_DISP_ SEL[1]	GR_VIN_DISP_ SEL[0]
	GAM_G_UPDATE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	GAM_G_VEN
	GAM_SW	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	GAM_ON
	GAM_G_LUT1	-	-	-	-	-	GAM_G_GAIN_00 [10]	GAM_G_GAIN_00 [9]	GAM_G_GAIN_00 [8]
		GAM_G_GAIN_00 [7]	GAM_G_GAIN_00 [6]	GAM_G_GAIN_00 [5]	GAM_G_GAIN_00 [4]	GAM_G_GAIN_00 [3]	GAM_G_GAIN_00 [2]	GAM_G_GAIN_00 [1]	GAM_G_GAIN_00 [0]
		-	-	-	-	-	GAM_G_GAIN_01 [10]	GAM_G_GAIN_01 [9]	GAM_G_GAIN_01 [8]
		GAM_G_GAIN_01 [7]	GAM_G_GAIN_01 [6]	GAM_G_GAIN_01 [5]	GAM_G_GAIN_01 [4]	GAM_G_GAIN_01 [3]	GAM_G_GAIN_01 [2]	GAM_G_GAIN_01 [1]	GAM_G_GAIN_01 [0]
	GAM_G_LUT2	-	-	-	-	-	GAM_G_GAIN_02 [10]	GAM_G_GAIN_02 [9]	GAM_G_GAIN_02 [8]
		GAM_G_GAIN_02 [7]	GAM_G_GAIN_02 [6]	GAM_G_GAIN_02 [5]	GAM_G_GAIN_02 [4]	GAM_G_GAIN_02 [3]	GAM_G_GAIN_02 [2]	GAM_G_GAIN_02 [1]	GAM_G_GAIN_02 [0]
		-	-	-	-	-	GAM_G_GAIN_03 [10]	GAM_G_GAIN_03 [9]	GAM_G_GAIN_03 [8]
		GAM_G_GAIN_03 [7]	GAM_G_GAIN_03 [6]	GAM_G_GAIN_03 [5]	GAM_G_GAIN_03 [4]	GAM_G_GAIN_03 [3]	GAM_G_GAIN_03 [2]	GAM_G_GAIN_03 [1]	GAM_G_GAIN_03 [0]
	GAM_G_LUT3	-	-	-	-	-	GAM_G_GAIN_04 [10]	GAM_G_GAIN_04 [9]	GAM_G_GAIN_04 [8]
		GAM_G_GAIN_04 [7]	GAM_G_GAIN_04 [6]	GAM_G_GAIN_04 [5]	GAM_G_GAIN_04 [4]	GAM_G_GAIN_04 [3]	GAM_G_GAIN_04 [2]	GAM_G_GAIN_04 [1]	GAM_G_GAIN_04 [0]
		-	-	-	-	-	GAM_G_GAIN_05 [10]	GAM_G_GAIN_05 [9]	GAM_G_GAIN_05 [8]
		GAM_G_GAIN_05 [7]	GAM_G_GAIN_05 [6]	GAM_G_GAIN_05 [5]	GAM_G_GAIN_05 [4]	GAM_G_GAIN_05 [3]	GAM_G_GAIN_05 [2]	GAM_G_GAIN_05 [1]	GAM_G_GAIN_05 [0]
	GAM_G_LUT4	-	-	-	-	-	GAM_G_GAIN_06 [10]	GAM_G_GAIN_06 [9]	GAM_G_GAIN_06 [8]
		GAM_G_GAIN_06 [7]	GAM_G_GAIN_06 [6]	GAM_G_GAIN_06 [5]	GAM_G_GAIN_06 [4]	GAM_G_GAIN_06 [3]	GAM_G_GAIN_06 [2]	GAM_G_GAIN_06 [1]	GAM_G_GAIN_06 [0]
		-	-	-	-	-	GAM_G_GAIN_07 [10]	GAM_G_GAIN_07 [9]	GAM_G_GAIN_07 [8]
		GAM_G_GAIN_07 [7]	GAM_G_GAIN_07 [6]	GAM_G_GAIN_07 [5]	GAM_G_GAIN_07 [4]	GAM_G_GAIN_07 [3]	GAM_G_GAIN_07 [2]	GAM_G_GAIN_07 [1]	GAM_G_GAIN_07 [0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Video display controller 5	GAM_G_LUT5	-	-	-	-	-	GAM_G_GAIN_08 [10]	GAM_G_GAIN_08 [9]	GAM_G_GAIN_08 [8]
		GAM_G_GAIN_08 [7]	GAM_G_GAIN_08 [6]	GAM_G_GAIN_08 [5]	GAM_G_GAIN_08 [4]	GAM_G_GAIN_08 [3]	GAM_G_GAIN_08 [2]	GAM_G_GAIN_08 [1]	GAM_G_GAIN_08 [0]
		-	-	-	-	-	GAM_G_GAIN_09 [10]	GAM_G_GAIN_09 [9]	GAM_G_GAIN_09 [8]
		GAM_G_GAIN_09 [7]	GAM_G_GAIN_09 [6]	GAM_G_GAIN_09 [5]	GAM_G_GAIN_09 [4]	GAM_G_GAIN_09 [3]	GAM_G_GAIN_09 [2]	GAM_G_GAIN_09 [1]	GAM_G_GAIN_09 [0]
	GAM_G_LUT6	-	-	-	-	-	GAM_G_GAIN_10 [10]	GAM_G_GAIN_10 [9]	GAM_G_GAIN_10 [8]
		GAM_G_GAIN_10 [7]	GAM_G_GAIN_10 [6]	GAM_G_GAIN_10 [5]	GAM_G_GAIN_10 [4]	GAM_G_GAIN_10 [3]	GAM_G_GAIN_10 [2]	GAM_G_GAIN_10 [1]	GAM_G_GAIN_10 [0]
		-	-	-	-	-	GAM_G_GAIN_11 [10]	GAM_G_GAIN_11 [9]	GAM_G_GAIN_11 [8]
		GAM_G_GAIN_11 [7]	GAM_G_GAIN_11 [6]	GAM_G_GAIN_11 [5]	GAM_G_GAIN_11 [4]	GAM_G_GAIN_11 [3]	GAM_G_GAIN_11 [2]	GAM_G_GAIN_11 [1]	GAM_G_GAIN_11 [0]
	GAM_G_LUT7	-	-	-	-	-	GAM_G_GAIN_12 [10]	GAM_G_GAIN_12 [9]	GAM_G_GAIN_12 [8]
		GAM_G_GAIN_12 [7]	GAM_G_GAIN_12 [6]	GAM_G_GAIN_12 [5]	GAM_G_GAIN_12 [4]	GAM_G_GAIN_12 [3]	GAM_G_GAIN_12 [2]	GAM_G_GAIN_12 [1]	GAM_G_GAIN_12 [0]
		-	-	-	-	-	GAM_G_GAIN_13 [10]	GAM_G_GAIN_13 [9]	GAM_G_GAIN_13 [8]
		GAM_G_GAIN_13 [7]	GAM_G_GAIN_13 [6]	GAM_G_GAIN_13 [5]	GAM_G_GAIN_13 [4]	GAM_G_GAIN_13 [3]	GAM_G_GAIN_13 [2]	GAM_G_GAIN_13 [1]	GAM_G_GAIN_13 [0]
	GAM_G_LUT8	-	-	-	-	-	GAM_G_GAIN_14 [10]	GAM_G_GAIN_14 [9]	GAM_G_GAIN_14 [8]
		GAM_G_GAIN_14 [7]	GAM_G_GAIN_14 [6]	GAM_G_GAIN_14 [5]	GAM_G_GAIN_14 [4]	GAM_G_GAIN_14 [3]	GAM_G_GAIN_14 [2]	GAM_G_GAIN_14 [1]	GAM_G_GAIN_14 [0]
		-	-	-	-	-	GAM_G_GAIN_15 [10]	GAM_G_GAIN_15 [9]	GAM_G_GAIN_15 [8]
		GAM_G_GAIN_15 [7]	GAM_G_GAIN_15 [6]	GAM_G_GAIN_15 [5]	GAM_G_GAIN_15 [4]	GAM_G_GAIN_15 [3]	GAM_G_GAIN_15 [2]	GAM_G_GAIN_15 [1]	GAM_G_GAIN_15 [0]
	GAM_G_LUT9	-	-	-	-	-	GAM_G_GAIN_16 [10]	GAM_G_GAIN_16 [9]	GAM_G_GAIN_16 [8]
		GAM_G_GAIN_16 [7]	GAM_G_GAIN_16 [6]	GAM_G_GAIN_16 [5]	GAM_G_GAIN_16 [4]	GAM_G_GAIN_16 [3]	GAM_G_GAIN_16 [2]	GAM_G_GAIN_16 [1]	GAM_G_GAIN_16 [0]
		-	-	-	-	-	GAM_G_GAIN_17 [10]	GAM_G_GAIN_17 [9]	GAM_G_GAIN_17 [8]
		GAM_G_GAIN_17 [7]	GAM_G_GAIN_17 [6]	GAM_G_GAIN_17 [5]	GAM_G_GAIN_17 [4]	GAM_G_GAIN_17 [3]	GAM_G_GAIN_17 [2]	GAM_G_GAIN_17 [1]	GAM_G_GAIN_17 [0]
	GAM_G_LUT10	-	-	-	-	-	GAM_G_GAIN_18 [10]	GAM_G_GAIN_18 [9]	GAM_G_GAIN_18 [8]
		GAM_G_GAIN_18 [7]	GAM_G_GAIN_18 [6]	GAM_G_GAIN_18 [5]	GAM_G_GAIN_18 [4]	GAM_G_GAIN_18 [3]	GAM_G_GAIN_18 [2]	GAM_G_GAIN_18 [1]	GAM_G_GAIN_18 [0]
		-	-	-	-	-	GAM_G_GAIN_19 [10]	GAM_G_GAIN_19 [9]	GAM_G_GAIN_19 [8]
		GAM_G_GAIN_19 [7]	GAM_G_GAIN_19 [6]	GAM_G_GAIN_19 [5]	GAM_G_GAIN_19 [4]	GAM_G_GAIN_19 [3]	GAM_G_GAIN_19 [2]	GAM_G_GAIN_19 [1]	GAM_G_GAIN_19 [0]
	GAM_G_LUT11	-	-	-	-	-	GAM_G_GAIN_20 [10]	GAM_G_GAIN_20 [9]	GAM_G_GAIN_20 [8]
		GAM_G_GAIN_20 [7]	GAM_G_GAIN_20 [6]	GAM_G_GAIN_20 [5]	GAM_G_GAIN_20 [4]	GAM_G_GAIN_20 [3]	GAM_G_GAIN_20 [2]	GAM_G_GAIN_20 [1]	GAM_G_GAIN_20 [0]
		-	-	-	-	-	GAM_G_GAIN_21 [10]	GAM_G_GAIN_21 [9]	GAM_G_GAIN_21 [8]
		GAM_G_GAIN_21 [7]	GAM_G_GAIN_21 [6]	GAM_G_GAIN_21 [5]	GAM_G_GAIN_21 [4]	GAM_G_GAIN_21 [3]	GAM_G_GAIN_21 [2]	GAM_G_GAIN_21 [1]	GAM_G_GAIN_21 [0]
	GAM_G_LUT12	-	-	-	-	-	GAM_G_GAIN_22 [10]	GAM_G_GAIN_22 [9]	GAM_G_GAIN_22 [8]
		GAM_G_GAIN_22 [7]	GAM_G_GAIN_22 [6]	GAM_G_GAIN_22 [5]	GAM_G_GAIN_22 [4]	GAM_G_GAIN_22 [3]	GAM_G_GAIN_22 [2]	GAM_G_GAIN_22 [1]	GAM_G_GAIN_22 [0]
		-	-	-	-	-	GAM_G_GAIN_23 [10]	GAM_G_GAIN_23 [9]	GAM_G_GAIN_23 [8]
		GAM_G_GAIN_23 [7]	GAM_G_GAIN_23 [6]	GAM_G_GAIN_23 [5]	GAM_G_GAIN_23 [4]	GAM_G_GAIN_23 [3]	GAM_G_GAIN_23 [2]	GAM_G_GAIN_23 [1]	GAM_G_GAIN_23 [0]
	GAM_G_LUT13	-	-	-	-	-	GAM_G_GAIN_24 [10]	GAM_G_GAIN_24 [9]	GAM_G_GAIN_24 [8]
		GAM_G_GAIN_24 [7]	GAM_G_GAIN_24 [6]	GAM_G_GAIN_24 [5]	GAM_G_GAIN_24 [4]	GAM_G_GAIN_24 [3]	GAM_G_GAIN_24 [2]	GAM_G_GAIN_24 [1]	GAM_G_GAIN_24 [0]
		-	-	-	-	-	GAM_G_GAIN_25 [10]	GAM_G_GAIN_25 [9]	GAM_G_GAIN_25 [8]
		GAM_G_GAIN_25 [7]	GAM_G_GAIN_25 [6]	GAM_G_GAIN_25 [5]	GAM_G_GAIN_25 [4]	GAM_G_GAIN_25 [3]	GAM_G_GAIN_25 [2]	GAM_G_GAIN_25 [1]	GAM_G_GAIN_25 [0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Video display controller 5	GAM_G_LUT14	-	-	-	-	-	GAM_G_GAIN_26 [10]	GAM_G_GAIN_26 [9]	GAM_G_GAIN_26 [8]
		GAM_G_GAIN_26 [7]	GAM_G_GAIN_26 [6]	GAM_G_GAIN_26 [5]	GAM_G_GAIN_26 [4]	GAM_G_GAIN_26 [3]	GAM_G_GAIN_26 [2]	GAM_G_GAIN_26 [1]	GAM_G_GAIN_26 [0]
		-	-	-	-	-	GAM_G_GAIN_27 [10]	GAM_G_GAIN_27 [9]	GAM_G_GAIN_27 [8]
		GAM_G_GAIN_27 [7]	GAM_G_GAIN_27 [6]	GAM_G_GAIN_27 [5]	GAM_G_GAIN_27 [4]	GAM_G_GAIN_27 [3]	GAM_G_GAIN_27 [2]	GAM_G_GAIN_27 [1]	GAM_G_GAIN_27 [0]
	GAM_G_LUT15	-	-	-	-	-	GAM_G_GAIN_28 [10]	GAM_G_GAIN_28 [9]	GAM_G_GAIN_28 [8]
		GAM_G_GAIN_28 [7]	GAM_G_GAIN_28 [6]	GAM_G_GAIN_28 [5]	GAM_G_GAIN_28 [4]	GAM_G_GAIN_28 [3]	GAM_G_GAIN_28 [2]	GAM_G_GAIN_28 [1]	GAM_G_GAIN_28 [0]
		-	-	-	-	-	GAM_G_GAIN_29 [10]	GAM_G_GAIN_29 [9]	GAM_G_GAIN_29 [8]
		GAM_G_GAIN_29 [7]	GAM_G_GAIN_29 [6]	GAM_G_GAIN_29 [5]	GAM_G_GAIN_29 [4]	GAM_G_GAIN_29 [3]	GAM_G_GAIN_29 [2]	GAM_G_GAIN_29 [1]	GAM_G_GAIN_29 [0]
	GAM_G_LUT16	-	-	-	-	-	GAM_G_GAIN_30 [10]	GAM_G_GAIN_30 [9]	GAM_G_GAIN_30 [8]
		GAM_G_GAIN_30 [7]	GAM_G_GAIN_30 [6]	GAM_G_GAIN_30 [5]	GAM_G_GAIN_30 [4]	GAM_G_GAIN_30 [3]	GAM_G_GAIN_30 [2]	GAM_G_GAIN_30 [1]	GAM_G_GAIN_30 [0]
		-	-	-	-	-	GAM_G_GAIN_31 [10]	GAM_G_GAIN_31 [9]	GAM_G_GAIN_31 [8]
		GAM_G_GAIN_31 [7]	GAM_G_GAIN_31 [6]	GAM_G_GAIN_31 [5]	GAM_G_GAIN_31 [4]	GAM_G_GAIN_31 [3]	GAM_G_GAIN_31 [2]	GAM_G_GAIN_31 [1]	GAM_G_GAIN_31 [0]
	GAM_G_AREA1	-	-	-	-	-	-	-	-
		GAM_G_TH_01[7]	GAM_G_TH_01[6]	GAM_G_TH_01[5]	GAM_G_TH_01[4]	GAM_G_TH_01[3]	GAM_G_TH_01[2]	GAM_G_TH_01[1]	GAM_G_TH_01[0]
		GAM_G_TH_02[7]	GAM_G_TH_02[6]	GAM_G_TH_02[5]	GAM_G_TH_02[4]	GAM_G_TH_02[3]	GAM_G_TH_02[2]	GAM_G_TH_02[1]	GAM_G_TH_02[0]
		GAM_G_TH_03[7]	GAM_G_TH_03[6]	GAM_G_TH_03[5]	GAM_G_TH_03[4]	GAM_G_TH_03[3]	GAM_G_TH_03[2]	GAM_G_TH_03[1]	GAM_G_TH_03[0]
	GAM_G_AREA2	GAM_G_TH_04[7]	GAM_G_TH_04[6]	GAM_G_TH_04[5]	GAM_G_TH_04[4]	GAM_G_TH_04[3]	GAM_G_TH_04[2]	GAM_G_TH_04[1]	GAM_G_TH_04[0]
		GAM_G_TH_05[7]	GAM_G_TH_05[6]	GAM_G_TH_05[5]	GAM_G_TH_05[4]	GAM_G_TH_05[3]	GAM_G_TH_05[2]	GAM_G_TH_05[1]	GAM_G_TH_05[0]
		GAM_G_TH_06[7]	GAM_G_TH_06[6]	GAM_G_TH_06[5]	GAM_G_TH_06[4]	GAM_G_TH_06[3]	GAM_G_TH_06[2]	GAM_G_TH_06[1]	GAM_G_TH_06[0]
		GAM_G_TH_07[7]	GAM_G_TH_07[6]	GAM_G_TH_07[5]	GAM_G_TH_07[4]	GAM_G_TH_07[3]	GAM_G_TH_07[2]	GAM_G_TH_07[1]	GAM_G_TH_07[0]
	GAM_G_AREA3	GAM_G_TH_08[7]	GAM_G_TH_08[6]	GAM_G_TH_08[5]	GAM_G_TH_08[4]	GAM_G_TH_08[3]	GAM_G_TH_08[2]	GAM_G_TH_08[1]	GAM_G_TH_08[0]
		GAM_G_TH_09[7]	GAM_G_TH_09[6]	GAM_G_TH_09[5]	GAM_G_TH_09[4]	GAM_G_TH_09[3]	GAM_G_TH_09[2]	GAM_G_TH_09[1]	GAM_G_TH_09[0]
		GAM_G_TH_10[7]	GAM_G_TH_10[6]	GAM_G_TH_10[5]	GAM_G_TH_10[4]	GAM_G_TH_10[3]	GAM_G_TH_10[2]	GAM_G_TH_10[1]	GAM_G_TH_10[0]
		GAM_G_TH_11[7]	GAM_G_TH_11[6]	GAM_G_TH_11[5]	GAM_G_TH_11[4]	GAM_G_TH_11[3]	GAM_G_TH_11[2]	GAM_G_TH_11[1]	GAM_G_TH_11[0]
	GAM_G_AREA4	GAM_G_TH_12[7]	GAM_G_TH_12[6]	GAM_G_TH_12[5]	GAM_G_TH_12[4]	GAM_G_TH_12[3]	GAM_G_TH_12[2]	GAM_G_TH_12[1]	GAM_G_TH_12[0]
		GAM_G_TH_13[7]	GAM_G_TH_13[6]	GAM_G_TH_13[5]	GAM_G_TH_13[4]	GAM_G_TH_13[3]	GAM_G_TH_13[2]	GAM_G_TH_13[1]	GAM_G_TH_13[0]
		GAM_G_TH_14[7]	GAM_G_TH_14[6]	GAM_G_TH_14[5]	GAM_G_TH_14[4]	GAM_G_TH_14[3]	GAM_G_TH_14[2]	GAM_G_TH_14[1]	GAM_G_TH_14[0]
		GAM_G_TH_15[7]	GAM_G_TH_15[6]	GAM_G_TH_15[5]	GAM_G_TH_15[4]	GAM_G_TH_15[3]	GAM_G_TH_15[2]	GAM_G_TH_15[1]	GAM_G_TH_15[0]
	GAM_G_AREA5	GAM_G_TH_16[7]	GAM_G_TH_16[6]	GAM_G_TH_16[5]	GAM_G_TH_16[4]	GAM_G_TH_16[3]	GAM_G_TH_16[2]	GAM_G_TH_16[1]	GAM_G_TH_16[0]
		GAM_G_TH_17[7]	GAM_G_TH_17[6]	GAM_G_TH_17[5]	GAM_G_TH_17[4]	GAM_G_TH_17[3]	GAM_G_TH_17[2]	GAM_G_TH_17[1]	GAM_G_TH_17[0]
		GAM_G_TH_18[7]	GAM_G_TH_18[6]	GAM_G_TH_18[5]	GAM_G_TH_18[4]	GAM_G_TH_18[3]	GAM_G_TH_18[2]	GAM_G_TH_18[1]	GAM_G_TH_18[0]
		GAM_G_TH_19[7]	GAM_G_TH_19[6]	GAM_G_TH_19[5]	GAM_G_TH_19[4]	GAM_G_TH_19[3]	GAM_G_TH_19[2]	GAM_G_TH_19[1]	GAM_G_TH_19[0]
	GAM_G_AREA6	GAM_G_TH_20[7]	GAM_G_TH_20[6]	GAM_G_TH_20[5]	GAM_G_TH_20[4]	GAM_G_TH_20[3]	GAM_G_TH_20[2]	GAM_G_TH_20[1]	GAM_G_TH_20[0]
		GAM_G_TH_21[7]	GAM_G_TH_21[6]	GAM_G_TH_21[5]	GAM_G_TH_21[4]	GAM_G_TH_21[3]	GAM_G_TH_21[2]	GAM_G_TH_21[1]	GAM_G_TH_21[0]
		GAM_G_TH_22[7]	GAM_G_TH_22[6]	GAM_G_TH_22[5]	GAM_G_TH_22[4]	GAM_G_TH_22[3]	GAM_G_TH_22[2]	GAM_G_TH_22[1]	GAM_G_TH_22[0]
		GAM_G_TH_23[7]	GAM_G_TH_23[6]	GAM_G_TH_23[5]	GAM_G_TH_23[4]	GAM_G_TH_23[3]	GAM_G_TH_23[2]	GAM_G_TH_23[1]	GAM_G_TH_23[0]
	GAM_G_AREA7	GAM_G_TH_24[7]	GAM_G_TH_24[6]	GAM_G_TH_24[5]	GAM_G_TH_24[4]	GAM_G_TH_24[3]	GAM_G_TH_24[2]	GAM_G_TH_24[1]	GAM_G_TH_24[0]
		GAM_G_TH_25[7]	GAM_G_TH_25[6]	GAM_G_TH_25[5]	GAM_G_TH_25[4]	GAM_G_TH_25[3]	GAM_G_TH_25[2]	GAM_G_TH_25[1]	GAM_G_TH_25[0]
		GAM_G_TH_26[7]	GAM_G_TH_26[6]	GAM_G_TH_26[5]	GAM_G_TH_26[4]	GAM_G_TH_26[3]	GAM_G_TH_26[2]	GAM_G_TH_26[1]	GAM_G_TH_26[0]
		GAM_G_TH_27[7]	GAM_G_TH_27[6]	GAM_G_TH_27[5]	GAM_G_TH_27[4]	GAM_G_TH_27[3]	GAM_G_TH_27[2]	GAM_G_TH_27[1]	GAM_G_TH_27[0]
	GAM_G_AREA8	GAM_G_TH_28[7]	GAM_G_TH_28[6]	GAM_G_TH_28[5]	GAM_G_TH_28[4]	GAM_G_TH_28[3]	GAM_G_TH_28[2]	GAM_G_TH_28[1]	GAM_G_TH_28[0]
		GAM_G_TH_29[7]	GAM_G_TH_29[6]	GAM_G_TH_29[5]	GAM_G_TH_29[4]	GAM_G_TH_29[3]	GAM_G_TH_29[2]	GAM_G_TH_29[1]	GAM_G_TH_29[0]
		GAM_G_TH_30[7]	GAM_G_TH_30[6]	GAM_G_TH_30[5]	GAM_G_TH_30[4]	GAM_G_TH_30[3]	GAM_G_TH_30[2]	GAM_G_TH_30[1]	GAM_G_TH_30[0]
		GAM_G_TH_31[7]	GAM_G_TH_31[6]	GAM_G_TH_31[5]	GAM_G_TH_31[4]	GAM_G_TH_31[3]	GAM_G_TH_31[2]	GAM_G_TH_31[1]	GAM_G_TH_31[0]
	GAM_B_UPDATE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	GAM_B_VEN
	GAM_B_LUT1	-	-	-	-	-	GAM_B_GAIN_00[10]	GAM_B_GAIN_00[9]	GAM_B_GAIN_00[8]
		GAM_B_GAIN_00[7]	GAM_B_GAIN_00[6]	GAM_B_GAIN_00[5]	GAM_B_GAIN_00[4]	GAM_B_GAIN_00[3]	GAM_B_GAIN_00[2]	GAM_B_GAIN_00[1]	GAM_B_GAIN_00[0]
		-	-	-	-	-	GAM_B_GAIN_01[10]	GAM_B_GAIN_01[9]	GAM_B_GAIN_01[8]
		GAM_B_GAIN_01[7]	GAM_B_GAIN_01[6]	GAM_B_GAIN_01[5]	GAM_B_GAIN_01[4]	GAM_B_GAIN_01[3]	GAM_B_GAIN_01[2]	GAM_B_GAIN_01[1]	GAM_B_GAIN_01[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Video display controller 5	GAM_B_LUT2	-	-	-	-	-	GAM_B_GAIN_02[10]	GAM_B_GAIN_02[9]	GAM_B_GAIN_02[8]
		GAM_B_GAIN_02[7]	GAM_B_GAIN_02[6]	GAM_B_GAIN_02[5]	GAM_B_GAIN_02[4]	GAM_B_GAIN_02[3]	GAM_B_GAIN_02[2]	GAM_B_GAIN_02[1]	GAM_B_GAIN_02[0]
		-	-	-	-	-	GAM_B_GAIN_03[10]	GAM_B_GAIN_03[9]	GAM_B_GAIN_03[8]
		GAM_B_GAIN_03[7]	GAM_B_GAIN_03[6]	GAM_B_GAIN_03[5]	GAM_B_GAIN_03[4]	GAM_B_GAIN_03[3]	GAM_B_GAIN_03[2]	GAM_B_GAIN_03[1]	GAM_B_GAIN_03[0]
	GAM_B_LUT3	-	-	-	-	-	GAM_B_GAIN_04[10]	GAM_B_GAIN_04[9]	GAM_B_GAIN_04[8]
		GAM_B_GAIN_04[7]	GAM_B_GAIN_04[6]	GAM_B_GAIN_04[5]	GAM_B_GAIN_04[4]	GAM_B_GAIN_04[3]	GAM_B_GAIN_04[2]	GAM_B_GAIN_04[1]	GAM_B_GAIN_04[0]
		-	-	-	-	-	GAM_B_GAIN_05[10]	GAM_B_GAIN_05[9]	GAM_B_GAIN_05[8]
		GAM_B_GAIN_05[7]	GAM_B_GAIN_05[6]	GAM_B_GAIN_05[5]	GAM_B_GAIN_05[4]	GAM_B_GAIN_05[3]	GAM_B_GAIN_05[2]	GAM_B_GAIN_05[1]	GAM_B_GAIN_05[0]
	GAM_B_LUT4	-	-	-	-	-	GAM_B_GAIN_06[10]	GAM_B_GAIN_06[9]	GAM_B_GAIN_06[8]
		GAM_B_GAIN_06[7]	GAM_B_GAIN_06[6]	GAM_B_GAIN_06[5]	GAM_B_GAIN_06[4]	GAM_B_GAIN_06[3]	GAM_B_GAIN_06[2]	GAM_B_GAIN_06[1]	GAM_B_GAIN_06[0]
		-	-	-	-	-	GAM_B_GAIN_07[10]	GAM_B_GAIN_07[9]	GAM_B_GAIN_07[8]
		GAM_B_GAIN_07[7]	GAM_B_GAIN_07[6]	GAM_B_GAIN_07[5]	GAM_B_GAIN_07[4]	GAM_B_GAIN_07[3]	GAM_B_GAIN_07[2]	GAM_B_GAIN_07[1]	GAM_B_GAIN_07[0]
	GAM_B_LUT5	-	-	-	-	-	GAM_B_GAIN_08[10]	GAM_B_GAIN_08[9]	GAM_B_GAIN_08[8]
		GAM_B_GAIN_08[7]	GAM_B_GAIN_08[6]	GAM_B_GAIN_08[5]	GAM_B_GAIN_08[4]	GAM_B_GAIN_08[3]	GAM_B_GAIN_08[2]	GAM_B_GAIN_08[1]	GAM_B_GAIN_08[0]
		-	-	-	-	-	GAM_B_GAIN_09[10]	GAM_B_GAIN_09[9]	GAM_B_GAIN_09[8]
		GAM_B_GAIN_09[7]	GAM_B_GAIN_09[6]	GAM_B_GAIN_09[5]	GAM_B_GAIN_09[4]	GAM_B_GAIN_09[3]	GAM_B_GAIN_09[2]	GAM_B_GAIN_09[1]	GAM_B_GAIN_09[0]
	GAM_B_LUT6	-	-	-	-	-	GAM_B_GAIN_10[10]	GAM_B_GAIN_10[9]	GAM_B_GAIN_10[8]
		GAM_B_GAIN_10[7]	GAM_B_GAIN_10[6]	GAM_B_GAIN_10[5]	GAM_B_GAIN_10[4]	GAM_B_GAIN_10[3]	GAM_B_GAIN_10[2]	GAM_B_GAIN_10[1]	GAM_B_GAIN_10[0]
		-	-	-	-	-	GAM_B_GAIN_11[10]	GAM_B_GAIN_11[9]	GAM_B_GAIN_11[8]
		GAM_B_GAIN_11[7]	GAM_B_GAIN_11[6]	GAM_B_GAIN_11[5]	GAM_B_GAIN_11[4]	GAM_B_GAIN_11[3]	GAM_B_GAIN_11[2]	GAM_B_GAIN_11[1]	GAM_B_GAIN_11[0]
	GAM_B_LUT7	-	-	-	-	-	GAM_B_GAIN_12[10]	GAM_B_GAIN_12[9]	GAM_B_GAIN_12[8]
		GAM_B_GAIN_12[7]	GAM_B_GAIN_12[6]	GAM_B_GAIN_12[5]	GAM_B_GAIN_12[4]	GAM_B_GAIN_12[3]	GAM_B_GAIN_12[2]	GAM_B_GAIN_12[1]	GAM_B_GAIN_12[0]
		-	-	-	-	-	GAM_B_GAIN_13[10]	GAM_B_GAIN_13[9]	GAM_B_GAIN_13[8]
		GAM_B_GAIN_13[7]	GAM_B_GAIN_13[6]	GAM_B_GAIN_13[5]	GAM_B_GAIN_13[4]	GAM_B_GAIN_13[3]	GAM_B_GAIN_13[2]	GAM_B_GAIN_13[1]	GAM_B_GAIN_13[0]
	GAM_B_LUT8	-	-	-	-	-	GAM_B_GAIN_14[10]	GAM_B_GAIN_14[9]	GAM_B_GAIN_14[8]
		GAM_B_GAIN_14[7]	GAM_B_GAIN_14[6]	GAM_B_GAIN_14[5]	GAM_B_GAIN_14[4]	GAM_B_GAIN_14[3]	GAM_B_GAIN_14[2]	GAM_B_GAIN_14[1]	GAM_B_GAIN_14[0]
		-	-	-	-	-	GAM_B_GAIN_15[10]	GAM_B_GAIN_15[9]	GAM_B_GAIN_15[8]
		GAM_B_GAIN_15[7]	GAM_B_GAIN_15[6]	GAM_B_GAIN_15[5]	GAM_B_GAIN_15[4]	GAM_B_GAIN_15[3]	GAM_B_GAIN_15[2]	GAM_B_GAIN_15[1]	GAM_B_GAIN_15[0]
	GAM_B_LUT9	-	-	-	-	-	GAM_B_GAIN_16[10]	GAM_B_GAIN_16[9]	GAM_B_GAIN_16[8]
		GAM_B_GAIN_16[7]	GAM_B_GAIN_16[6]	GAM_B_GAIN_16[5]	GAM_B_GAIN_16[4]	GAM_B_GAIN_16[3]	GAM_B_GAIN_16[2]	GAM_B_GAIN_16[1]	GAM_B_GAIN_16[0]
		-	-	-	-	-	GAM_B_GAIN_17[10]	GAM_B_GAIN_17[9]	GAM_B_GAIN_17[8]
		GAM_B_GAIN_17[7]	GAM_B_GAIN_17[6]	GAM_B_GAIN_17[5]	GAM_B_GAIN_17[4]	GAM_B_GAIN_17[3]	GAM_B_GAIN_17[2]	GAM_B_GAIN_17[1]	GAM_B_GAIN_17[0]
	GAM_B_LUT10	-	-	-	-	-	GAM_B_GAIN_18[10]	GAM_B_GAIN_18[9]	GAM_B_GAIN_18[8]
		GAM_B_GAIN_18[7]	GAM_B_GAIN_18[6]	GAM_B_GAIN_18[5]	GAM_B_GAIN_18[4]	GAM_B_GAIN_18[3]	GAM_B_GAIN_18[2]	GAM_B_GAIN_18[1]	GAM_B_GAIN_18[0]
		-	-	-	-	-	GAM_B_GAIN_19[10]	GAM_B_GAIN_19[9]	GAM_B_GAIN_19[8]
		GAM_B_GAIN_19[7]	GAM_B_GAIN_19[6]	GAM_B_GAIN_19[5]	GAM_B_GAIN_19[4]	GAM_B_GAIN_19[3]	GAM_B_GAIN_19[2]	GAM_B_GAIN_19[1]	GAM_B_GAIN_19[0]
	GAM_B_LUT11	-	-	-	-	-	GAM_B_GAIN_20[10]	GAM_B_GAIN_20[9]	GAM_B_GAIN_20[8]
		GAM_B_GAIN_20[7]	GAM_B_GAIN_20[6]	GAM_B_GAIN_20[5]	GAM_B_GAIN_20[4]	GAM_B_GAIN_20[3]	GAM_B_GAIN_20[2]	GAM_B_GAIN_20[1]	GAM_B_GAIN_20[0]
		-	-	-	-	-	GAM_B_GAIN_21[10]	GAM_B_GAIN_21[9]	GAM_B_GAIN_21[8]
		GAM_B_GAIN_21[7]	GAM_B_GAIN_21[6]	GAM_B_GAIN_21[5]	GAM_B_GAIN_21[4]	GAM_B_GAIN_21[3]	GAM_B_GAIN_21[2]	GAM_B_GAIN_21[1]	GAM_B_GAIN_21[0]
	GAM_B_LUT12	-	-	-	-	-	GAM_B_GAIN_22[10]	GAM_B_GAIN_22[9]	GAM_B_GAIN_22[8]
		GAM_B_GAIN_22[7]	GAM_B_GAIN_22[6]	GAM_B_GAIN_22[5]	GAM_B_GAIN_22[4]	GAM_B_GAIN_22[3]	GAM_B_GAIN_22[2]	GAM_B_GAIN_22[1]	GAM_B_GAIN_22[0]
		-	-	-	-	-	GAM_B_GAIN_23[10]	GAM_B_GAIN_23[9]	GAM_B_GAIN_23[8]
		GAM_B_GAIN_23[7]	GAM_B_GAIN_23[6]	GAM_B_GAIN_23[5]	GAM_B_GAIN_23[4]	GAM_B_GAIN_23[3]	GAM_B_GAIN_23[2]	GAM_B_GAIN_23[1]	GAM_B_GAIN_23[0]
	GAM_B_LUT13	-	-	-	-	-	GAM_B_GAIN_24[10]	GAM_B_GAIN_24[9]	GAM_B_GAIN_24[8]
		GAM_B_GAIN_24[7]	GAM_B_GAIN_24[6]	GAM_B_GAIN_24[5]	GAM_B_GAIN_24[4]	GAM_B_GAIN_24[3]	GAM_B_GAIN_24[2]	GAM_B_GAIN_24[1]	GAM_B_GAIN_24[0]
		-	-	-	-	-	GAM_B_GAIN_25[10]	GAM_B_GAIN_25[9]	GAM_B_GAIN_25[8]
		GAM_B_GAIN_25[7]	GAM_B_GAIN_25[6]	GAM_B_GAIN_25[5]	GAM_B_GAIN_25[4]	GAM_B_GAIN_25[3]	GAM_B_GAIN_25[2]	GAM_B_GAIN_25[1]	GAM_B_GAIN_25[0]
	GAM_B_LUT14	-	-	-	-	-	GAM_B_GAIN_26[10]	GAM_B_GAIN_26[9]	GAM_B_GAIN_26[8]
		GAM_B_GAIN_26[7]	GAM_B_GAIN_26[6]	GAM_B_GAIN_26[5]	GAM_B_GAIN_26[4]	GAM_B_GAIN_26[3]	GAM_B_GAIN_26[2]	GAM_B_GAIN_26[1]	GAM_B_GAIN_26[0]
		-	-	-	-	-	GAM_B_GAIN_27[10]	GAM_B_GAIN_27[9]	GAM_B_GAIN_27[8]
		GAM_B_GAIN_27[7]	GAM_B_GAIN_27[6]	GAM_B_GAIN_27[5]	GAM_B_GAIN_27[4]	GAM_B_GAIN_27[3]	GAM_B_GAIN_27[2]	GAM_B_GAIN_27[1]	GAM_B_GAIN_27[0]
	GAM_B_LUT15	-	-	-	-	-	GAM_B_GAIN_28[10]	GAM_B_GAIN_28[9]	GAM_B_GAIN_28[8]
		GAM_B_GAIN_28[7]	GAM_B_GAIN_28[6]	GAM_B_GAIN_28[5]	GAM_B_GAIN_28[4]	GAM_B_GAIN_28[3]	GAM_B_GAIN_28[2]	GAM_B_GAIN_28[1]	GAM_B_GAIN_28[0]
		-	-	-	-	-	GAM_B_GAIN_29[10]	GAM_B_GAIN_29[9]	GAM_B_GAIN_29[8]
		GAM_B_GAIN_29[7]	GAM_B_GAIN_29[6]	GAM_B_GAIN_29[5]	GAM_B_GAIN_29[4]	GAM_B_GAIN_29[3]	GAM_B_GAIN_29[2]	GAM_B_GAIN_29[1]	GAM_B_GAIN_29[0]
	GAM_B_LUT16	-	-	-	-	-	GAM_B_GAIN_30[10]	GAM_B_GAIN_30[9]	GAM_B_GAIN_30[8]
		GAM_B_GAIN_30[7]	GAM_B_GAIN_30[6]	GAM_B_GAIN_30[5]	GAM_B_GAIN_30[4]	GAM_B_GAIN_30[3]	GAM_B_GAIN_30[2]	GAM_B_GAIN_30[1]	GAM_B_GAIN_30[0]
		-	-	-	-	-	GAM_B_GAIN_31[10]	GAM_B_GAIN_31[9]	GAM_B_GAIN_31[8]
		GAM_B_GAIN_31[7]	GAM_B_GAIN_31[6]	GAM_B_GAIN_31[5]	GAM_B_GAIN_31[4]	GAM_B_GAIN_31[3]	GAM_B_GAIN_31[2]	GAM_B_GAIN_31[1]	GAM_B_GAIN_31[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Video display controller 5	GAM_B_AREA1	-	-	-	-	-	-	-	-
		GAM_B_TH_01[7]	GAM_B_TH_01[6]	GAM_B_TH_01[5]	GAM_B_TH_01[4]	GAM_B_TH_01[3]	GAM_B_TH_01[2]	GAM_B_TH_01[1]	GAM_B_TH_01[0]
		GAM_B_TH_02[7]	GAM_B_TH_02[6]	GAM_B_TH_02[5]	GAM_B_TH_02[4]	GAM_B_TH_02[3]	GAM_B_TH_02[2]	GAM_B_TH_02[1]	GAM_B_TH_02[0]
		GAM_B_TH_03[7]	GAM_B_TH_03[6]	GAM_B_TH_03[5]	GAM_B_TH_03[4]	GAM_B_TH_03[3]	GAM_B_TH_03[2]	GAM_B_TH_03[1]	GAM_B_TH_03[0]
	GAM_B_AREA2	GAM_B_TH_04[7]	GAM_B_TH_04[6]	GAM_B_TH_04[5]	GAM_B_TH_04[4]	GAM_B_TH_04[3]	GAM_B_TH_04[2]	GAM_B_TH_04[1]	GAM_B_TH_04[0]
		GAM_B_TH_05[7]	GAM_B_TH_05[6]	GAM_B_TH_05[5]	GAM_B_TH_05[4]	GAM_B_TH_05[3]	GAM_B_TH_05[2]	GAM_B_TH_05[1]	GAM_B_TH_05[0]
		GAM_B_TH_06[7]	GAM_B_TH_06[6]	GAM_B_TH_06[5]	GAM_B_TH_06[4]	GAM_B_TH_06[3]	GAM_B_TH_06[2]	GAM_B_TH_06[1]	GAM_B_TH_06[0]
		GAM_B_TH_07[7]	GAM_B_TH_07[6]	GAM_B_TH_07[5]	GAM_B_TH_07[4]	GAM_B_TH_07[3]	GAM_B_TH_07[2]	GAM_B_TH_07[1]	GAM_B_TH_07[0]
	GAM_B_AREA3	GAM_B_TH_08[7]	GAM_B_TH_08[6]	GAM_B_TH_08[5]	GAM_B_TH_08[4]	GAM_B_TH_08[3]	GAM_B_TH_08[2]	GAM_B_TH_08[1]	GAM_B_TH_08[0]
		GAM_B_TH_09[7]	GAM_B_TH_09[6]	GAM_B_TH_09[5]	GAM_B_TH_09[4]	GAM_B_TH_09[3]	GAM_B_TH_09[2]	GAM_B_TH_09[1]	GAM_B_TH_09[0]
		GAM_B_TH_10[7]	GAM_B_TH_10[6]	GAM_B_TH_10[5]	GAM_B_TH_10[4]	GAM_B_TH_10[3]	GAM_B_TH_10[2]	GAM_B_TH_10[1]	GAM_B_TH_10[0]
		GAM_B_TH_11[7]	GAM_B_TH_11[6]	GAM_B_TH_11[5]	GAM_B_TH_11[4]	GAM_B_TH_11[3]	GAM_B_TH_11[2]	GAM_B_TH_11[1]	GAM_B_TH_11[0]
	GAM_B_AREA4	GAM_B_TH_12[7]	GAM_B_TH_12[6]	GAM_B_TH_12[5]	GAM_B_TH_12[4]	GAM_B_TH_12[3]	GAM_B_TH_12[2]	GAM_B_TH_12[1]	GAM_B_TH_12[0]
		GAM_B_TH_13[7]	GAM_B_TH_13[6]	GAM_B_TH_13[5]	GAM_B_TH_13[4]	GAM_B_TH_13[3]	GAM_B_TH_13[2]	GAM_B_TH_13[1]	GAM_B_TH_13[0]
		GAM_B_TH_14[7]	GAM_B_TH_14[6]	GAM_B_TH_14[5]	GAM_B_TH_14[4]	GAM_B_TH_14[3]	GAM_B_TH_14[2]	GAM_B_TH_14[1]	GAM_B_TH_14[0]
		GAM_B_TH_15[7]	GAM_B_TH_15[6]	GAM_B_TH_15[5]	GAM_B_TH_15[4]	GAM_B_TH_15[3]	GAM_B_TH_15[2]	GAM_B_TH_15[1]	GAM_B_TH_15[0]
	GAM_B_AREA5	GAM_B_TH_16[7]	GAM_B_TH_16[6]	GAM_B_TH_16[5]	GAM_B_TH_16[4]	GAM_B_TH_16[3]	GAM_B_TH_16[2]	GAM_B_TH_16[1]	GAM_B_TH_16[0]
		GAM_B_TH_17[7]	GAM_B_TH_17[6]	GAM_B_TH_17[5]	GAM_B_TH_17[4]	GAM_B_TH_17[3]	GAM_B_TH_17[2]	GAM_B_TH_17[1]	GAM_B_TH_17[0]
		GAM_B_TH_18[7]	GAM_B_TH_18[6]	GAM_B_TH_18[5]	GAM_B_TH_18[4]	GAM_B_TH_18[3]	GAM_B_TH_18[2]	GAM_B_TH_18[1]	GAM_B_TH_18[0]
		GAM_B_TH_19[7]	GAM_B_TH_19[6]	GAM_B_TH_19[5]	GAM_B_TH_19[4]	GAM_B_TH_19[3]	GAM_B_TH_19[2]	GAM_B_TH_19[1]	GAM_B_TH_19[0]
	GAM_B_AREA6	GAM_B_TH_20[7]	GAM_B_TH_20[6]	GAM_B_TH_20[5]	GAM_B_TH_20[4]	GAM_B_TH_20[3]	GAM_B_TH_20[2]	GAM_B_TH_20[1]	GAM_B_TH_20[0]
		GAM_B_TH_21[7]	GAM_B_TH_21[6]	GAM_B_TH_21[5]	GAM_B_TH_21[4]	GAM_B_TH_21[3]	GAM_B_TH_21[2]	GAM_B_TH_21[1]	GAM_B_TH_21[0]
		GAM_B_TH_22[7]	GAM_B_TH_22[6]	GAM_B_TH_22[5]	GAM_B_TH_22[4]	GAM_B_TH_22[3]	GAM_B_TH_22[2]	GAM_B_TH_22[1]	GAM_B_TH_22[0]
		GAM_B_TH_23[7]	GAM_B_TH_23[6]	GAM_B_TH_23[5]	GAM_B_TH_23[4]	GAM_B_TH_23[3]	GAM_B_TH_23[2]	GAM_B_TH_23[1]	GAM_B_TH_23[0]
	GAM_B_AREA7	GAM_B_TH_24[7]	GAM_B_TH_24[6]	GAM_B_TH_24[5]	GAM_B_TH_24[4]	GAM_B_TH_24[3]	GAM_B_TH_24[2]	GAM_B_TH_24[1]	GAM_B_TH_24[0]
		GAM_B_TH_25[7]	GAM_B_TH_25[6]	GAM_B_TH_25[5]	GAM_B_TH_25[4]	GAM_B_TH_25[3]	GAM_B_TH_25[2]	GAM_B_TH_25[1]	GAM_B_TH_25[0]
		GAM_B_TH_26[7]	GAM_B_TH_26[6]	GAM_B_TH_26[5]	GAM_B_TH_26[4]	GAM_B_TH_26[3]	GAM_B_TH_26[2]	GAM_B_TH_26[1]	GAM_B_TH_26[0]
		GAM_B_TH_27[7]	GAM_B_TH_27[6]	GAM_B_TH_27[5]	GAM_B_TH_27[4]	GAM_B_TH_27[3]	GAM_B_TH_27[2]	GAM_B_TH_27[1]	GAM_B_TH_27[0]
	GAM_B_AREA8	GAM_B_TH_28[7]	GAM_B_TH_28[6]	GAM_B_TH_28[5]	GAM_B_TH_28[4]	GAM_B_TH_28[3]	GAM_B_TH_28[2]	GAM_B_TH_28[1]	GAM_B_TH_28[0]
		GAM_B_TH_29[7]	GAM_B_TH_29[6]	GAM_B_TH_29[5]	GAM_B_TH_29[4]	GAM_B_TH_29[3]	GAM_B_TH_29[2]	GAM_B_TH_29[1]	GAM_B_TH_29[0]
		GAM_B_TH_30[7]	GAM_B_TH_30[6]	GAM_B_TH_30[5]	GAM_B_TH_30[4]	GAM_B_TH_30[3]	GAM_B_TH_30[2]	GAM_B_TH_30[1]	GAM_B_TH_30[0]
		GAM_B_TH_31[7]	GAM_B_TH_31[6]	GAM_B_TH_31[5]	GAM_B_TH_31[4]	GAM_B_TH_31[3]	GAM_B_TH_31[2]	GAM_B_TH_31[1]	GAM_B_TH_31[0]
	GAM_R_UPDATE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	GAM_R_VEN
	GAM_R_LUT1	-	-	-	-	-	GAM_R_GAIN_00[10]	GAM_R_GAIN_00[9]	GAM_R_GAIN_00[8]
		GAM_R_GAIN_00[7]	GAM_R_GAIN_00[6]	GAM_R_GAIN_00[5]	GAM_R_GAIN_00[4]	GAM_R_GAIN_00[3]	GAM_R_GAIN_00[2]	GAM_R_GAIN_00[1]	GAM_R_GAIN_00[0]
		-	-	-	-	-	GAM_R_GAIN_01[10]	GAM_R_GAIN_01[9]	GAM_R_GAIN_01[8]
		GAM_R_GAIN_01[7]	GAM_R_GAIN_01[6]	GAM_R_GAIN_01[5]	GAM_R_GAIN_01[4]	GAM_R_GAIN_01[3]	GAM_R_GAIN_01[2]	GAM_R_GAIN_01[1]	GAM_R_GAIN_01[0]
	GAM_R_LUT2	-	-	-	-	-	GAM_R_GAIN_02[10]	GAM_R_GAIN_02[9]	GAM_R_GAIN_02[8]
		GAM_R_GAIN_02[7]	GAM_R_GAIN_02[6]	GAM_R_GAIN_02[5]	GAM_R_GAIN_02[4]	GAM_R_GAIN_02[3]	GAM_R_GAIN_02[2]	GAM_R_GAIN_02[1]	GAM_R_GAIN_02[0]
		-	-	-	-	-	GAM_R_GAIN_03[10]	GAM_R_GAIN_03[9]	GAM_R_GAIN_03[8]
		GAM_R_GAIN_03[7]	GAM_R_GAIN_03[6]	GAM_R_GAIN_03[5]	GAM_R_GAIN_03[4]	GAM_R_GAIN_03[3]	GAM_R_GAIN_03[2]	GAM_R_GAIN_03[1]	GAM_R_GAIN_03[0]
	GAM_R_LUT3	-	-	-	-	-	GAM_R_GAIN_04[10]	GAM_R_GAIN_04[9]	GAM_R_GAIN_04[8]
		GAM_R_GAIN_04[7]	GAM_R_GAIN_04[6]	GAM_R_GAIN_04[5]	GAM_R_GAIN_04[4]	GAM_R_GAIN_04[3]	GAM_R_GAIN_04[2]	GAM_R_GAIN_04[1]	GAM_R_GAIN_04[0]
		-	-	-	-	-	GAM_R_GAIN_05[10]	GAM_R_GAIN_05[9]	GAM_R_GAIN_05[8]
		GAM_R_GAIN_05[7]	GAM_R_GAIN_05[6]	GAM_R_GAIN_05[5]	GAM_R_GAIN_05[4]	GAM_R_GAIN_05[3]	GAM_R_GAIN_05[2]	GAM_R_GAIN_05[1]	GAM_R_GAIN_05[0]
	GAM_R_LUT4	-	-	-	-	-	GAM_R_GAIN_06[10]	GAM_R_GAIN_06[9]	GAM_R_GAIN_06[8]
		GAM_R_GAIN_06[7]	GAM_R_GAIN_06[6]	GAM_R_GAIN_06[5]	GAM_R_GAIN_06[4]	GAM_R_GAIN_06[3]	GAM_R_GAIN_06[2]	GAM_R_GAIN_06[1]	GAM_R_GAIN_06[0]
		-	-	-	-	-	GAM_R_GAIN_07[10]	GAM_R_GAIN_07[9]	GAM_R_GAIN_07[8]
		GAM_R_GAIN_07[7]	GAM_R_GAIN_07[6]	GAM_R_GAIN_07[5]	GAM_R_GAIN_07[4]	GAM_R_GAIN_07[3]	GAM_R_GAIN_07[2]	GAM_R_GAIN_07[1]	GAM_R_GAIN_07[0]
	GAM_R_LUT5	-	-	-	-	-	GAM_R_GAIN_08[10]	GAM_R_GAIN_08[9]	GAM_R_GAIN_08[8]
		GAM_R_GAIN_08[7]	GAM_R_GAIN_08[6]	GAM_R_GAIN_08[5]	GAM_R_GAIN_08[4]	GAM_R_GAIN_08[3]	GAM_R_GAIN_08[2]	GAM_R_GAIN_08[1]	GAM_R_GAIN_08[0]
		-	-	-	-	-	GAM_R_GAIN_09[10]	GAM_R_GAIN_09[9]	GAM_R_GAIN_09[8]
		GAM_R_GAIN_09[7]	GAM_R_GAIN_09[6]	GAM_R_GAIN_09[5]	GAM_R_GAIN_09[4]	GAM_R_GAIN_09[3]	GAM_R_GAIN_09[2]	GAM_R_GAIN_09[1]	GAM_R_GAIN_09[0]
	GAM_R_LUT6	-	-	-	-	-	GAM_R_GAIN_10[10]	GAM_R_GAIN_10[9]	GAM_R_GAIN_10[8]
		GAM_R_GAIN_10[7]	GAM_R_GAIN_10[6]	GAM_R_GAIN_10[5]	GAM_R_GAIN_10[4]	GAM_R_GAIN_10[3]	GAM_R_GAIN_10[2]	GAM_R_GAIN_10[1]	GAM_R_GAIN_10[0]
		-	-	-	-	-	GAM_R_GAIN_11[10]	GAM_R_GAIN_11[9]	GAM_R_GAIN_11[8]
		GAM_R_GAIN_11[7]	GAM_R_GAIN_11[6]	GAM_R_GAIN_11[5]	GAM_R_GAIN_11[4]	GAM_R_GAIN_11[3]	GAM_R_GAIN_11[2]	GAM_R_GAIN_11[1]	GAM_R_GAIN_11[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Video display controller 5	GAM_R_LUT7	-	-	-	-	-	GAM_R_GAIN_12[10]	GAM_R_GAIN_12[9]	GAM_R_GAIN_12[8]
		GAM_R_GAIN_12[7]	GAM_R_GAIN_12[6]	GAM_R_GAIN_12[5]	GAM_R_GAIN_12[4]	GAM_R_GAIN_12[3]	GAM_R_GAIN_12[2]	GAM_R_GAIN_12[1]	GAM_R_GAIN_12[0]
		-	-	-	-	-	GAM_R_GAIN_13[10]	GAM_R_GAIN_13[9]	GAM_R_GAIN_13[8]
		GAM_R_GAIN_13[7]	GAM_R_GAIN_13[6]	GAM_R_GAIN_13[5]	GAM_R_GAIN_13[4]	GAM_R_GAIN_13[3]	GAM_R_GAIN_13[2]	GAM_R_GAIN_13[1]	GAM_R_GAIN_13[0]
	GAM_R_LUT8	-	-	-	-	-	GAM_R_GAIN_14[10]	GAM_R_GAIN_14[9]	GAM_R_GAIN_14[8]
		GAM_R_GAIN_14[7]	GAM_R_GAIN_14[6]	GAM_R_GAIN_14[5]	GAM_R_GAIN_14[4]	GAM_R_GAIN_14[3]	GAM_R_GAIN_14[2]	GAM_R_GAIN_14[1]	GAM_R_GAIN_14[0]
		-	-	-	-	-	GAM_R_GAIN_15[10]	GAM_R_GAIN_15[9]	GAM_R_GAIN_15[8]
		GAM_R_GAIN_15[7]	GAM_R_GAIN_15[6]	GAM_R_GAIN_15[5]	GAM_R_GAIN_15[4]	GAM_R_GAIN_15[3]	GAM_R_GAIN_15[2]	GAM_R_GAIN_15[1]	GAM_R_GAIN_15[0]
	GAM_R_LUT9	-	-	-	-	-	GAM_R_GAIN_16[10]	GAM_R_GAIN_16[9]	GAM_R_GAIN_16[8]
		GAM_R_GAIN_16[7]	GAM_R_GAIN_16[6]	GAM_R_GAIN_16[5]	GAM_R_GAIN_16[4]	GAM_R_GAIN_16[3]	GAM_R_GAIN_16[2]	GAM_R_GAIN_16[1]	GAM_R_GAIN_16[0]
		-	-	-	-	-	GAM_R_GAIN_17[10]	GAM_R_GAIN_17[9]	GAM_R_GAIN_17[8]
		GAM_R_GAIN_17[7]	GAM_R_GAIN_17[6]	GAM_R_GAIN_17[5]	GAM_R_GAIN_17[4]	GAM_R_GAIN_17[3]	GAM_R_GAIN_17[2]	GAM_R_GAIN_17[1]	GAM_R_GAIN_17[0]
	GAM_R_LUT10	-	-	-	-	-	GAM_R_GAIN_18[10]	GAM_R_GAIN_18[9]	GAM_R_GAIN_18[8]
		GAM_R_GAIN_18[7]	GAM_R_GAIN_18[6]	GAM_R_GAIN_18[5]	GAM_R_GAIN_18[4]	GAM_R_GAIN_18[3]	GAM_R_GAIN_18[2]	GAM_R_GAIN_18[1]	GAM_R_GAIN_18[0]
		-	-	-	-	-	GAM_R_GAIN_19[10]	GAM_R_GAIN_19[9]	GAM_R_GAIN_19[8]
		GAM_R_GAIN_19[7]	GAM_R_GAIN_19[6]	GAM_R_GAIN_19[5]	GAM_R_GAIN_19[4]	GAM_R_GAIN_19[3]	GAM_R_GAIN_19[2]	GAM_R_GAIN_19[1]	GAM_R_GAIN_19[0]
	GAM_R_LUT11	-	-	-	-	-	GAM_R_GAIN_20[10]	GAM_R_GAIN_20[9]	GAM_R_GAIN_20[8]
		GAM_R_GAIN_20[7]	GAM_R_GAIN_20[6]	GAM_R_GAIN_20[5]	GAM_R_GAIN_20[4]	GAM_R_GAIN_20[3]	GAM_R_GAIN_20[2]	GAM_R_GAIN_20[1]	GAM_R_GAIN_20[0]
		-	-	-	-	-	GAM_R_GAIN_21[10]	GAM_R_GAIN_21[9]	GAM_R_GAIN_21[8]
		GAM_R_GAIN_21[7]	GAM_R_GAIN_21[6]	GAM_R_GAIN_21[5]	GAM_R_GAIN_21[4]	GAM_R_GAIN_21[3]	GAM_R_GAIN_21[2]	GAM_R_GAIN_21[1]	GAM_R_GAIN_21[0]
	GAM_R_LUT12	-	-	-	-	-	GAM_R_GAIN_22[10]	GAM_R_GAIN_22[9]	GAM_R_GAIN_22[8]
		GAM_R_GAIN_22[7]	GAM_R_GAIN_22[6]	GAM_R_GAIN_22[5]	GAM_R_GAIN_22[4]	GAM_R_GAIN_22[3]	GAM_R_GAIN_22[2]	GAM_R_GAIN_22[1]	GAM_R_GAIN_22[0]
		-	-	-	-	-	GAM_R_GAIN_23[10]	GAM_R_GAIN_23[9]	GAM_R_GAIN_23[8]
		GAM_R_GAIN_23[7]	GAM_R_GAIN_23[6]	GAM_R_GAIN_23[5]	GAM_R_GAIN_23[4]	GAM_R_GAIN_23[3]	GAM_R_GAIN_23[2]	GAM_R_GAIN_23[1]	GAM_R_GAIN_23[0]
	GAM_R_LUT13	-	-	-	-	-	GAM_R_GAIN_24[10]	GAM_R_GAIN_24[9]	GAM_R_GAIN_24[8]
		GAM_R_GAIN_24[7]	GAM_R_GAIN_24[6]	GAM_R_GAIN_24[5]	GAM_R_GAIN_24[4]	GAM_R_GAIN_24[3]	GAM_R_GAIN_24[2]	GAM_R_GAIN_24[1]	GAM_R_GAIN_24[0]
		-	-	-	-	-	GAM_R_GAIN_25[10]	GAM_R_GAIN_25[9]	GAM_R_GAIN_25[8]
		GAM_R_GAIN_25[7]	GAM_R_GAIN_25[6]	GAM_R_GAIN_25[5]	GAM_R_GAIN_25[4]	GAM_R_GAIN_25[3]	GAM_R_GAIN_25[2]	GAM_R_GAIN_25[1]	GAM_R_GAIN_25[0]
	GAM_R_LUT14	-	-	-	-	-	GAM_R_GAIN_26[10]	GAM_R_GAIN_26[9]	GAM_R_GAIN_26[8]
		GAM_R_GAIN_26[7]	GAM_R_GAIN_26[6]	GAM_R_GAIN_26[5]	GAM_R_GAIN_26[4]	GAM_R_GAIN_26[3]	GAM_R_GAIN_26[2]	GAM_R_GAIN_26[1]	GAM_R_GAIN_26[0]
		-	-	-	-	-	GAM_R_GAIN_27[10]	GAM_R_GAIN_27[9]	GAM_R_GAIN_27[8]
		GAM_R_GAIN_27[7]	GAM_R_GAIN_27[6]	GAM_R_GAIN_27[5]	GAM_R_GAIN_27[4]	GAM_R_GAIN_27[3]	GAM_R_GAIN_27[2]	GAM_R_GAIN_27[1]	GAM_R_GAIN_27[0]
	GAM_R_LUT15	-	-	-	-	-	GAM_R_GAIN_28[10]	GAM_R_GAIN_28[9]	GAM_R_GAIN_28[8]
		GAM_R_GAIN_28[7]	GAM_R_GAIN_28[6]	GAM_R_GAIN_28[5]	GAM_R_GAIN_28[4]	GAM_R_GAIN_28[3]	GAM_R_GAIN_28[2]	GAM_R_GAIN_28[1]	GAM_R_GAIN_28[0]
		-	-	-	-	-	GAM_R_GAIN_29[10]	GAM_R_GAIN_29[9]	GAM_R_GAIN_29[8]
		GAM_R_GAIN_29[7]	GAM_R_GAIN_29[6]	GAM_R_GAIN_29[5]	GAM_R_GAIN_29[4]	GAM_R_GAIN_29[3]	GAM_R_GAIN_29[2]	GAM_R_GAIN_29[1]	GAM_R_GAIN_29[0]
	GAM_R_LUT16	-	-	-	-	-	GAM_R_GAIN_30[10]	GAM_R_GAIN_30[9]	GAM_R_GAIN_30[8]
		GAM_R_GAIN_30[7]	GAM_R_GAIN_30[6]	GAM_R_GAIN_30[5]	GAM_R_GAIN_30[4]	GAM_R_GAIN_30[3]	GAM_R_GAIN_30[2]	GAM_R_GAIN_30[1]	GAM_R_GAIN_30[0]
		-	-	-	-	-	GAM_R_GAIN_31[10]	GAM_R_GAIN_31[9]	GAM_R_GAIN_31[8]
		GAM_R_GAIN_31[7]	GAM_R_GAIN_31[6]	GAM_R_GAIN_31[5]	GAM_R_GAIN_31[4]	GAM_R_GAIN_31[3]	GAM_R_GAIN_31[2]	GAM_R_GAIN_31[1]	GAM_R_GAIN_31[0]
	GAM_R_AREA1	-	-	-	-	-	-	-	-
		GAM_R_TH_01[7]	GAM_R_TH_01[6]	GAM_R_TH_01[5]	GAM_R_TH_01[4]	GAM_R_TH_01[3]	GAM_R_TH_01[2]	GAM_R_TH_01[1]	GAM_R_TH_01[0]
		GAM_R_TH_02[7]	GAM_R_TH_02[6]	GAM_R_TH_02[5]	GAM_R_TH_02[4]	GAM_R_TH_02[3]	GAM_R_TH_02[2]	GAM_R_TH_02[1]	GAM_R_TH_02[0]
		GAM_R_TH_03[7]	GAM_R_TH_03[6]	GAM_R_TH_03[5]	GAM_R_TH_03[4]	GAM_R_TH_03[3]	GAM_R_TH_03[2]	GAM_R_TH_03[1]	GAM_R_TH_03[0]
	GAM_R_AREA2	GAM_R_TH_04[7]	GAM_R_TH_04[6]	GAM_R_TH_04[5]	GAM_R_TH_04[4]	GAM_R_TH_04[3]	GAM_R_TH_04[2]	GAM_R_TH_04[1]	GAM_R_TH_04[0]
		GAM_R_TH_05[7]	GAM_R_TH_05[6]	GAM_R_TH_05[5]	GAM_R_TH_05[4]	GAM_R_TH_05[3]	GAM_R_TH_05[2]	GAM_R_TH_05[1]	GAM_R_TH_05[0]
		GAM_R_TH_06[7]	GAM_R_TH_06[6]	GAM_R_TH_06[5]	GAM_R_TH_06[4]	GAM_R_TH_06[3]	GAM_R_TH_06[2]	GAM_R_TH_06[1]	GAM_R_TH_06[0]
		GAM_R_TH_07[7]	GAM_R_TH_07[6]	GAM_R_TH_07[5]	GAM_R_TH_07[4]	GAM_R_TH_07[3]	GAM_R_TH_07[2]	GAM_R_TH_07[1]	GAM_R_TH_07[0]
	GAM_R_AREA3	GAM_R_TH_08[7]	GAM_R_TH_08[6]	GAM_R_TH_08[5]	GAM_R_TH_08[4]	GAM_R_TH_08[3]	GAM_R_TH_08[2]	GAM_R_TH_08[1]	GAM_R_TH_08[0]
		GAM_R_TH_09[7]	GAM_R_TH_09[6]	GAM_R_TH_09[5]	GAM_R_TH_09[4]	GAM_R_TH_09[3]	GAM_R_TH_09[2]	GAM_R_TH_09[1]	GAM_R_TH_09[0]
		GAM_R_TH_10[7]	GAM_R_TH_10[6]	GAM_R_TH_10[5]	GAM_R_TH_10[4]	GAM_R_TH_10[3]	GAM_R_TH_10[2]	GAM_R_TH_10[1]	GAM_R_TH_10[0]
		GAM_R_TH_11[7]	GAM_R_TH_11[6]	GAM_R_TH_11[5]	GAM_R_TH_11[4]	GAM_R_TH_11[3]	GAM_R_TH_11[2]	GAM_R_TH_11[1]	GAM_R_TH_11[0]
	GAM_R_AREA4	GAM_R_TH_12[7]	GAM_R_TH_12[6]	GAM_R_TH_12[5]	GAM_R_TH_12[4]	GAM_R_TH_12[3]	GAM_R_TH_12[2]	GAM_R_TH_12[1]	GAM_R_TH_12[0]
		GAM_R_TH_13[7]	GAM_R_TH_13[6]	GAM_R_TH_13[5]	GAM_R_TH_13[4]	GAM_R_TH_13[3]	GAM_R_TH_13[2]	GAM_R_TH_13[1]	GAM_R_TH_13[0]
		GAM_R_TH_14[7]	GAM_R_TH_14[6]	GAM_R_TH_14[5]	GAM_R_TH_14[4]	GAM_R_TH_14[3]	GAM_R_TH_14[2]	GAM_R_TH_14[1]	GAM_R_TH_14[0]
		GAM_R_TH_15[7]	GAM_R_TH_15[6]	GAM_R_TH_15[5]	GAM_R_TH_15[4]	GAM_R_TH_15[3]	GAM_R_TH_15[2]	GAM_R_TH_15[1]	GAM_R_TH_15[0]
	GAM_R_AREA5	GAM_R_TH_16[7]	GAM_R_TH_16[6]	GAM_R_TH_16[5]	GAM_R_TH_16[4]	GAM_R_TH_16[3]	GAM_R_TH_16[2]	GAM_R_TH_16[1]	GAM_R_TH_16[0]
		GAM_R_TH_17[7]	GAM_R_TH_17[6]	GAM_R_TH_17[5]	GAM_R_TH_17[4]	GAM_R_TH_17[3]	GAM_R_TH_17[2]	GAM_R_TH_17[1]	GAM_R_TH_17[0]
		GAM_R_TH_18[7]	GAM_R_TH_18[6]	GAM_R_TH_18[5]	GAM_R_TH_18[4]	GAM_R_TH_18[3]	GAM_R_TH_18[2]	GAM_R_TH_18[1]	GAM_R_TH_18[0]
		GAM_R_TH_19[7]	GAM_R_TH_19[6]	GAM_R_TH_19[5]	GAM_R_TH_19[4]	GAM_R_TH_19[3]	GAM_R_TH_19[2]	GAM_R_TH_19[1]	GAM_R_TH_19[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Video display controller 5	GAM_R_AREA6	GAM_R_TH_20[7]	GAM_R_TH_20[6]	GAM_R_TH_20[5]	GAM_R_TH_20[4]	GAM_R_TH_20[3]	GAM_R_TH_20[2]	GAM_R_TH_20[1]	GAM_R_TH_20[0]
		GAM_R_TH_21[7]	GAM_R_TH_21[6]	GAM_R_TH_21[5]	GAM_R_TH_21[4]	GAM_R_TH_21[3]	GAM_R_TH_21[2]	GAM_R_TH_21[1]	GAM_R_TH_21[0]
		GAM_R_TH_22[7]	GAM_R_TH_22[6]	GAM_R_TH_22[5]	GAM_R_TH_22[4]	GAM_R_TH_22[3]	GAM_R_TH_22[2]	GAM_R_TH_22[1]	GAM_R_TH_22[0]
		GAM_R_TH_23[7]	GAM_R_TH_23[6]	GAM_R_TH_23[5]	GAM_R_TH_23[4]	GAM_R_TH_23[3]	GAM_R_TH_23[2]	GAM_R_TH_23[1]	GAM_R_TH_23[0]
	GAM_R_AREA7	GAM_R_TH_24[7]	GAM_R_TH_24[6]	GAM_R_TH_24[5]	GAM_R_TH_24[4]	GAM_R_TH_24[3]	GAM_R_TH_24[2]	GAM_R_TH_24[1]	GAM_R_TH_24[0]
		GAM_R_TH_25[7]	GAM_R_TH_25[6]	GAM_R_TH_25[5]	GAM_R_TH_25[4]	GAM_R_TH_25[3]	GAM_R_TH_25[2]	GAM_R_TH_25[1]	GAM_R_TH_25[0]
		GAM_R_TH_26[7]	GAM_R_TH_26[6]	GAM_R_TH_26[5]	GAM_R_TH_26[4]	GAM_R_TH_26[3]	GAM_R_TH_26[2]	GAM_R_TH_26[1]	GAM_R_TH_26[0]
		GAM_R_TH_27[7]	GAM_R_TH_27[6]	GAM_R_TH_27[5]	GAM_R_TH_27[4]	GAM_R_TH_27[3]	GAM_R_TH_27[2]	GAM_R_TH_27[1]	GAM_R_TH_27[0]
	GAM_R_AREA8	GAM_R_TH_28[7]	GAM_R_TH_28[6]	GAM_R_TH_28[5]	GAM_R_TH_28[4]	GAM_R_TH_28[3]	GAM_R_TH_28[2]	GAM_R_TH_28[1]	GAM_R_TH_28[0]
		GAM_R_TH_29[7]	GAM_R_TH_29[6]	GAM_R_TH_29[5]	GAM_R_TH_29[4]	GAM_R_TH_29[3]	GAM_R_TH_29[2]	GAM_R_TH_29[1]	GAM_R_TH_29[0]
		GAM_R_TH_30[7]	GAM_R_TH_30[6]	GAM_R_TH_30[5]	GAM_R_TH_30[4]	GAM_R_TH_30[3]	GAM_R_TH_30[2]	GAM_R_TH_30[1]	GAM_R_TH_30[0]
		GAM_R_TH_31[7]	GAM_R_TH_31[6]	GAM_R_TH_31[5]	GAM_R_TH_31[4]	GAM_R_TH_31[3]	GAM_R_TH_31[2]	GAM_R_TH_31[1]	GAM_R_TH_31[0]
	TCON_UPDATE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	TCON_VEN
	TCON_TIM	-	-	-	-	-	TCON_HALF[10]	TCON_HALF[9]	TCON_HALF[8]
		TCON_HALF[7]	TCON_HALF[6]	TCON_HALF[5]	TCON_HALF[4]	TCON_HALF[3]	TCON_HALF[2]	TCON_HALF[1]	TCON_HALF[0]
		-	-	-	-	-	TCON_OFFSET [10]	TCON_OFFSET[9]	TCON_OFFSET[8]
		TCON_OFFSET[7]	TCON_OFFSET[6]	TCON_OFFSET[5]	TCON_OFFSET[4]	TCON_OFFSET[3]	TCON_OFFSET[2]	TCON_OFFSET[1]	TCON_OFFSET[0]
	TCON_TIM_STVA1	-	-	-	-	-	TCON_STVA_VS [10]	TCON_STVA_VS [9]	TCON_STVA_VS [8]
		TCON_STVA_VS [7]	TCON_STVA_VS [6]	TCON_STVA_VS [5]	TCON_STVA_VS [4]	TCON_STVA_VS [3]	TCON_STVA_VS [2]	TCON_STVA_VS [1]	TCON_STVA_VS [0]
		-	-	-	-	-	TCON_STVA_VW[10]	TCON_STVA_VW[9]	TCON_STVA_VW[8]
		TCON_STVA_VW[7]	TCON_STVA_VW[6]	TCON_STVA_VW[5]	TCON_STVA_VW[4]	TCON_STVA_VW[3]	TCON_STVA_VW[2]	TCON_STVA_VW[1]	TCON_STVA_VW[0]
	TCON_TIM_STVA2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	TCON_STVA_INV	-	TCON_STVA_ SEL[2]	TCON_STVA_ SEL[1]	TCON_STVA_ SEL[0]
	TCON_TIM_STVB1	-	-	-	-	-	TCON_STVB_VS [10]	TCON_STVB_VS [9]	TCON_STVB_VS [8]
		TCON_STVB_VS [7]	TCON_STVB_VS [6]	TCON_STVB_VS [5]	TCON_STVB_VS [4]	TCON_STVB_VS [3]	TCON_STVB_VS [2]	TCON_STVB_VS [1]	TCON_STVB_VS [0]
		-	-	-	-	-	TCON_STVB_VW[10]	TCON_STVB_VW[9]	TCON_STVB_VW[8]
		TCON_STVB_VW[7]	TCON_STVB_VW[6]	TCON_STVB_VW[5]	TCON_STVB_VW[4]	TCON_STVB_VW[3]	TCON_STVB_VW[2]	TCON_STVB_VW[1]	TCON_STVB_VW[0]
	TCON_TIM_STVB2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	TCON_STVB_INV	-	TCON_STVB_ SEL[2]	TCON_STVB_ SEL[1]	TCON_STVB_ SEL[0]
	TCON_TIM_STH1	-	-	-	-	-	TCON_STH_HS [10]	TCON_STH_HS[9]	TCON_STH_HS[8]
		TCON_STH_HS[7]	TCON_STH_HS[6]	TCON_STH_HS[5]	TCON_STH_HS[4]	TCON_STH_HS[3]	TCON_STH_HS[2]	TCON_STH_HS[1]	TCON_STH_HS[0]
		-	-	-	-	-	TCON_STH_HW [10]	TCON_STH_HW [9]	TCON_STH_HW [8]
		TCON_STH_HW [7]	TCON_STH_HW [6]	TCON_STH_HW [5]	TCON_STH_HW [4]	TCON_STH_HW [3]	TCON_STH_HW [2]	TCON_STH_HW [1]	TCON_STH_HW [0]
	TCON_TIM_STH2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	TCON_STH_HS_SEL
		-	-	-	TCON_STH_INV	-	TCON_STH_SEL [2]	TCON_STH_SEL [1]	TCON_STH_SEL [0]
	TCON_TIM_STB1	-	-	-	-	-	TCON_STB_HS [10]	TCON_STB_HS[9]	TCON_STB_HS[8]
		TCON_STB_HS[7]	TCON_STB_HS[6]	TCON_STB_HS[5]	TCON_STB_HS[4]	TCON_STB_HS[3]	TCON_STB_HS[2]	TCON_STB_HS[1]	TCON_STB_HS[0]
		-	-	-	-	-	TCON_STB_HW [10]	TCON_STB_HW [9]	TCON_STB_HW [8]
		TCON_STB_HW [7]	TCON_STB_HW [6]	TCON_STB_HW [5]	TCON_STB_HW [4]	TCON_STB_HW [3]	TCON_STB_HW [2]	TCON_STB_HW [1]	TCON_STB_HW [0]
	TCON_TIM_STB2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	TCON_STB_HS_SEL
		-	-	-	TCON_STB_INV	-	TCON_STB_SEL [2]	TCON_STB_SEL [1]	TCON_STB_SEL [0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Video display controller 5	TCON_TIM_CPV1	-	-	-	-	-	TCON_CPV_HS [10]	TCON_CPV_HS[9]	TCON_CPV_HS[8]
		TCON_CPV_HS[7]	TCON_CPV_HS[6]	TCON_CPV_HS[5]	TCON_CPV_HS[4]	TCON_CPV_HS[3]	TCON_CPV_HS[2]	TCON_CPV_HS[1]	TCON_CPV_HS[0]
		-	-	-	-	-	TCON_CPV_HW [10]	TCON_CPV_HW [9]	TCON_CPV_HW [8]
		TCON_CPV_HW [7]	TCON_CPV_HW [6]	TCON_CPV_HW [5]	TCON_CPV_HW [4]	TCON_CPV_HW [3]	TCON_CPV_HW [2]	TCON_CPV_HW [1]	TCON_CPV_HW [0]
	TCON_TIM_CPV2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	TCON_CPV_HS_SEL
		-	-	-	TCON_CPV_INV	-	TCON_CPV_SEL [2]	TCON_CPV_SEL [1]	TCON_CPV_SEL [0]
	TCON_TIM_POLA1	-	-	-	-	-	TCON_POLA_HS[10]	TCON_POLA_HS[9]	TCON_POLA_HS[8]
		TCON_POLA_HS[7]	TCON_POLA_HS[6]	TCON_POLA_HS[5]	TCON_POLA_HS[4]	TCON_POLA_HS[3]	TCON_POLA_HS[2]	TCON_POLA_HS[1]	TCON_POLA_HS[0]
		-	-	-	-	-	TCON_POLA_ HW[10]	TCON_POLA_ HW[9]	TCON_POLA_ HW[8]
		TCON_POLA_ HW[7]	TCON_POLA_ HW[6]	TCON_POLA_ HW[5]	TCON_POLA_ HW[4]	TCON_POLA_ HW[3]	TCON_POLA_ HW[2]	TCON_POLA_ HW[1]	TCON_POLA_ HW[0]
	TCON_TIM_POLA2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	TCON_POLA_MD[1]	TCON_POLA_MD[0]	-	-	-	TCON_POLA_HS_ SEL
		-	-	-	TCON_POLA_ INV	-	TCON_POLA_ SEL[2]	TCON_POLA_ SEL[1]	TCON_POLA_ SEL[0]
	TCON_TIM_POLB1	-	-	-	-	-	TCON_POLB_HS[10]	TCON_POLB_HS[9]	TCON_POLB_HS[8]
		TCON_POLB_HS[7]	TCON_POLB_HS[6]	TCON_POLB_HS[5]	TCON_POLB_HS[4]	TCON_POLB_HS[3]	TCON_POLB_HS[2]	TCON_POLB_HS[1]	TCON_POLB_HS[0]
		-	-	-	-	-	TCON_POLB_ HW[10]	TCON_POLB_ HW[9]	TCON_POLB_ HW[8]
		TCON_POLB_ HW[7]	TCON_POLB_ HW[6]	TCON_POLB_ HW[5]	TCON_POLB_ HW[4]	TCON_POLB_ HW[3]	TCON_POLB_ HW[2]	TCON_POLB_ HW[1]	TCON_POLB_ HW[0]
	TCON_TIM_POLB2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	TCON_POLB_MD[1]	TCON_POLB_MD[0]	-	-	-	TCON_POLB_HS_ SEL
		-	-	-	TCON_POLB_ INV	-	TCON_POLB_ SEL[2]	TCON_POLB_ SEL[1]	TCON_POLB_ SEL[0]
	TCON_TIM_DE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	TCON_DE_INV
	OUT_UPDATE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	OUTCNT_VEN
	OUT_SET	-	-	-	OUT_ENDIAN_ON	-	-	-	OUT_SWAP_ON
		-	-	-	-	-	-	-	-
		-	-	OUT_FORMAT[1]	OUT_FORMAT[0]	-	-	OUT_FRQ_SEL[1]	OUT_FRQ_SEL[0]
		-	-	-	OUT_DIR_SEL	-	-	OUT_PHASE[1]	OUT_PHASE[0]
	OUT_BRIGHT1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	PBRT_G[9]	PBRT_G[8]
		PBRT_G[7]	PBRT_G[6]	PBRT_G[5]	PBRT_G[4]	PBRT_G[3]	PBRT_G[2]	PBRT_G[1]	PBRT_G[0]
	OUT_BRIGHT2	-	-	-	-	-	-	PBRT_B[9]	PBRT_B[8]
		PBRT_B[7]	PBRT_B[6]	PBRT_B[5]	PBRT_B[4]	PBRT_B[3]	PBRT_B[2]	PBRT_B[1]	PBRT_B[0]
		-	-	-	-	-	-	PBRT_R[9]	PBRT_R[8]
		PBRT_R[7]	PBRT_R[6]	PBRT_R[5]	PBRT_R[4]	PBRT_R[3]	PBRT_R[2]	PBRT_R[1]	PBRT_R[0]
	OUT_CONTRAST	-	-	-	-	-	-	-	-
		CONT_G[7]	CONT_G[6]	CONT_G[5]	CONT_G[4]	CONT_G[3]	CONT_G[2]	CONT_G[1]	CONT_G[0]
		CONT_B[7]	CONT_B[6]	CONT_B[5]	CONT_B[4]	CONT_B[3]	CONT_B[2]	CONT_B[1]	CONT_B[0]
		CONT_R[7]	CONT_R[6]	CONT_R[5]	CONT_R[4]	CONT_R[3]	CONT_R[2]	CONT_R[1]	CONT_R[0]
	OUT_PDTHA	-	-	-	-	-	-	-	-
		-	-	PDTH_SEL[1]	PDTH_SEL[0]	-	-	PDTH_FORMAT [1]	PDTH_FORMAT [0]
		-	-	PDTH_PA[1]	PDTH_PA[0]	-	-	PDTH_PB[1]	PDTH_PB[0]
		-	-	PDTH_PC[1]	PDTH_PC[0]	-	-	PDTH_PD[1]	PDTH_PD[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Video display controller 5	OUT_CLK_PHASE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	OUTCNT_FRONT_GAM	-	-	-	OUTCNT_LCD_EDGE
		-	OUTCNT_STVA_EDGE	OUTCNT_STVB_EDGE	OUTCNT_STH_EDGE	OUTCNT_STB_EDGE	OUTCNT_CPV_EDGE	OUTCNT_POLA_EDGE	OUTCNT_POLB_EDGE
	SYSCNT_INT1	-	-	-	INT_STA7	-	-	-	INT_STA6
		-	-	-	INT_STA5	-	-	-	INT_STA4
		-	-	-	INT_STA3	-	-	-	INT_STA2
		-	-	-	INT_STA1	-	-	-	INT_STA0
	SYSCNT_INT2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	INT_STA9	-	-	-	INT_STA8
	SYSCNT_INT4	-	-	-	INT_OUT7_ON	-	-	-	INT_OUT6_ON
		-	-	-	INT_OUT5_ON	-	-	-	INT_OUT4_ON
		-	-	-	INT_OUT3_ON	-	-	-	INT_OUT2_ON
		-	-	-	INT_OUT1_ON	-	-	-	INT_OUT0_ON
	SYSCNT_INT5	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	INT_OUT9_ON	-	-	-	INT_OUT8_ON
	SYSCNT_PANEL_CLK	-	-	PANEL_ICKSEL[1]	PANEL_ICKSEL[0]	-	-	-	PANEL_ICKEN
		-	-	PANEL_DCDR[5]	PANEL_DCDR[4]	PANEL_DCDR[3]	PANEL_DCDR[2]	PANEL_DCDR[1]	PANEL_DCDR[0]
	SYSCNT_CLUT	-	-	-	GR3_CLT_SEL_ST	-	-	-	GR2_CLT_SEL_ST
		-	-	-	-	-	-	-	GR0_CLT_SEL_ST
Capture engine unit	CAPSR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	CPKIL
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	CE
	CAPCR	FDRP[7]	FDRP[6]	FDRP[5]	FDRP[4]	FDRP[3]	FDRP[2]	FDRP[1]	FDRP[0]
		-	-	MTCM[1]	MTCM[0]	-	-	-	CTNCP
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	CAMCR	-	-	-	-	VDSEL*6	HDSEL*6	FLDSEL*6	DSEL*6
		-	-	-	-	-	-	-	FLDPOL
		-	-	-	DTIF	-	-	DTARY[1]	DTARY[0]
		-	-	JPG[1]	JPG[0]	-	-	VDPOL	HDPOL
	CMCYR	-	-	VCYL[13]	VCYL[12]	VCYL[11]	VCYL[10]	VCYL[9]	VCYL[8]
		VCYL[7]	VCYL[6]	VCYL[5]	VCYL[4]	VCYL[3]	VCYL[2]	VCYL[1]	VCYL[0]
		-	-	HCYL[13]	HCYL[12]	HCYL[11]	HCYL[10]	HCYL[9]	HCYL[8]
		HCYL[7]	HCYL[6]	HCYL[5]	HCYL[4]	HCYL[3]	HCYL[2]	HCYL[1]	HCYL[0]
	CAMOR	-	-	-	-	VOFST[11]	VOFST[10]	VOFST[9]	VOFST[8]
		VOFST[7]	VOFST[6]	VOFST[5]	VOFST[4]	VOFST[3]	VOFST[2]	VOFST[1]	VOFST[0]
		-	-	-	HOFST[12]	HOFST[11]	HOFST[10]	HOFST[9]	HOFST[8]
		HOFST[7]	HOFST[6]	HOFST[5]	HOFST[4]	HOFST[3]	HOFST[2]	HOFST[1]	HOFST[0]
	CAPWR	-	-	-	-	VWDTH[11]	VWDTH[10]	VWDTH[9]	VWDTH[8]
		VWDTH[7]	VWDTH[6]	VWDTH[5]	VWDTH[4]	VWDTH[3]	VWDTH[2]	VWDTH[1]	VWDTH[0]
		-	-	-	HWDTH[12]	HWDTH[11]	HWDTH[10]	HWDTH[9]	HWDTH[8]
		HWDTH[7]	HWDTH[6]	HWDTH[5]	HWDTH[4]	HWDTH[3]	HWDTH[2]	HWDTH[1]	HWDTH[0]
	CAIFR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	IFS
		-	-	-	CIM	-	-	FCI[1]	FCI[0]
	CRCNTR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	RVS	-	-	RS	RC

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Capture engine unit	CRCMPR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	RA
	CFLCR	VMANT[3]	VMANT[2]	VMANT[1]	VMANT[0]	VFRAC[11]	VFRAC[10]	VFRAC[9]	VFRAC[8]
		VFRAC[7]	VFRAC[6]	VFRAC[5]	VFRAC[4]	VFRAC[3]	VFRAC[2]	VFRAC[1]	VFRAC[0]
		HMANT[3]	HMANT[2]	HMANT[1]	HMANT[0]	HFRAC[11]	HFRAC[10]	HFRAC[9]	HFRAC[8]
		HFRAC[7]	HFRAC[6]	HFRAC[5]	HFRAC[4]	HFRAC[3]	HFRAC[2]	HFRAC[1]	HFRAC[0]
	CFSZR	-	-	-	-	VFCLP[11]	VFCLP[10]	VFCLP[9]	VFCLP[8]
		VFCLP[7]	VFCLP[6]	VFCLP[5]	VFCLP[4]	VFCLP[3]	VFCLP[2]	VFCLP[1]	VFCLP[0]
		-	-	-	-	HFCLP[11]	HFCLP[10]	HFCLP[9]	HFCLP[8]
		HFCLP[7]	HFCLP[6]	HFCLP[5]	HFCLP[4]	HFCLP[3]	HFCLP[2]	HFCLP[1]	HFCLP[0]
	CDWDR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	CHDW[12]	CHDW[11]	CHDW[10]	CHDW[9]	CHDW[8]
		CHDW[7]	CHDW[6]	CHDW[5]	CHDW[4]	CHDW[3]	CHDW[2]	CHDW[1]	CHDW[0]
	CDAYR	CAYR[31]	CAYR[30]	CAYR[29]	CAYR[28]	CAYR[27]	CAYR[26]	CAYR[25]	CAYR[24]
		CAYR[23]	CAYR[22]	CAYR[21]	CAYR[20]	CAYR[19]	CAYR[18]	CAYR[17]	CAYR[16]
		CAYR[15]	CAYR[14]	CAYR[13]	CAYR[12]	CAYR[11]	CAYR[10]	CAYR[9]	CAYR[8]
		CAYR[7]	CAYR[6]	CAYR[5]	CAYR[4]	CAYR[3]	CAYR[2]	CAYR[1]	CAYR[0]
	CDACR	CACR[31]	CACR[30]	CACR[29]	CACR[28]	CACR[27]	CACR[26]	CACR[25]	CACR[24]
		CACR[23]	CACR[22]	CACR[21]	CACR[20]	CACR[19]	CACR[18]	CACR[17]	CACR[16]
		CACR[15]	CACR[14]	CACR[13]	CACR[12]	CACR[11]	CACR[10]	CACR[9]	CACR[8]
		CACR[7]	CACR[6]	CACR[5]	CACR[4]	CACR[3]	CACR[2]	CACR[1]	CACR[0]
	CDBYR	CBYR[31]	CBYR[30]	CBYR[29]	CBYR[28]	CBYR[27]	CBYR[26]	CBYR[25]	CBYR[24]
		CBYR[23]	CBYR[22]	CBYR[21]	CBYR[20]	CBYR[19]	CBYR[18]	CBYR[17]	CBYR[16]
		CBYR[15]	CBYR[14]	CBYR[13]	CBYR[12]	CBYR[11]	CBYR[10]	CBYR[9]	CBYR[8]
		CBYR[7]	CBYR[6]	CBYR[5]	CBYR[4]	CBYR[3]	CBYR[2]	CBYR[1]	CBYR[0]
	CDBCR	CBCR[31]	CBCR[30]	CBCR[29]	CBCR[28]	CBCR[27]	CBCR[26]	CBCR[25]	CBCR[24]
		CBCR[23]	CBCR[22]	CBCR[21]	CBCR[20]	CBCR[19]	CBCR[18]	CBCR[17]	CBCR[16]
		CBCR[15]	CBCR[14]	CBCR[13]	CBCR[12]	CBCR[11]	CBCR[10]	CBCR[9]	CBCR[8]
		CBCR[7]	CBCR[6]	CBCR[5]	CBCR[4]	CBCR[3]	CBCR[2]	CBCR[1]	CBCR[0]
	CBDSR	-	-	-	-	-	-	-	-
		-	CBVS[22]	CBVS[21]	CBVS[20]	CBVS[19]	CBVS[18]	CBVS[17]	CBVS[16]
		CBVS[15]	CBVS[14]	CBVS[13]	CBVS[12]	CBVS[11]	CBVS[10]	CBVS[9]	CBVS[8]
		CBVS[7]	CBVS[6]	CBVS[5]	CBVS[4]	CBVS[3]	CBVS[2]	CBVS[1]	CBVS[0]
	CFWCR	FWV[26]	FWV[25]	FWV[24]	FWV[23]	FWV[22]	FWV[21]	FWV[20]	FWV[19]
		FWV[18]	FWV[17]	FWV[16]	FWV[15]	FWV[14]	FWV[13]	FWV[12]	FWV[11]
		FWV[10]	FWV[9]	FWV[8]	FWV[7]	FWV[6]	FWV[5]	FWV[4]	FWV[3]
		FWV[2]	FWV[1]	FWV[0]	-	-	-	-	FWE
	CLFCR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	LPF
	CDOCR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	CBE
		-	-	-	-	-	-	-	-
		-	-	-	CDS	-	COLS	COWS	COBS
	CEIER	-	-	-	-	-	-	NVDIE	NHDIE
		FWFIE	-	-	VBPIE	-	IGVSIE	IGHSIE	CDTOFIE
		CPBE4IE	CPBE3IE	CPBE2IE	CPBE1IE	-	-	VDIE	HDIE
		-	-	-	IGRWIE	-	-	CFEIE	CPEIE
	CETCR	-	-	-	-	-	-	NVD	NHD
		FWF	-	-	VBP	-	IGVS	IGHS	CDTOF
		CPBE4	CPBE3	CPBE2	CPBE1	-	-	VD	HD
		-	-	-	IGRW	-	-	CFE	CPE
	CSTSR	-	-	-	-	-	-	-	CRST
		-	-	-	-	-	-	-	CPFLD
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	CPTON

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Capture engine unit	CDSSR	CDSS[31]	CDSS[30]	CDSS[29]	CDSS[28]	CDSS[27]	CDSS[26]	CDSS[25]	CDSS[24]
		CDSS[23]	CDSS[22]	CDSS[21]	CDSS[20]	CDSS[19]	CDSS[18]	CDSS[17]	CDSS[16]
		CDSS[15]	CDSS[14]	CDSS[13]	CDSS[12]	CDSS[11]	CDSS[10]	CDSS[9]	CDSS[8]
		CDSS[7]	CDSS[6]	CDSS[5]	CDSS[4]	CDSS[3]	CDSS[2]	CDSS[1]	CDSS[0]
	CDAYR2	CAYR2[31]	CAYR2[30]	CAYR2[29]	CAYR2[28]	CAYR2[27]	CAYR2[26]	CAYR2[25]	CAYR2[24]
		CAYR2[23]	CAYR2[22]	CAYR2[21]	CAYR2[20]	CAYR2[19]	CAYR2[18]	CAYR2[17]	CAYR2[16]
		CAYR2[15]	CAYR2[14]	CAYR2[13]	CAYR2[12]	CAYR2[11]	CAYR2[10]	CAYR2[9]	CAYR2[8]
		CAYR2[7]	CAYR2[6]	CAYR2[5]	CAYR2[4]	CAYR2[3]	CAYR2[2]	CAYR2[1]	CAYR2[0]
	CDACR2	CACR2[31]	CACR2[30]	CACR2[29]	CACR2[28]	CACR2[27]	CACR2[26]	CACR2[25]	CACR2[24]
		CACR2[23]	CACR2[22]	CACR2[21]	CACR2[20]	CACR2[19]	CACR2[18]	CACR2[17]	CACR2[16]
		CACR2[15]	CACR2[14]	CACR2[13]	CACR2[12]	CACR2[11]	CACR2[10]	CACR2[9]	CACR2[8]
		CACR2[7]	CACR2[6]	CACR2[5]	CACR2[4]	CACR2[3]	CACR2[2]	CACR2[1]	CACR2[0]
	CDBYR2	CBYR2[31]	CBYR2[30]	CBYR2[29]	CBYR2[28]	CBYR2[27]	CBYR2[26]	CBYR2[25]	CBYR2[24]
		CBYR2[23]	CBYR2[22]	CBYR2[21]	CBYR2[20]	CBYR2[19]	CBYR2[18]	CBYR2[17]	CBYR2[16]
		CBYR2[15]	CBYR2[14]	CBYR2[13]	CBYR2[12]	CBYR2[11]	CBYR2[10]	CBYR2[9]	CBYR2[8]
		CBYR2[7]	CBYR2[6]	CBYR2[5]	CBYR2[4]	CBYR2[3]	CBYR2[2]	CBYR2[1]	CBYR2[0]
	CDBCR2	CBCR2[31]	CBCR2[30]	CBCR2[29]	CBCR2[28]	CBCR2[27]	CBCR2[26]	CBCR2[25]	CBCR2[24]
		CBCR2[23]	CBCR2[22]	CBCR2[21]	CBCR2[20]	CBCR2[19]	CBCR2[18]	CBCR2[17]	CBCR2[16]
		CBCR2[15]	CBCR2[14]	CBCR2[13]	CBCR2[12]	CBCR2[11]	CBCR2[10]	CBCR2[9]	CBCR2[8]
		CBCR2[7]	CBCR2[6]	CBCR2[5]	CBCR2[4]	CBCR2[3]	CBCR2[2]	CBCR2[1]	CBCR2[0]
SCUX	IPCIR_IPC0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT
	IPSLR_IPC0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	IPC_PASS_SEL[2]	IPC_PASS_SEL[1]	IPC_PASS_SEL[0]
	IPCIR_IPC0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT
	IPSLR_IPC0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	IPC_PASS_SEL[2]	IPC_PASS_SEL[1]	IPC_PASS_SEL[0]
	IPCIR_IPC0_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT
	IPSLR_IPC0_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	IPC_PASS_SEL[2]	IPC_PASS_SEL[1]	IPC_PASS_SEL[0]
	IPCIR_IPC0_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT
	IPSLR_IPC0_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	IPC_PASS_SEL[2]	IPC_PASS_SEL[1]	IPC_PASS_SEL[0]
	OPCIR_OPC0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT
	OPSLR_OPC0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	OPC_PASS_SEL[2]	OPC_PASS_SEL[1]	OPC_PASS_SEL[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
SCUX	OPCIR_OPC0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT
	OPSLR_OPC0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	OPC_PASS_SEL [2]	OPC_PASS_SEL [1]	OPC_PASS_SEL [0]
	OPCIR_OPC0_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT
	OPSLR_OPC0_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	OPC_PASS_SEL [2]	OPC_PASS_SEL [1]	OPC_PASS_SEL [0]
	OPCIR_OPC0_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT
	OPSLR_OPC0_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	OPC_PASS_SEL [2]	OPC_PASS_SEL [1]	OPC_PASS_SEL [0]
	FFDIR_FFD0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT
	FDAIR_FFD0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CHNUM[3]	CHNUM[2]	CHNUM[1]	CHNUM[0]
	DRQSR_FFD0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	SIZE[3]	SIZE[2]	SIZE[1]	SIZE[0]
	FFDPR_FFD0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	PASS[1]	PASS[0]
	FFDBR_FFD0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	BOOT
	DEVMR_FFD0_0	DEVMUF	DEVMOF	DEVMOL	DEVMIUF	-	-	-	-
		-	-	-	-	-	-	-	-
		DEVMRQ	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	DEVCR_FFD0_0	DEVCUF	DEVCOF	DEVCOL	DEVCIUF	-	-	-	-
		-	-	-	-	-	-	-	-
		DEVCRQ	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	FFDIR_FFD0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
SCUX	FDAIR_FFD0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CHNUM[3]	CHNUM[2]	CHNUM[1]	CHNUM[0]
	DRQSR_FFD0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	SIZE[3]	SIZE[2]	SIZE[1]	SIZE[0]
	FFDPR_FFD0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	PASS[1]	PASS[0]
	FFDBR_FFD0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	BOOT
	DEVMR_FFD0_1	DEVMUF	DEVMOF	DEVMOL	DEVMIUF	-	-	-	-
		-	-	-	-	-	-	-	-
		DEVMRQ	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	DEVCR_FFD0_1	DEVCUF	DEVCOF	DEVCOL	DEVCIUF	-	-	-	-
		-	-	-	-	-	-	-	-
		DEVCRQ	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	FFDIR_FFD0_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT
	FDAIR_FFD0_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CHNUM[3]	CHNUM[2]	CHNUM[1]	CHNUM[0]
	DRQSR_FFD0_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	SIZE[3]	SIZE[2]	SIZE[1]	SIZE[0]
	FFDPR_FFD0_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	PASS[1]	PASS[0]
	FFDBR_FFD0_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	BOOT
	DEVMR_FFD0_2	DEVMUF	DEVMOF	DEVMOL	DEVMIUF	-	-	-	-
		-	-	-	-	-	-	-	-
		DEVMRQ	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	DEVCR_FFD0_2	DEVCUF	DEVCOF	DEVCOL	DEVCIUF	-	-	-	-
		-	-	-	-	-	-	-	-
		DEVCRQ	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	FFDIR_FFD0_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT
	FDAIR_FFD0_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CHNUM[3]	CHNUM[2]	CHNUM[1]	CHNUM[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
SCUX	DRQSR_FFD0_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	SIZE[3]	SIZE[2]	SIZE[1]	SIZE[0]
	FFDPR_FFD0_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	PASS[1]	PASS[0]
	FFDBR_FFD0_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	BOOT
	DEVMR_FFD0_3	DEVMUF	DEVMOF	DEVMOL	DEVMIUF	-	-	-	-
		-	-	-	-	-	-	-	-
		DEVMRQ	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	DEVCR_FFD0_3	DEVCUF	DEVCOF	DEVCOL	DEVCIUF	-	-	-	-
		-	-	-	-	-	-	-	-
		DEVCRQ	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	FFUIR_FFU0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT
	FUAIR_FFU0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CHNUM[3]	CHNUM[2]	CHNUM[1]	CHNUM[0]
	URQSR_FFU0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	SIZE[3]	SIZE[2]	SIZE[1]	SIZE[0]
	FFUPR_FFU0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	PASS[1]	PASS[0]
	UEVMR_FFU0_0	UEVMUF	UEVMOF	UEVMOL	-	-	-	-	-
		-	-	-	-	-	-	-	-
		UEVMRQ	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	UEVCR_FFU0_0	UEVCUF	UEVCOF	UEVCOL	-	-	-	-	-
		-	-	-	-	-	-	-	-
		UEVCRQ	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	FFUIR_FFU0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT
	FUAIR_FFU0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CHNUM[3]	CHNUM[2]	CHNUM[1]	CHNUM[0]
	URQSR_FFU0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	SIZE[3]	SIZE[2]	SIZE[1]	SIZE[0]
	FFUPR_FFU0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	PASS[1]	PASS[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
SCUX	UEVMR_FFU0_1	UEVMUF	UEVMOF	UEVMOL	-	-	-	-	-
		-	-	-	-	-	-	-	-
		UEVMRQ	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	UEVCR_FFU0_1	UEVCUF	UEVCOF	UEVCOL	-	-	-	-	-
		-	-	-	-	-	-	-	-
		UEVCRQ	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	FFUIR_FFU0_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT
	FUAIR_FFU0_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CHNUM[3]	CHNUM[2]	CHNUM[1]	CHNUM[0]
	URQSR_FFU0_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	SIZE[3]	SIZE[2]	SIZE[1]	SIZE[0]
	FFUPR_FFU0_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	PASS[1]	PASS[0]
	UEVMR_FFU0_2	UEVMUF	UEVMOF	UEVMOL	-	-	-	-	-
		-	-	-	-	-	-	-	-
		UEVMRQ	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	UEVCR_FFU0_2	UEVCUF	UEVCOF	UEVCOL	-	-	-	-	-
		-	-	-	-	-	-	-	-
		UEVCRQ	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	FFUIR_FFU0_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT
	FUAIR_FFU0_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CHNUM[3]	CHNUM[2]	CHNUM[1]	CHNUM[0]
	URQSR_FFU0_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	SIZE[3]	SIZE[2]	SIZE[1]	SIZE[0]
	FFUPR_FFU0_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	PASS[1]	PASS[0]
	UEVMR_FFU0_3	UEVMUF	UEVMOF	UEVMOL	-	-	-	-	-
		-	-	-	-	-	-	-	-
		UEVMRQ	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	UEVCR_FFU0_3	UEVCUF	UEVCOF	UEVCOL	-	-	-	-	-
		-	-	-	-	-	-	-	-
		UEVCRQ	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	SRCIR0_2SRC0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
SCUX	SADIR0_2SRC0_0	-	-	-	-	-	-	-	-
		-	-	-	OTBL[4]	OTBL[3]	OTBL[2]	OTBL[1]	OTBL[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	CHNUM[3]	CHNUM[2]	CHNUM[1]	CHNUM[0]
	SRCBR0_2SRC0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	BYPASS
	IFSCR0_2SRC0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INTIFSEN
	IFSVR0_2SRC0_0	-	-	-	-	INTIFS[27]	INTIFS[26]	INTIFS[25]	INTIFS[24]
		INTIFS[23]	INTIFS[22]	INTIFS[21]	INTIFS[20]	INTIFS[19]	INTIFS[18]	INTIFS[17]	INTIFS[16]
		INTIFS[15]	INTIFS[14]	INTIFS[13]	INTIFS[11]	INTIFS[11]	INTIFS[10]	INTIFS[9]	INTIFS[8]
		INTIFS[7]	INTIFS[6]	INTIFS[5]	INTIFS[4]	INTIFS[3]	INTIFS[2]	INTIFS[1]	INTIFS[0]
	SRCCR0_2SRC0_0	-	-	-	-	-	-	-	-
		-	-	-	WATMD	-	-	-	-
		-	-	-	BUFMD	-	-	-	-
		-	-	-	-	-	-	-	SRCMD
	MNFSR0_2SRC0_0	-	-	-	-	MINFS[27]	MINFS[26]	MINFS[25]	MINFS[24]
		MINFS[23]	MINFS[22]	MINFS[21]	MINFS[20]	MINFS[19]	MINFS[18]	MINFS[17]	MINFS[16]
		MINFS[15]	MINFS[14]	MINFS[13]	MINFS[11]	MINFS[11]	MINFS[10]	MINFS[9]	MINFS[8]
		MINFS[7]	MINFS[6]	MINFS[5]	MINFS[4]	MINFS[3]	MINFS[2]	MINFS[1]	MINFS[0]
	BFSSR0_2SRC0_0	-	-	-	-	-	-	BUFDATA[9]	BUFDATA[8]
		BUFDATA[7]	BUFDATA[6]	BUFDATA[5]	BUFDATA[4]	BUFDATA[3]	BUFDATA[1]	BUFDATA[1]	BUFDATA[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	BUFIN[3]	BUFIN[2]	BUFIN[1]	BUFIN[0]
	SC2SR0_2SRC0_0	SRCWSTS	SC2MUTE	SC2STS[1]	SC2STS[0]	SC2FSI[27]	SC2FSI[26]	SC2FSI[25]	SC2FSI[24]
		SC2FSI[23]	SC2FSI[22]	SC2FSI[21]	SC2FSI[20]	SC2FSI[19]	SC2FSI[18]	SC2FSI[17]	SC2FSI[16]
		SC2FSI[15]	SC2FSI[14]	SC2FSI[13]	SC2FSI[11]	SC2FSI[11]	SC2FSI[10]	SC2FSI[9]	SC2FSI[8]
		SC2FSI[7]	SC2FSI[6]	SC2FSI[5]	SC2FSI[4]	SC2FSI[3]	SC2FSI[2]	SC2FSI[1]	SC2FSI[0]
	WATSR0_2SRC0_0	-	-	-	-	-	-	-	-
		WTIME[23]	WTIME[22]	WTIME[21]	WTIME[20]	WTIME[19]	WTIME[18]	WTIME[17]	WTIME[16]
		WTIME[15]	WTIME[14]	WTIME[13]	WTIME[11]	WTIME[11]	WTIME[10]	WTIME[9]	WTIME[8]
		WTIME[7]	WTIME[6]	WTIME[5]	WTIME[4]	WTIME[3]	WTIME[2]	WTIME[1]	WTIME[0]
	SEVMR0_2SRC0_0	EVMUF	EVMOF	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	EVMWAIT	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	SEVCR0_2SRC0_0	EVCUF	EVCOF	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	EVCWAIT	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	SRCIR1_2SRC0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT
	SADIR1_2SRC0_0	-	-	-	-	-	-	-	-
		-	-	-	OTBL[4]	OTBL[3]	OTBL[2]	OTBL[1]	OTBL[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	CHNUM[3]	CHNUM[2]	CHNUM[1]	CHNUM[0]
	SRCBR1_2SRC0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	BYPASS
	IFSCR1_2SRC0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INTIFSEN

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
SCUX	IFSVR1_2SRC0_0	-	-	-	-	INTIFS[27]	INTIFS[26]	INTIFS[25]	INTIFS[24]
		INTIFS[23]	INTIFS[22]	INTIFS[21]	INTIFS[20]	INTIFS[19]	INTIFS[18]	INTIFS[17]	INTIFS[16]
		INTIFS[15]	INTIFS[14]	INTIFS[13]	INTIFS[11]	INTIFS[11]	INTIFS[10]	INTIFS[9]	INTIFS[8]
		INTIFS[7]	INTIFS[6]	INTIFS[5]	INTIFS[4]	INTIFS[3]	INTIFS[2]	INTIFS[1]	INTIFS[0]
	SRCCR1_2SRC0_0	-	-	-	-	-	-	-	-
		-	-	-	WATMD	-	-	-	-
		-	-	-	BUFMD	-	-	-	-
		-	-	-	-	-	-	-	SRCMD
	MNFSR1_2SRC0_0	-	-	-	-	MINFS[27]	MINFS[26]	MINFS[25]	MINFS[24]
		MINFS[23]	MINFS[22]	MINFS[21]	MINFS[20]	MINFS[19]	MINFS[18]	MINFS[17]	MINFS[16]
		MINFS[15]	MINFS[14]	MINFS[13]	MINFS[11]	MINFS[11]	MINFS[10]	MINFS[9]	MINFS[8]
		MINFS[7]	MINFS[6]	MINFS[5]	MINFS[4]	MINFS[3]	MINFS[2]	MINFS[1]	MINFS[0]
	BFSSR1_2SRC0_0	-	-	-	-	-	-	BUFDATA[9]	BUFDATA[8]
		BUFDATA[7]	BUFDATA[6]	BUFDATA[5]	BUFDATA[4]	BUFDATA[3]	BUFDATA[1]	BUFDATA[1]	BUFDATA[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	BUFIN[3]	BUFIN[2]	BUFIN[1]	BUFIN[0]
	SC2SR1_2SRC0_0	SRCWSTS	SC2MUTE	SC2STS[1]	SC2STS[0]	SC2FSI[27]	SC2FSI[26]	SC2FSI[25]	SC2FSI[24]
		SC2FSI[23]	SC2FSI[22]	SC2FSI[21]	SC2FSI[20]	SC2FSI[19]	SC2FSI[18]	SC2FSI[17]	SC2FSI[16]
		SC2FSI[15]	SC2FSI[14]	SC2FSI[13]	SC2FSI[11]	SC2FSI[11]	SC2FSI[10]	SC2FSI[9]	SC2FSI[8]
		SC2FSI[7]	SC2FSI[6]	SC2FSI[5]	SC2FSI[4]	SC2FSI[3]	SC2FSI[2]	SC2FSI[1]	SC2FSI[0]
	WATSR1_2SRC0_0	-	-	-	-	-	-	-	-
		WTIME[23]	WTIME[22]	WTIME[21]	WTIME[20]	WTIME[19]	WTIME[18]	WTIME[17]	WTIME[16]
		WTIME[15]	WTIME[14]	WTIME[13]	WTIME[11]	WTIME[11]	WTIME[10]	WTIME[9]	WTIME[8]
		WTIME[7]	WTIME[6]	WTIME[5]	WTIME[4]	WTIME[3]	WTIME[2]	WTIME[1]	WTIME[0]
	SEVMR1_2SRC0_0	EVMUF	EVMOF	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	EVMWAIT	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	SEVCR1_2SRC0_0	EVCUF	EVCOF	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	EVCWAIT	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	SRCIRR_2SRC0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT
	SRCIR0_2SRC0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT
	SADIR0_2SRC0_1	-	-	-	-	-	-	-	-
		-	-	-	OTBL[4]	OTBL[3]	OTBL[2]	OTBL[1]	OTBL[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	CHNUM[3]	CHNUM[2]	CHNUM[1]	CHNUM[0]
	SRCBR0_2SRC0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	BYPASS
	IFSCR0_2SRC0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INTIFSEN
	IFSVR0_2SRC0_1	-	-	-	-	INTIFS[27]	INTIFS[26]	INTIFS[25]	INTIFS[24]
		INTIFS[23]	INTIFS[22]	INTIFS[21]	INTIFS[20]	INTIFS[19]	INTIFS[18]	INTIFS[17]	INTIFS[16]
		INTIFS[15]	INTIFS[14]	INTIFS[13]	INTIFS[11]	INTIFS[11]	INTIFS[10]	INTIFS[9]	INTIFS[8]
		INTIFS[7]	INTIFS[6]	INTIFS[5]	INTIFS[4]	INTIFS[3]	INTIFS[2]	INTIFS[1]	INTIFS[0]
	SRCCR0_2SRC0_1	-	-	-	-	-	-	-	-
		-	-	-	WATMD	-	-	-	-
		-	-	-	BUFMD	-	-	-	-
		-	-	-	-	-	-	-	SRCMD

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
SCUX	MNFSR0_2SRC0_1	-	-	-	-	MINFS[27]	MINFS[26]	MINFS[25]	MINFS[24]
		MINFS[23]	MINFS[22]	MINFS[21]	MINFS[20]	MINFS[19]	MINFS[18]	MINFS[17]	MINFS[16]
		MINFS[15]	MINFS[14]	MINFS[13]	MINFS[11]	MINFS[11]	MINFS[10]	MINFS[9]	MINFS[8]
		MINFS[7]	MINFS[6]	MINFS[5]	MINFS[4]	MINFS[3]	MINFS[2]	MINFS[1]	MINFS[0]
	BFSSR0_2SRC0_1	-	-	-	-	-	-	BUFDATA[9]	BUFDATA[8]
		BUFDATA[7]	BUFDATA[6]	BUFDATA[5]	BUFDATA[4]	BUFDATA[3]	BUFDATA[1]	BUFDATA[1]	BUFDATA[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	BUFIN[3]	BUFIN[2]	BUFIN[1]	BUFIN[0]
	SC2SR0_2SRC0_1	SRCWSTS	SC2MUTE	SC2STS[1]	SC2STS[0]	SC2FSI[27]	SC2FSI[26]	SC2FSI[25]	SC2FSI[24]
		SC2FSI[23]	SC2FSI[22]	SC2FSI[21]	SC2FSI[20]	SC2FSI[19]	SC2FSI[18]	SC2FSI[17]	SC2FSI[16]
		SC2FSI[15]	SC2FSI[14]	SC2FSI[13]	SC2FSI[11]	SC2FSI[11]	SC2FSI[10]	SC2FSI[9]	SC2FSI[8]
		SC2FSI[7]	SC2FSI[6]	SC2FSI[5]	SC2FSI[4]	SC2FSI[3]	SC2FSI[2]	SC2FSI[1]	SC2FSI[0]
	WATSR0_2SRC0_1	-	-	-	-	-	-	-	-
		WTIME[23]	WTIME[22]	WTIME[21]	WTIME[20]	WTIME[19]	WTIME[18]	WTIME[17]	WTIME[16]
		WTIME[15]	WTIME[14]	WTIME[13]	WTIME[11]	WTIME[11]	WTIME[10]	WTIME[9]	WTIME[8]
		WTIME[7]	WTIME[6]	WTIME[5]	WTIME[4]	WTIME[3]	WTIME[2]	WTIME[1]	WTIME[0]
	SEVMR0_2SRC0_1	EVMUF	EVMOF	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	EVMWAIT	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	SEVCR0_2SRC0_1	EVCUF	EVCOF	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	EVCWAIT	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	SRCIR1_2SRC0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT
	SADIR1_2SRC0_1	-	-	-	-	-	-	-	-
		-	-	-	OTBL[4]	OTBL[3]	OTBL[2]	OTBL[1]	OTBL[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	CHNUM[3]	CHNUM[2]	CHNUM[1]	CHNUM[0]
	SRCBR1_2SRC0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	BYPASS
	IFSCR1_2SRC0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INTIFSEN
	IFSVR1_2SRC0_1	-	-	-	-	INTIFS[27]	INTIFS[26]	INTIFS[25]	INTIFS[24]
		INTIFS[23]	INTIFS[22]	INTIFS[21]	INTIFS[20]	INTIFS[19]	INTIFS[18]	INTIFS[17]	INTIFS[16]
		INTIFS[15]	INTIFS[14]	INTIFS[13]	INTIFS[11]	INTIFS[11]	INTIFS[10]	INTIFS[9]	INTIFS[8]
		INTIFS[7]	INTIFS[6]	INTIFS[5]	INTIFS[4]	INTIFS[3]	INTIFS[2]	INTIFS[1]	INTIFS[0]
	SRCCR1_2SRC0_1	-	-	-	-	-	-	-	-
		-	-	-	WATMD	-	-	-	-
		-	-	-	BUFMD	-	-	-	-
		-	-	-	-	-	-	-	SRCMD
	MNFSR1_2SRC0_1	-	-	-	-	MINFS[27]	MINFS[26]	MINFS[25]	MINFS[24]
		MINFS[23]	MINFS[22]	MINFS[21]	MINFS[20]	MINFS[19]	MINFS[18]	MINFS[17]	MINFS[16]
		MINFS[15]	MINFS[14]	MINFS[13]	MINFS[11]	MINFS[11]	MINFS[10]	MINFS[9]	MINFS[8]
		MINFS[7]	MINFS[6]	MINFS[5]	MINFS[4]	MINFS[3]	MINFS[2]	MINFS[1]	MINFS[0]
	BFSSR1_2SRC0_1	-	-	-	-	-	-	BUFDATA[9]	BUFDATA[8]
		BUFDATA[7]	BUFDATA[6]	BUFDATA[5]	BUFDATA[4]	BUFDATA[3]	BUFDATA[1]	BUFDATA[1]	BUFDATA[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	BUFIN[3]	BUFIN[2]	BUFIN[1]	BUFIN[0]
	SC2SR1_2SRC0_1	SRCWSTS	SC2MUTE	SC2STS[1]	SC2STS[0]	SC2FSI[27]	SC2FSI[26]	SC2FSI[25]	SC2FSI[24]
		SC2FSI[23]	SC2FSI[22]	SC2FSI[21]	SC2FSI[20]	SC2FSI[19]	SC2FSI[18]	SC2FSI[17]	SC2FSI[16]
		SC2FSI[15]	SC2FSI[14]	SC2FSI[13]	SC2FSI[11]	SC2FSI[11]	SC2FSI[10]	SC2FSI[9]	SC2FSI[8]
		SC2FSI[7]	SC2FSI[6]	SC2FSI[5]	SC2FSI[4]	SC2FSI[3]	SC2FSI[2]	SC2FSI[1]	SC2FSI[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
SCUX	WATSR1_2SRC0_1	-	-	-	-	-	-	-	-
		WTIME[23]	WTIME[22]	WTIME[21]	WTIME[20]	WTIME[19]	WTIME[18]	WTIME[17]	WTIME[16]
		WTIME[15]	WTIME[14]	WTIME[13]	WTIME[11]	WTIME[11]	WTIME[10]	WTIME[9]	WTIME[8]
		WTIME[7]	WTIME[6]	WTIME[5]	WTIME[4]	WTIME[3]	WTIME[2]	WTIME[1]	WTIME[0]
	SEVMR1_2SRC0_1	EVMUF	EVMOF	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	EVMWAIT	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	SEVCR1_2SRC0_1	EVCUF	EVCOF	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	EVCWAIT	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	SRCIRR_2SRC0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT
	DVUIR_DVU0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT
	VADIR_DVU0_0	-	-	-	-	-	-	-	-
		-	-	-	OTBL[4]	OTBL[3]	OTBL[2]	OTBL[1]	OTBL[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	CHNUM[3]	CHNUM[2]	CHNUM[1]	CHNUM[0]
	DVUBR_DVU0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	BYPASS
	DVUCR_DVU0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	VVMD
		-	-	-	VRMD	-	-	-	ZCMD
	ZCMCR_DVU0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ZCEN7	ZCEN6	ZCEN5	ZCEN4	ZCEN3	ZCEN2	ZCEN1	ZCEN0
	VRCTR_DVU0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		VREN7	VREN6	VREN5	VREN4	VREN3	VREN2	VREN1	VREN0
	VRPDR_DVU0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	VRPDUP[4]	VRPDUP[3]	VRPDUP[1]	VRPDUP[1]	VRPDUP[0]
		-	-	-	VRPDDW[4]	VRPDDW[3]	VRPDDW[2]	VRPDDW[1]	VRPDDW[0]
	VRDBR_DVU0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	VRDB[9]	VRDB[8]
		VRDB[7]	VRDB[6]	VRDB[5]	VRDB[4]	VRDB[3]	VRDB[2]	VRDB[1]	VRDB[0]
	VRWTR_DVU0_0	-	-	-	-	-	-	-	-
		VRWT[23]	VRWT[22]	VRWT[21]	VRWT[20]	VRWT[19]	VRWT[18]	VRWT[17]	VRWT[16]
		VRWT[15]	VRWT[14]	VRWT[13]	VRWT[11]	VRWT[11]	VRWT[10]	VRWT[9]	VRWT[8]
		VRWT[7]	VRWT[6]	VRWT[5]	VRWT[4]	VRWT[3]	VRWT[2]	VRWT[1]	VRWT[0]
	VOL0R_DVU0_0	-	-	-	-	-	-	-	-
		VOLVAL0[23]	VOLVAL0[22]	VOLVAL0[21]	VOLVAL0[20]	VOLVAL0[19]	VOLVAL0[18]	VOLVAL0[17]	VOLVAL0[16]
		VOLVAL0[15]	VOLVAL0[14]	VOLVAL0[13]	VOLVAL0[11]	VOLVAL0[11]	VOLVAL0[10]	VOLVAL0[9]	VOLVAL0[8]
		VOLVAL0[7]	VOLVAL0[6]	VOLVAL0[5]	VOLVAL0[4]	VOLVAL0[3]	VOLVAL0[2]	VOLVAL0[1]	VOLVAL0[0]
	VOL1R_DVU0_0	-	-	-	-	-	-	-	-
		VOLVAL1[23]	VOLVAL1[22]	VOLVAL1[21]	VOLVAL1[20]	VOLVAL1[19]	VOLVAL1[18]	VOLVAL1[17]	VOLVAL1[16]
		VOLVAL1[15]	VOLVAL1[14]	VOLVAL1[13]	VOLVAL1[11]	VOLVAL1[11]	VOLVAL1[10]	VOLVAL1[9]	VOLVAL1[8]
		VOLVAL1[7]	VOLVAL1[6]	VOLVAL1[5]	VOLVAL1[4]	VOLVAL1[3]	VOLVAL1[2]	VOLVAL1[1]	VOLVAL1[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
SCUX	VOL2R_DVU0_0	-	-	-	-	-	-	-	-
		VOLVAL2[23]	VOLVAL2[22]	VOLVAL2[21]	VOLVAL2[20]	VOLVAL2[19]	VOLVAL2[18]	VOLVAL2[17]	VOLVAL2[16]
		VOLVAL2[15]	VOLVAL2[14]	VOLVAL2[13]	VOLVAL2[11]	VOLVAL2[11]	VOLVAL2[10]	VOLVAL2[9]	VOLVAL2[8]
		VOLVAL2[7]	VOLVAL2[6]	VOLVAL2[5]	VOLVAL2[4]	VOLVAL2[3]	VOLVAL2[2]	VOLVAL2[1]	VOLVAL2[0]
	VOL3R_DVU0_0	-	-	-	-	-	-	-	-
		VOLVAL3[23]	VOLVAL3[22]	VOLVAL3[21]	VOLVAL3[20]	VOLVAL3[19]	VOLVAL3[18]	VOLVAL3[17]	VOLVAL3[16]
		VOLVAL3[15]	VOLVAL3[14]	VOLVAL3[13]	VOLVAL3[11]	VOLVAL3[11]	VOLVAL3[10]	VOLVAL3[9]	VOLVAL3[8]
		VOLVAL3[7]	VOLVAL3[6]	VOLVAL3[5]	VOLVAL3[4]	VOLVAL3[3]	VOLVAL3[2]	VOLVAL3[1]	VOLVAL3[0]
	VOL4R_DVU0_0	-	-	-	-	-	-	-	-
		VOLVAL4[23]	VOLVAL4[22]	VOLVAL4[21]	VOLVAL4[20]	VOLVAL4[19]	VOLVAL4[18]	VOLVAL4[17]	VOLVAL4[16]
		VOLVAL4[15]	VOLVAL4[14]	VOLVAL4[13]	VOLVAL4[11]	VOLVAL4[11]	VOLVAL4[10]	VOLVAL4[9]	VOLVAL4[8]
		VOLVAL4[7]	VOLVAL4[6]	VOLVAL4[5]	VOLVAL4[4]	VOLVAL4[3]	VOLVAL4[2]	VOLVAL4[1]	VOLVAL4[0]
	VOL5R_DVU0_0	-	-	-	-	-	-	-	-
		VOLVAL5[23]	VOLVAL5[22]	VOLVAL5[21]	VOLVAL5[20]	VOLVAL5[19]	VOLVAL5[18]	VOLVAL5[17]	VOLVAL5[16]
		VOLVAL5[15]	VOLVAL5[14]	VOLVAL5[13]	VOLVAL5[11]	VOLVAL5[11]	VOLVAL5[10]	VOLVAL5[9]	VOLVAL5[8]
		VOLVAL5[7]	VOLVAL5[6]	VOLVAL5[5]	VOLVAL5[4]	VOLVAL5[3]	VOLVAL5[2]	VOLVAL5[1]	VOLVAL5[0]
	VOL6R_DVU0_0	-	-	-	-	-	-	-	-
		VOLVAL6[23]	VOLVAL6[22]	VOLVAL6[21]	VOLVAL6[20]	VOLVAL6[19]	VOLVAL6[18]	VOLVAL6[17]	VOLVAL6[16]
		VOLVAL6[15]	VOLVAL6[14]	VOLVAL6[13]	VOLVAL6[11]	VOLVAL6[11]	VOLVAL6[10]	VOLVAL6[9]	VOLVAL6[8]
		VOLVAL6[7]	VOLVAL6[6]	VOLVAL6[5]	VOLVAL6[4]	VOLVAL6[3]	VOLVAL6[2]	VOLVAL6[1]	VOLVAL6[0]
	VOL7R_DVU0_0	-	-	-	-	-	-	-	-
		VOLVAL7[23]	VOLVAL7[22]	VOLVAL7[21]	VOLVAL7[20]	VOLVAL7[19]	VOLVAL7[18]	VOLVAL7[17]	VOLVAL7[16]
		VOLVAL7[15]	VOLVAL7[14]	VOLVAL7[13]	VOLVAL7[11]	VOLVAL7[11]	VOLVAL7[10]	VOLVAL7[9]	VOLVAL7[8]
		VOLVAL7[7]	VOLVAL7[6]	VOLVAL7[5]	VOLVAL7[4]	VOLVAL7[3]	VOLVAL7[2]	VOLVAL7[1]	VOLVAL7[0]
	DVUER_DVU0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	DVUEN
	DVUSR_DVU0_0	-	-	-	-	-	-	-	-
		ZSTS7	ZSTS6	ZSTS5	ZSTS4	ZSTS3	ZSTS2	ZSTS1	ZSTS0
		-	-	-	-	-	-	-	-
		-	-	-	-	-	VRSTS[2]	VRSTS[1]	VRSTS[0]
	VEVMR_DVU0_0	VEVMZCM7	VEVMZCM6	VEVMZCM5	VEVMZCM4	VEVMZCM3	VEVMZCM2	VEVMZCM1	VEVMZCM0
		-	-	-	-	-	-	-	-
		VEVMVR	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	VEVCR_DVU0_0	VEVCZCM7	VEVCZCM6	VEVCZCM5	VEVCZCM4	VEVCZCM3	VEVCZCM2	VEVCZCM1	VEVCZCM0
		-	-	-	-	-	-	-	-
		VEVCVR	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	DVUIR_DVU0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT
	VADIR_DVU0_1	-	-	-	-	-	-	-	-
		-	-	-	OTBL[4]	OTBL[3]	OTBL[2]	OTBL[1]	OTBL[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	CHNUM[3]	CHNUM[2]	CHNUM[1]	CHNUM[0]
	DVUBR_DVU0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	BYPASS
	DVUCR_DVU0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	VVMD
		-	-	-	VRMD	-	-	-	ZCMD
	ZCMCR_DVU0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ZCEN7	ZCEN6	ZCEN5	ZCEN4	ZCEN3	ZCEN2	ZCEN1	ZCEN0

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
SCUX	VRCTR_DVU0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		VREN7	VREN6	VREN5	VREN4	VREN3	VREN2	VREN1	VREN0
	VRPDR_DVU0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	VRPDUP[4]	VRPDUP[3]	VRPDUP[1]	VRPDUP[1]	VRPDUP[0]
		-	-	-	VRPDDW[4]	VRPDDW[3]	VRPDDW[2]	VRPDDW[1]	VRPDDW[0]
	VRDBR_DVU0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	VRDB[9]	VRDB[8]
		VRDB[7]	VRDB[6]	VRDB[5]	VRDB[4]	VRDB[3]	VRDB[2]	VRDB[1]	VRDB[0]
	VRWTR_DVU0_1	-	-	-	-	-	-	-	-
		VRWT[23]	VRWT[22]	VRWT[21]	VRWT[20]	VRWT[19]	VRWT[18]	VRWT[17]	VRWT[16]
		VRWT[15]	VRWT[14]	VRWT[13]	VRWT[11]	VRWT[11]	VRWT[10]	VRWT[9]	VRWT[8]
		VRWT[7]	VRWT[6]	VRWT[5]	VRWT[4]	VRWT[3]	VRWT[2]	VRWT[1]	VRWT[0]
	VOL0R_DVU0_1	-	-	-	-	-	-	-	-
		VOLVAL0[23]	VOLVAL0[22]	VOLVAL0[21]	VOLVAL0[20]	VOLVAL0[19]	VOLVAL0[18]	VOLVAL0[17]	VOLVAL0[16]
		VOLVAL0[15]	VOLVAL0[14]	VOLVAL0[13]	VOLVAL0[11]	VOLVAL0[11]	VOLVAL0[10]	VOLVAL0[9]	VOLVAL0[8]
		VOLVAL0[7]	VOLVAL0[6]	VOLVAL0[5]	VOLVAL0[4]	VOLVAL0[3]	VOLVAL0[2]	VOLVAL0[1]	VOLVAL0[0]
	VOL1R_DVU0_1	-	-	-	-	-	-	-	-
		VOLVAL1[23]	VOLVAL1[22]	VOLVAL1[21]	VOLVAL1[20]	VOLVAL1[19]	VOLVAL1[18]	VOLVAL1[17]	VOLVAL1[16]
		VOLVAL1[15]	VOLVAL1[14]	VOLVAL1[13]	VOLVAL1[11]	VOLVAL1[11]	VOLVAL1[10]	VOLVAL1[9]	VOLVAL1[8]
		VOLVAL1[7]	VOLVAL1[6]	VOLVAL1[5]	VOLVAL1[4]	VOLVAL1[3]	VOLVAL1[2]	VOLVAL1[1]	VOLVAL1[0]
	VOL2R_DVU0_1	-	-	-	-	-	-	-	-
		VOLVAL2[23]	VOLVAL2[22]	VOLVAL2[21]	VOLVAL2[20]	VOLVAL2[19]	VOLVAL2[18]	VOLVAL2[17]	VOLVAL2[16]
		VOLVAL2[15]	VOLVAL2[14]	VOLVAL2[13]	VOLVAL2[11]	VOLVAL2[11]	VOLVAL2[10]	VOLVAL2[9]	VOLVAL2[8]
		VOLVAL2[7]	VOLVAL2[6]	VOLVAL2[5]	VOLVAL2[4]	VOLVAL2[3]	VOLVAL2[2]	VOLVAL2[1]	VOLVAL2[0]
	VOL3R_DVU0_1	-	-	-	-	-	-	-	-
		VOLVAL3[23]	VOLVAL3[22]	VOLVAL3[21]	VOLVAL3[20]	VOLVAL3[19]	VOLVAL3[18]	VOLVAL3[17]	VOLVAL3[16]
		VOLVAL3[15]	VOLVAL3[14]	VOLVAL3[13]	VOLVAL3[11]	VOLVAL3[11]	VOLVAL3[10]	VOLVAL3[9]	VOLVAL3[8]
		VOLVAL3[7]	VOLVAL3[6]	VOLVAL3[5]	VOLVAL3[4]	VOLVAL3[3]	VOLVAL3[2]	VOLVAL3[1]	VOLVAL3[0]
	VOL4R_DVU0_1	-	-	-	-	-	-	-	-
		VOLVAL4[23]	VOLVAL4[22]	VOLVAL4[21]	VOLVAL4[20]	VOLVAL4[19]	VOLVAL4[18]	VOLVAL4[17]	VOLVAL4[16]
		VOLVAL4[15]	VOLVAL4[14]	VOLVAL4[13]	VOLVAL4[11]	VOLVAL4[11]	VOLVAL4[10]	VOLVAL4[9]	VOLVAL4[8]
		VOLVAL4[7]	VOLVAL4[6]	VOLVAL4[5]	VOLVAL4[4]	VOLVAL4[3]	VOLVAL4[2]	VOLVAL4[1]	VOLVAL4[0]
	VOL5R_DVU0_1	-	-	-	-	-	-	-	-
		VOLVAL5[23]	VOLVAL5[22]	VOLVAL5[21]	VOLVAL5[20]	VOLVAL5[19]	VOLVAL5[18]	VOLVAL5[17]	VOLVAL5[16]
		VOLVAL5[15]	VOLVAL5[14]	VOLVAL5[13]	VOLVAL5[11]	VOLVAL5[11]	VOLVAL5[10]	VOLVAL5[9]	VOLVAL5[8]
		VOLVAL5[7]	VOLVAL5[6]	VOLVAL5[5]	VOLVAL5[4]	VOLVAL5[3]	VOLVAL5[2]	VOLVAL5[1]	VOLVAL5[0]
	VOL6R_DVU0_1	-	-	-	-	-	-	-	-
		VOLVAL6[23]	VOLVAL6[22]	VOLVAL6[21]	VOLVAL6[20]	VOLVAL6[19]	VOLVAL6[18]	VOLVAL6[17]	VOLVAL6[16]
		VOLVAL6[15]	VOLVAL6[14]	VOLVAL6[13]	VOLVAL6[11]	VOLVAL6[11]	VOLVAL6[10]	VOLVAL6[9]	VOLVAL6[8]
		VOLVAL6[7]	VOLVAL6[6]	VOLVAL6[5]	VOLVAL6[4]	VOLVAL6[3]	VOLVAL6[2]	VOLVAL6[1]	VOLVAL6[0]
	VOL7R_DVU0_1	-	-	-	-	-	-	-	-
		VOLVAL7[23]	VOLVAL7[22]	VOLVAL7[21]	VOLVAL7[20]	VOLVAL7[19]	VOLVAL7[18]	VOLVAL7[17]	VOLVAL7[16]
		VOLVAL7[15]	VOLVAL7[14]	VOLVAL7[13]	VOLVAL7[11]	VOLVAL7[11]	VOLVAL7[10]	VOLVAL7[9]	VOLVAL7[8]
		VOLVAL7[7]	VOLVAL7[6]	VOLVAL7[5]	VOLVAL7[4]	VOLVAL7[3]	VOLVAL7[2]	VOLVAL7[1]	VOLVAL7[0]
	DVUER_DVU0_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	DVUEN
	DVUSR_DVU0_1	-	-	-	-	-	-	-	-
		ZSTS7	ZSTS6	ZSTS5	ZSTS4	ZSTS3	ZSTS2	ZSTS1	ZSTS0
		-	-	-	-	-	-	-	-
		-	-	-	-	-	VRSTS[2]	VRSTS[1]	VRSTS[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
SCUX	VEVMR_DVU0_1	VEVMZCM7	VEVMZCM6	VEVMZCM5	VEVMZCM4	VEVMZCM3	VEVMZCM2	VEVMZCM1	VEVMZCM0
		-	-	-	-	-	-	-	-
		VEVMVR	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	VEVCR_DVU0_1	VEVCZCM7	VEVCZCM6	VEVCZCM5	VEVCZCM4	VEVCZCM3	VEVCZCM2	VEVCZCM1	VEVCZCM0
		-	-	-	-	-	-	-	-
		VEVCVR	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	DVUIR_DVU0_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT
	VADIR_DVU0_2	-	-	-	-	-	-	-	-
		-	-	-	OTBL[4]	OTBL[3]	OTBL[2]	OTBL[1]	OTBL[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	CHNUM[3]	CHNUM[2]	CHNUM[1]	CHNUM[0]
	DVUBR_DVU0_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	BYPASS
	DVUCR_DVU0_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	VVMD
		-	-	-	VRMD	-	-	-	ZCMD
	ZCMCR_DVU0_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ZCEN7	ZCEN6	ZCEN5	ZCEN4	ZCEN3	ZCEN2	ZCEN1	ZCEN0
	VRCTR_DVU0_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		VREN7	VREN6	VREN5	VREN4	VREN3	VREN2	VREN1	VREN0
	VRPDR_DVU0_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	VRPDUP[4]	VRPDUP[3]	VRPDUP[1]	VRPDUP[1]	VRPDUP[0]
		-	-	-	VRPDDW[4]	VRPDDW[3]	VRPDDW[2]	VRPDDW[1]	VRPDDW[0]
	VRDBR_DVU0_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	VRDB[9]	VRDB[8]
		VRDB[7]	VRDB[6]	VRDB[5]	VRDB[4]	VRDB[3]	VRDB[2]	VRDB[1]	VRDB[0]
	VRWTR_DVU0_2	-	-	-	-	-	-	-	-
		VRWT[23]	VRWT[22]	VRWT[21]	VRWT[20]	VRWT[19]	VRWT[18]	VRWT[17]	VRWT[16]
		VRWT[15]	VRWT[14]	VRWT[13]	VRWT[11]	VRWT[11]	VRWT[10]	VRWT[9]	VRWT[8]
		VRWT[7]	VRWT[6]	VRWT[5]	VRWT[4]	VRWT[3]	VRWT[2]	VRWT[1]	VRWT[0]
	VOL0R_DVU0_2	-	-	-	-	-	-	-	-
		VOLVAL0[23]	VOLVAL0[22]	VOLVAL0[21]	VOLVAL0[20]	VOLVAL0[19]	VOLVAL0[18]	VOLVAL0[17]	VOLVAL0[16]
		VOLVAL0[15]	VOLVAL0[14]	VOLVAL0[13]	VOLVAL0[11]	VOLVAL0[11]	VOLVAL0[10]	VOLVAL0[9]	VOLVAL0[8]
		VOLVAL0[7]	VOLVAL0[6]	VOLVAL0[5]	VOLVAL0[4]	VOLVAL0[3]	VOLVAL0[2]	VOLVAL0[1]	VOLVAL0[0]
	VOL1R_DVU0_2	-	-	-	-	-	-	-	-
		VOLVAL1[23]	VOLVAL1[22]	VOLVAL1[21]	VOLVAL1[20]	VOLVAL1[19]	VOLVAL1[18]	VOLVAL1[17]	VOLVAL1[16]
		VOLVAL1[15]	VOLVAL1[14]	VOLVAL1[13]	VOLVAL1[11]	VOLVAL1[11]	VOLVAL1[10]	VOLVAL1[9]	VOLVAL1[8]
		VOLVAL1[7]	VOLVAL1[6]	VOLVAL1[5]	VOLVAL1[4]	VOLVAL1[3]	VOLVAL1[2]	VOLVAL1[1]	VOLVAL1[0]
	VOL2R_DVU0_2	-	-	-	-	-	-	-	-
		VOLVAL2[23]	VOLVAL2[22]	VOLVAL2[21]	VOLVAL2[20]	VOLVAL2[19]	VOLVAL2[18]	VOLVAL2[17]	VOLVAL2[16]
		VOLVAL2[15]	VOLVAL2[14]	VOLVAL2[13]	VOLVAL2[11]	VOLVAL2[11]	VOLVAL2[10]	VOLVAL2[9]	VOLVAL2[8]
		VOLVAL2[7]	VOLVAL2[6]	VOLVAL2[5]	VOLVAL2[4]	VOLVAL2[3]	VOLVAL2[2]	VOLVAL2[1]	VOLVAL2[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
SCUX	VOL3R_DVU0_2	-	-	-	-	-	-	-	-
		VOLVAL3[23]	VOLVAL3[22]	VOLVAL3[21]	VOLVAL3[20]	VOLVAL3[19]	VOLVAL3[18]	VOLVAL3[17]	VOLVAL3[16]
		VOLVAL3[15]	VOLVAL3[14]	VOLVAL3[13]	VOLVAL3[11]	VOLVAL3[11]	VOLVAL3[10]	VOLVAL3[9]	VOLVAL3[8]
		VOLVAL3[7]	VOLVAL3[6]	VOLVAL3[5]	VOLVAL3[4]	VOLVAL3[3]	VOLVAL3[2]	VOLVAL3[1]	VOLVAL3[0]
	VOL4R_DVU0_2	-	-	-	-	-	-	-	-
		VOLVAL4[23]	VOLVAL4[22]	VOLVAL4[21]	VOLVAL4[20]	VOLVAL4[19]	VOLVAL4[18]	VOLVAL4[17]	VOLVAL4[16]
		VOLVAL4[15]	VOLVAL4[14]	VOLVAL4[13]	VOLVAL4[11]	VOLVAL4[11]	VOLVAL4[10]	VOLVAL4[9]	VOLVAL4[8]
		VOLVAL4[7]	VOLVAL4[6]	VOLVAL4[5]	VOLVAL4[4]	VOLVAL4[3]	VOLVAL4[2]	VOLVAL4[1]	VOLVAL4[0]
	VOL5R_DVU0_2	-	-	-	-	-	-	-	-
		VOLVAL5[23]	VOLVAL5[22]	VOLVAL5[21]	VOLVAL5[20]	VOLVAL5[19]	VOLVAL5[18]	VOLVAL5[17]	VOLVAL5[16]
		VOLVAL5[15]	VOLVAL5[14]	VOLVAL5[13]	VOLVAL5[11]	VOLVAL5[11]	VOLVAL5[10]	VOLVAL5[9]	VOLVAL5[8]
		VOLVAL5[7]	VOLVAL5[6]	VOLVAL5[5]	VOLVAL5[4]	VOLVAL5[3]	VOLVAL5[2]	VOLVAL5[1]	VOLVAL5[0]
	VOL6R_DVU0_2	-	-	-	-	-	-	-	-
		VOLVAL6[23]	VOLVAL6[22]	VOLVAL6[21]	VOLVAL6[20]	VOLVAL6[19]	VOLVAL6[18]	VOLVAL6[17]	VOLVAL6[16]
		VOLVAL6[15]	VOLVAL6[14]	VOLVAL6[13]	VOLVAL6[11]	VOLVAL6[11]	VOLVAL6[10]	VOLVAL6[9]	VOLVAL6[8]
		VOLVAL6[7]	VOLVAL6[6]	VOLVAL6[5]	VOLVAL6[4]	VOLVAL6[3]	VOLVAL6[2]	VOLVAL6[1]	VOLVAL6[0]
	VOL7R_DVU0_2	-	-	-	-	-	-	-	-
		VOLVAL7[23]	VOLVAL7[22]	VOLVAL7[21]	VOLVAL7[20]	VOLVAL7[19]	VOLVAL7[18]	VOLVAL7[17]	VOLVAL7[16]
		VOLVAL7[15]	VOLVAL7[14]	VOLVAL7[13]	VOLVAL7[11]	VOLVAL7[11]	VOLVAL7[10]	VOLVAL7[9]	VOLVAL7[8]
		VOLVAL7[7]	VOLVAL7[6]	VOLVAL7[5]	VOLVAL7[4]	VOLVAL7[3]	VOLVAL7[2]	VOLVAL7[1]	VOLVAL7[0]
	DVUER_DVU0_2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	DVUEN
	DVUSR_DVU0_2	-	-	-	-	-	-	-	-
		ZSTS7	ZSTS6	ZSTS5	ZSTS4	ZSTS3	ZSTS2	ZSTS1	ZSTS0
		-	-	-	-	-	-	-	-
		-	-	-	-	-	VRSTS[2]	VRSTS[1]	VRSTS[0]
	VEVMR_DVU0_2	VEVMZCM7	VEVMZCM6	VEVMZCM5	VEVMZCM4	VEVMZCM3	VEVMZCM2	VEVMZCM1	VEVMZCM0
		-	-	-	-	-	-	-	-
		VEVMVR	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	VEVCR_DVU0_2	VEVCZCM7	VEVCZCM6	VEVCZCM5	VEVCZCM4	VEVCZCM3	VEVCZCM2	VEVCZCM1	VEVCZCM0
		-	-	-	-	-	-	-	-
		VEVCVR	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	DVUIR_DVU0_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT
	VADIR_DVU0_3	-	-	-	-	-	-	-	-
		-	-	-	OTBL[4]	OTBL[3]	OTBL[2]	OTBL[1]	OTBL[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	CHNUM[3]	CHNUM[2]	CHNUM[1]	CHNUM[0]
	DVUBR_DVU0_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	BYPASS
	DVUCR_DVU0_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	VVMD
		-	-	-	VRMD	-	-	-	ZCMD
	ZCMCR_DVU0_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ZCEN7	ZCEN6	ZCEN5	ZCEN4	ZCEN3	ZCEN2	ZCEN1	ZCEN0

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
SCUX	VRCTR_DVU0_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		VREN7	VREN6	VREN5	VREN4	VREN3	VREN2	VREN1	VREN0
	VRPDR_DVU0_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	VRPDUP[4]	VRPDUP[3]	VRPDUP[1]	VRPDUP[1]	VRPDUP[0]
		-	-	-	VRPDDW[4]	VRPDDW[3]	VRPDDW[2]	VRPDDW[1]	VRPDDW[0]
	VRDBR_DVU0_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	VRDB[9]	VRDB[8]
		VRDB[7]	VRDB[6]	VRDB[5]	VRDB[4]	VRDB[3]	VRDB[2]	VRDB[1]	VRDB[0]
	VRWTR_DVU0_3	-	-	-	-	-	-	-	-
		VRWT[23]	VRWT[22]	VRWT[21]	VRWT[20]	VRWT[19]	VRWT[18]	VRWT[17]	VRWT[16]
		VRWT[15]	VRWT[14]	VRWT[13]	VRWT[11]	VRWT[11]	VRWT[10]	VRWT[9]	VRWT[8]
		VRWT[7]	VRWT[6]	VRWT[5]	VRWT[4]	VRWT[3]	VRWT[2]	VRWT[1]	VRWT[0]
	VOL0R_DVU0_3	-	-	-	-	-	-	-	-
		VOLVAL0[23]	VOLVAL0[22]	VOLVAL0[21]	VOLVAL0[20]	VOLVAL0[19]	VOLVAL0[18]	VOLVAL0[17]	VOLVAL0[16]
		VOLVAL0[15]	VOLVAL0[14]	VOLVAL0[13]	VOLVAL0[11]	VOLVAL0[11]	VOLVAL0[10]	VOLVAL0[9]	VOLVAL0[8]
		VOLVAL0[7]	VOLVAL0[6]	VOLVAL0[5]	VOLVAL0[4]	VOLVAL0[3]	VOLVAL0[2]	VOLVAL0[1]	VOLVAL0[0]
	VOL1R_DVU0_3	-	-	-	-	-	-	-	-
		VOLVAL1[23]	VOLVAL1[22]	VOLVAL1[21]	VOLVAL1[20]	VOLVAL1[19]	VOLVAL1[18]	VOLVAL1[17]	VOLVAL1[16]
		VOLVAL1[15]	VOLVAL1[14]	VOLVAL1[13]	VOLVAL1[11]	VOLVAL1[11]	VOLVAL1[10]	VOLVAL1[9]	VOLVAL1[8]
		VOLVAL1[7]	VOLVAL1[6]	VOLVAL1[5]	VOLVAL1[4]	VOLVAL1[3]	VOLVAL1[2]	VOLVAL1[1]	VOLVAL1[0]
	VOL2R_DVU0_3	-	-	-	-	-	-	-	-
		VOLVAL2[23]	VOLVAL2[22]	VOLVAL2[21]	VOLVAL2[20]	VOLVAL2[19]	VOLVAL2[18]	VOLVAL2[17]	VOLVAL2[16]
		VOLVAL2[15]	VOLVAL2[14]	VOLVAL2[13]	VOLVAL2[11]	VOLVAL2[11]	VOLVAL2[10]	VOLVAL2[9]	VOLVAL2[8]
		VOLVAL2[7]	VOLVAL2[6]	VOLVAL2[5]	VOLVAL2[4]	VOLVAL2[3]	VOLVAL2[2]	VOLVAL2[1]	VOLVAL2[0]
	VOL3R_DVU0_3	-	-	-	-	-	-	-	-
		VOLVAL3[23]	VOLVAL3[22]	VOLVAL3[21]	VOLVAL3[20]	VOLVAL3[19]	VOLVAL3[18]	VOLVAL3[17]	VOLVAL3[16]
		VOLVAL3[15]	VOLVAL3[14]	VOLVAL3[13]	VOLVAL3[11]	VOLVAL3[11]	VOLVAL3[10]	VOLVAL3[9]	VOLVAL3[8]
		VOLVAL3[7]	VOLVAL3[6]	VOLVAL3[5]	VOLVAL3[4]	VOLVAL3[3]	VOLVAL3[2]	VOLVAL3[1]	VOLVAL3[0]
	VOL4R_DVU0_3	-	-	-	-	-	-	-	-
		VOLVAL4[23]	VOLVAL4[22]	VOLVAL4[21]	VOLVAL4[20]	VOLVAL4[19]	VOLVAL4[18]	VOLVAL4[17]	VOLVAL4[16]
		VOLVAL4[15]	VOLVAL4[14]	VOLVAL4[13]	VOLVAL4[11]	VOLVAL4[11]	VOLVAL4[10]	VOLVAL4[9]	VOLVAL4[8]
		VOLVAL4[7]	VOLVAL4[6]	VOLVAL4[5]	VOLVAL4[4]	VOLVAL4[3]	VOLVAL4[2]	VOLVAL4[1]	VOLVAL4[0]
	VOL5R_DVU0_3	-	-	-	-	-	-	-	-
		VOLVAL5[23]	VOLVAL5[22]	VOLVAL5[21]	VOLVAL5[20]	VOLVAL5[19]	VOLVAL5[18]	VOLVAL5[17]	VOLVAL5[16]
		VOLVAL5[15]	VOLVAL5[14]	VOLVAL5[13]	VOLVAL5[11]	VOLVAL5[11]	VOLVAL5[10]	VOLVAL5[9]	VOLVAL5[8]
		VOLVAL5[7]	VOLVAL5[6]	VOLVAL5[5]	VOLVAL5[4]	VOLVAL5[3]	VOLVAL5[2]	VOLVAL5[1]	VOLVAL5[0]
	VOL6R_DVU0_3	-	-	-	-	-	-	-	-
		VOLVAL6[23]	VOLVAL6[22]	VOLVAL6[21]	VOLVAL6[20]	VOLVAL6[19]	VOLVAL6[18]	VOLVAL6[17]	VOLVAL6[16]
		VOLVAL6[15]	VOLVAL6[14]	VOLVAL6[13]	VOLVAL6[11]	VOLVAL6[11]	VOLVAL6[10]	VOLVAL6[9]	VOLVAL6[8]
		VOLVAL6[7]	VOLVAL6[6]	VOLVAL6[5]	VOLVAL6[4]	VOLVAL6[3]	VOLVAL6[2]	VOLVAL6[1]	VOLVAL6[0]
	VOL7R_DVU0_3	-	-	-	-	-	-	-	-
		VOLVAL7[23]	VOLVAL7[22]	VOLVAL7[21]	VOLVAL7[20]	VOLVAL7[19]	VOLVAL7[18]	VOLVAL7[17]	VOLVAL7[16]
		VOLVAL7[15]	VOLVAL7[14]	VOLVAL7[13]	VOLVAL7[11]	VOLVAL7[11]	VOLVAL7[10]	VOLVAL7[9]	VOLVAL7[8]
		VOLVAL7[7]	VOLVAL7[6]	VOLVAL7[5]	VOLVAL7[4]	VOLVAL7[3]	VOLVAL7[2]	VOLVAL7[1]	VOLVAL7[0]
	DVUER_DVU0_3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	DVUEN
	DVUSR_DVU0_3	-	-	-	-	-	-	-	-
		ZSTS7	ZSTS6	ZSTS5	ZSTS4	ZSTS3	ZSTS2	ZSTS1	ZSTS0
		-	-	-	-	-	-	-	-
		-	-	-	-	-	VRSTS[2]	VRSTS[1]	VRSTS[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
SCUX	VEVMR_DVU0_3	VEVMZCM7	VEVMZCM6	VEVMZCM5	VEVMZCM4	VEVMZCM3	VEVMZCM2	VEVMZCM1	VEVMZCM0
		-	-	-	-	-	-	-	-
		VEVMVR	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	VEVCR_DVU0_3	VEVCZCM7	VEVCZCM6	VEVCZCM5	VEVCZCM4	VEVCZCM3	VEVCZCM2	VEVCZCM1	VEVCZCM0
		-	-	-	-	-	-	-	-
		VEVCVR	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	
	MIXIR_MIX0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	INIT
	MADIR_MIX0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CHNUM[3]	CHNUM[2]	CHNUM[1]	CHNUM[0]
	MIXBR_MIX0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	BPSYS[1]	BPSYS[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	BYPASS
	MIXMR_MIX0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	MIXMODE
	MVPDR_MIX0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	MXPDUP[3]	MXPDUP[2]	MXPDUP[1]	MXPDUP[0]
		-	-	-	-	MXPDDW[3]	MXPDDW[2]	MXPDDW[1]	MXPDDW[0]
	MDBAR_MIX0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	MIXDBA[9]	MIXDBA[8]
		MIXDBA[7]	MIXDBA[6]	MIXDBA[5]	MIXDBA[4]	MIXDBA[3]	MIXDBA[2]	MIXDBA[1]	MIXDBA[0]
	MDBBR_MIX0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	MIXDBB[9]	MIXDBB[8]
		MIXDBB[7]	MIXDBB[6]	MIXDBB[5]	MIXDBB[4]	MIXDBB[3]	MIXDBB[2]	MIXDBB[1]	MIXDBB[0]
	MDBCR_MIX0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	MIXDBC[9]	MIXDBC[8]
		MIXDBC[7]	MIXDBC[6]	MIXDBC[5]	MIXDBC[4]	MIXDBC[3]	MIXDBC[2]	MIXDBC[1]	MIXDBC[0]
	MDBDR_MIX0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	MIXDBD[9]	MIXDBD[8]
		MIXDBD[7]	MIXDBD[6]	MIXDBD[5]	MIXDBD[4]	MIXDBD[3]	MIXDBD[2]	MIXDBD[1]	MIXDBD[0]
	MDBER_MIX0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	MIXDBEN
	MIXSR_MIX0_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	MIXSTS[1]	MIXSTS[0]
	SWRSR_CIM	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	SWRST

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
SCUX	DMACR_CIM	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DMAMDFFU3	DMAMDFFU2	DMAMDFFU1	DMAMDFFU0	DMAMDFFD3	DMAMDFFD2	DMAMDFFD1	DMAMDFFD0
	DMATD0_CIM								
	DMATD1_CIM								
	DMATD2_CIM								
	DMATD3_CIM								
	DMATU0_CIM								
	DMATU1_CIM								
	DMATU2_CIM								
	DMATU3_CIM								
	SSIRSEL_CIM	SISEL3[1]	SISEL3[0]	SISEL2[1]	SISEL2[0]	SISEL1[1]	SISEL1[0]	SISEL0[1]	SISEL0[0]
		-	-	-	-	-	-	-	-
		-	-	SOSEL3[1]	SOSEL3[0]	-	-	SOSEL2[1]	SOSEL2[0]
		-	-	SOSEL1[1]	SOSEL1[0]	-	-	SOSEL0[1]	SOSEL0[0]
	FDTSEL0_CIM	-	-	-	-	-	SCKDIV[11]	SCKDIV[10]	SCKDIV[9]
		SCKDIV[8]	SCKDIV[7]	SCKDIV[6]	SCKDIV[5]	SCKDIV[4]	SCKDIV[3]	SCKDIV[2]	SCKDIV[1]
		SCKDIV[0]	-	-	-	-	-	-	DIVEN
		-	-	-	-	SCKSEL[3]	SCKSEL[2]	SCKSEL[1]	SCKSEL[0]
	FDTSEL1_CIM	-	-	-	-	-	SCKDIV[10]	SCKDIV[9]	SCKDIV[8]
		SCKDIV[7]	SCKDIV[6]	SCKDIV[5]	SCKDIV[4]	SCKDIV[3]	SCKDIV[2]	SCKDIV[1]	SCKDIV[0]
		-	-	-	-	-	-	-	DIVEN
		-	-	-	-	SCKSEL[3]	SCKSEL[2]	SCKSEL[1]	SCKSEL[0]
	FDTSEL2_CIM	-	-	-	-	-	SCKDIV[10]	SCKDIV[9]	SCKDIV[8]
		SCKDIV[7]	SCKDIV[6]	SCKDIV[5]	SCKDIV[4]	SCKDIV[3]	SCKDIV[2]	SCKDIV[1]	SCKDIV[0]
		-	-	-	-	-	-	-	DIVEN
		-	-	-	-	SCKSEL[3]	SCKSEL[2]	SCKSEL[1]	SCKSEL[0]
	FDTSEL3_CIM	-	-	-	-	-	SCKDIV[10]	SCKDIV[9]	SCKDIV[8]
		SCKDIV[7]	SCKDIV[6]	SCKDIV[5]	SCKDIV[4]	SCKDIV[3]	SCKDIV[2]	SCKDIV[1]	SCKDIV[0]
		-	-	-	-	-	-	-	DIVEN
		-	-	-	-	SCKSEL[3]	SCKSEL[2]	SCKSEL[1]	SCKSEL[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
SCUX	FUTSEL0_CIM	-	-	-	-	-	SCKDIV[10]	SCKDIV[9]	SCKDIV[8]
		SCKDIV[7]	SCKDIV[6]	SCKDIV[5]	SCKDIV[4]	SCKDIV[3]	SCKDIV[2]	SCKDIV[1]	SCKDIV[0]
		-	-	-	-	-	-	-	DIVEN
		-	-	-	-	SCKSEL[3]	SCKSEL[2]	SCKSEL[1]	SCKSEL[0]
	FUTSEL1_CIM	-	-	-	-	-	SCKDIV[10]	SCKDIV[9]	SCKDIV[8]
		SCKDIV[7]	SCKDIV[6]	SCKDIV[5]	SCKDIV[4]	SCKDIV[3]	SCKDIV[2]	SCKDIV[1]	SCKDIV[0]
		-	-	-	-	-	-	-	DIVEN
		-	-	-	-	SCKSEL[3]	SCKSEL[2]	SCKSEL[1]	SCKSEL[0]
	FUTSEL2_CIM	-	-	-	-	-	SCKDIV[10]	SCKDIV[9]	SCKDIV[8]
		SCKDIV[7]	SCKDIV[6]	SCKDIV[5]	SCKDIV[4]	SCKDIV[3]	SCKDIV[2]	SCKDIV[1]	SCKDIV[0]
		-	-	-	-	-	-	-	DIVEN
		-	-	-	-	SCKSEL[3]	SCKSEL[2]	SCKSEL[1]	SCKSEL[0]
	FUTSEL3_CIM	-	-	-	-	-	SCKDIV[10]	SCKDIV[9]	SCKDIV[8]
		SCKDIV[7]	SCKDIV[6]	SCKDIV[5]	SCKDIV[4]	SCKDIV[3]	SCKDIV[2]	SCKDIV[1]	SCKDIV[0]
		-	-	-	-	-	-	-	DIVEN
		-	-	-	-	SCKSEL[3]	SCKSEL[2]	SCKSEL[1]	SCKSEL[0]
	SSIPMD_CIM	-	-	-	-	-	-	-	-
		-	-	-	-	SSI3CKS	SSI2CKS	SSI1CKS	SSI0CKS
		SSI3PMD[1]	SSI3PMD[0]	-	-	-	-	-	-
		-	-	-	SSI012EN	SSI2PMD[1]	SSI2PMD[0]	SSI1PMD[1]	SSI1PMD[0]
	SSICTRL_CIM	-	SSI3TX	-	-	-	SSI3RX	-	-
		-	-	-	-	-	-	-	-
		-	SSI0TX	SSI1TX	SSI2TX	-	SSI0RX	SSI1RX	SSI2RX
		-	-	-	-	-	-	SSI012TEN	SSI012REN
	SRCRSEL0_CIM	-	PLACE7[2]	PLACE7[1]	PLACE7[0]	-	PLACE6[2]	PLACE6[1]	PLACE6[0]
		-	PLACE5[2]	PLACE5[1]	PLACE5[0]	-	PLACE4[2]	PLACE4[1]	PLACE4[0]
		-	PLACE3[2]	PLACE3[1]	PLACE3[0]	-	PLACE2[2]	PLACE2[1]	PLACE2[0]
		-	PLACE1[2]	PLACE1[1]	PLACE1[0]	-	PLACE0[2]	PLACE0[1]	PLACE0[0]
	SRCRSEL1_CIM	-	PLACE7[2]	PLACE7[1]	PLACE7[0]	-	PLACE6[2]	PLACE6[1]	PLACE6[0]
		-	PLACE5[2]	PLACE5[1]	PLACE5[0]	-	PLACE4[2]	PLACE4[1]	PLACE4[0]
		-	PLACE3[2]	PLACE3[1]	PLACE3[0]	-	PLACE2[2]	PLACE2[1]	PLACE2[0]
		-	PLACE1[2]	PLACE1[1]	PLACE1[0]	-	PLACE0[2]	PLACE0[1]	PLACE0[0]
	SRCRSEL2_CIM	-	PLACE7[2]	PLACE7[1]	PLACE7[0]	-	PLACE6[2]	PLACE6[1]	PLACE6[0]
		-	PLACE5[2]	PLACE5[1]	PLACE5[0]	-	PLACE4[2]	PLACE4[1]	PLACE4[0]
		-	PLACE3[2]	PLACE3[1]	PLACE3[0]	-	PLACE2[2]	PLACE2[1]	PLACE2[0]
		-	PLACE1[2]	PLACE1[1]	PLACE1[0]	-	PLACE0[2]	PLACE0[1]	PLACE0[0]
	SRCRSEL3_CIM	-	PLACE7[2]	PLACE7[1]	PLACE7[0]	-	PLACE6[2]	PLACE6[1]	PLACE6[0]
		-	PLACE5[2]	PLACE5[1]	PLACE5[0]	-	PLACE4[2]	PLACE4[1]	PLACE4[0]
		-	PLACE3[2]	PLACE3[1]	PLACE3[0]	-	PLACE2[2]	PLACE2[1]	PLACE2[0]
		-	PLACE1[2]	PLACE1[1]	PLACE1[0]	-	PLACE0[2]	PLACE0[1]	PLACE0[0]
	MIXRSEL_CIM	-	PLACE7[2]	PLACE7[1]	PLACE7[0]	-	PLACE6[2]	PLACE6[1]	PLACE6[0]
		-	PLACE5[2]	PLACE5[1]	PLACE5[0]	-	PLACE4[2]	PLACE4[1]	PLACE4[0]
		-	PLACE3[2]	PLACE3[1]	PLACE3[0]	-	PLACE2[2]	PLACE2[1]	PLACE2[0]
		-	PLACE1[2]	PLACE1[1]	PLACE1[0]	-	PLACE0[2]	PLACE0[1]	PLACE0[0]
SD host interface	SD_CMD_0	MD[7]	MD[6]	MD[5]	MD[4]	MD[3]	MD[2]	MD[1]	MD[0]
		C[1]	C[0]	CF[45]	CF[44]	CF[43]	CF[42]	CF[41]	CF[40]
	SD_ARG0_0	CF[23]	CF[22]	CF[21]	CF[20]	CF[19]	CF[18]	CF[17]	CF[16]
		CF[15]	CF[14]	CF[13]	CF[12]	CF[11]	CF[10]	CF[9]	CF[8]
	SD_ARG1_0	CF[39]	CF[38]	CF[37]	CF[36]	CF[35]	CF[34]	CF[33]	CF[32]
		CF[31]	CF[30]	CF[29]	CF[28]	CF[27]	CF[26]	CF[25]	CF[24]
	SD_STOP_0	-	-	-	-	-	-	-	SEC
		-	-	-	-	-	-	-	STP
	SD_SECCNT_0	CNT[15]	CNT[14]	CNT[13]	CNT[12]	CNT[11]	CNT[10]	CNT[9]	CNT[8]
		CNT[7]	CNT[6]	CNT[5]	CNT[4]	CNT[3]	CNT[2]	CNT[1]	CNT[0]
	SD_RSP00_0	R[23]	R[22]	R[21]	R[20]	R[19]	R[18]	R[17]	R[16]
		R[15]	R[14]	R[13]	R[12]	R[11]	R[10]	R[9]	R[8]
	SD_RSP01_0	R[39]	R[38]	R[37]	R[36]	R[35]	R[34]	R[33]	R[32]
		R[31]	R[30]	R[29]	R[28]	R[27]	R[26]	R[25]	R[24]
	SD_RSP02_0	R[55]	R[54]	R[53]	R[52]	R[51]	R[50]	R[49]	R[48]
		R[47]	R[46]	R[45]	R[44]	R[43]	R[42]	R[41]	R[40]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
SD host interface	SD_RSP03_0	R[71]	R[70]	R[69]	R[68]	R[67]	R[66]	R[65]	R[64]
		R[63]	R[62]	R[61]	R[60]	R[59]	R[58]	R[57]	R[56]
	SD_RSP04_0	R[87]	R[86]	R[85]	R[84]	R[83]	R[82]	R[81]	R[80]
		R[79]	R[78]	R[77]	R[76]	R[75]	R[74]	R[73]	R[72]
	SD_RSP05_0	R[103]	R[102]	R[101]	R[100]	R[99]	R[98]	R[97]	R[96]
		R[95]	R[94]	R[93]	R[92]	R[91]	R[90]	R[89]	R[88]
	SD_RSP06_0	R[119]	R[118]	R[117]	R[116]	R[115]	R[114]	R[113]	R[112]
		R[111]	R[110]	R[109]	R[108]	R[107]	R[106]	R[105]	R[104]
	SD_RSP07_0	-	-	-	-	-	-	-	-
		R[127]	R[126]	R[125]	R[124]	R[123]	R[122]	R[121]	R[120]
	SD_INFO1_0	-	-	-	-	-	INFO[10]	INFO[9]	INFO[8]
		INFO[7]	-	INFO[5]	INFO[4]	INFO[3]	INFO[2]	-	INFO[0]
	SD_INFO2_0	ILA	CBSY	SCLKDIVEN	-	-	-	BWE	BRE
		DAT0	ERR[6]	ERR[5]	ERR[4]	ERR[3]	ERR[2]	ERR[1]	ERR[0]
	SD_INFO1_MASK_0	-	-	-	-	-	-	IMASK[9]	IMASK[8]
		-	-	-	IMASK[4]	IMASK[3]	IMASK[2]	-	IMASK[0]
	SD_INFO2_MASK_0	IMASK	-	-	-	-	-	BMASK[1]	BMASK[0]
		-	EMASK[6]	EMASK[5]	EMASK[4]	EMASK[3]	EMASK[2]	EMASK[1]	EMASK[0]
	SD_CLK_CTRL_0	-	-	-	-	-	-	SDCLKOFFEN	SCLKEN
		DIV[7]	DIV[6]	DIV[5]	DIV[4]	DIV[3]	DIV[2]	DIV[1]	DIV[0]
	SD_SIZE_0	-	-	-	-	-	-	LEN[9]	LEN[8]
		LEN[7]	LEN[6]	LEN[5]	LEN[4]	LEN[3]	LEN[2]	LEN[1]	LEN[0]
	SD_OPTION_0	WIDTH	-	-	-	-	-	-	-
		TOP[27]	TOP[26]	TOP[25]	TOP[24]	CTOP[24]	CTOP[23]	CTOP[22]	CTOP[21]
	SD_ERR_STS1_0	-	E[14]	E[13]	E[12]	E[11]	E[10]	E[9]	E[8]
		-	-	E[5]	E[4]	E[3]	E[2]	E[1]	E[0]
	SD_ERR_STS2_0	-	-	-	-	-	-	-	-
		-	E[6]	E[5]	E[4]	E[3]	E[2]	E[1]	E[0]
	SD_BUF0_0	BUF[31]	BUF[30]	BUF[29]	BUF[28]	BUF[27]	BUF[26]	BUF[25]	BUF[24]
		BUF[23]	BUF[22]	BUF[21]	BUF[20]	BUF[19]	BUF[18]	BUF[17]	BUF[16]
		BUF[15]	BUF[14]	BUF[13]	BUF[12]	BUF[11]	BUF[10]	BUF[9]	BUF[8]
		BUF[7]	BUF[6]	BUF[5]	BUF[4]	BUF[3]	BUF[2]	BUF[1]	BUF[0]
	SDIO_MODE_0	-	-	-	-	-	-	C52PUB	IOABT
		-	-	-	-	-	RWREQ	-	IOMOD
	SDIO_INFO1_0	EXWT	EXPUB52	-	-	-	-	-	-
		-	-	-	-	-	-	-	IOIRQ
	SDIO_INFO1_MASK_0	MEXWT	MEXPUB52	-	-	-	-	-	-
		-	-	-	-	-	-	-	IOMSK
	CC_EXT_MODE_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	DMASDRW	-
	SOFT_RST_0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	SDRST
	VERSION_0	-	-	-	-	UR[3]	UR[2]	UR[1]	UR[0]
		IP[7]	IP[6]	IP[5]	IP[4]	IP[3]	IP[2]	IP[1]	IP[0]
	EXT_SWAP_0	-	-	-	-	-	-	-	DMASEL
		SDBRSWAP	SDBWSWAP	-	-	-	-	-	-
	SD_CMD_1	MD[7]	MD[6]	MD[5]	MD[4]	MD[3]	MD[2]	MD[1]	MD[0]
		C[1]	C[0]	CF[45]	CF[44]	CF[43]	CF[42]	CF[41]	CF[40]
	SD_ARG0_1	CF[23]	CF[22]	CF[21]	CF[20]	CF[19]	CF[18]	CF[17]	CF[16]
		CF[15]	CF[14]	CF[13]	CF[12]	CF[11]	CF[10]	CF[9]	CF[8]
	SD_ARG1_1	CF[39]	CF[38]	CF[37]	CF[36]	CF[35]	CF[34]	CF[33]	CF[32]
		CF[31]	CF[30]	CF[29]	CF[28]	CF[27]	CF[26]	CF[25]	CF[24]
	SD_STOP_1	-	-	-	-	-	-	-	SEC
		-	-	-	-	-	-	-	STP
	SD_SECCNT_1	CNT[15]	CNT[14]	CNT[13]	CNT[12]	CNT[11]	CNT[10]	CNT[9]	CNT[8]
		CNT[7]	CNT[6]	CNT[5]	CNT[4]	CNT[3]	CNT[2]	CNT[1]	CNT[0]
	SD_RSP00_1	R[23]	R[22]	R[21]	R[20]	R[19]	R[18]	R[17]	R[16]
		R[15]	R[14]	R[13]	R[12]	R[11]	R[10]	R[9]	R[8]
	SD_RSP01_1	R[39]	R[38]	R[37]	R[36]	R[35]	R[34]	R[33]	R[32]
		R[31]	R[30]	R[29]	R[28]	R[27]	R[26]	R[25]	R[24]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
SD host interface	SD_RSP02_1	R[55]	R[54]	R[53]	R[52]	R[51]	R[50]	R[49]	R[48]
		R[47]	R[46]	R[45]	R[44]	R[43]	R[42]	R[41]	R[40]
	SD_RSP03_1	R[71]	R[70]	R[69]	R[68]	R[67]	R[66]	R[65]	R[64]
		R[63]	R[62]	R[61]	R[60]	R[59]	R[58]	R[57]	R[56]
	SD_RSP04_1	R[87]	R[86]	R[85]	R[84]	R[83]	R[82]	R[81]	R[80]
		R[79]	R[78]	R[77]	R[76]	R[75]	R[74]	R[73]	R[72]
	SD_RSP05_1	R[103]	R[102]	R[101]	R[100]	R[99]	R[98]	R[97]	R[96]
		R[95]	R[94]	R[93]	R[92]	R[91]	R[90]	R[89]	R[88]
	SD_RSP06_1	R[119]	R[118]	R[117]	R[116]	R[115]	R[114]	R[113]	R[112]
		R[111]	R[110]	R[109]	R[108]	R[107]	R[106]	R[105]	R[104]
	SD_RSP07_1	-	-	-	-	-	-	-	-
		R[127]	R[126]	R[125]	R[124]	R[123]	R[122]	R[121]	R[120]
	SD_INFO1_1	-	-	-	-	-	INFO[10]	INFO[9]	INFO[8]
		INFO[7]	-	INFO[5]	INFO[4]	INFO[3]	INFO[2]	-	INFO[0]
	SD_INFO2_1	ILA	CBSY	SCLKDIVEN	-	-	-	BWE	BRE
		DAT0	ERR[6]	ERR[5]	ERR[4]	ERR[3]	ERR[2]	ERR[1]	ERR[0]
	SD_INFO1_MASK_1	-	-	-	-	-	-	IMASK[9]	IMASK[8]
		-	-	-	IMASK[4]	IMASK[3]	IMASK[2]	-	IMASK[0]
	SD_INFO2_MASK_1	IMASK	-	-	-	-	-	BMASK[1]	BMASK[0]
		-	EMASK[6]	EMASK[5]	EMASK[4]	EMASK[3]	EMASK[2]	EMASK[1]	EMASK[0]
	SD_CLK_CTRL_1	-	-	-	-	-	-	SDCLKOFFEN	SCLKEN
		DIV[7]	DIV[6]	DIV[5]	DIV[4]	DIV[3]	DIV[2]	DIV[1]	DIV[0]
	SD_SIZE_1	-	-	-	-	-	-	LEN[9]	LEN[8]
		LEN[7]	LEN[6]	LEN[5]	LEN[4]	LEN[3]	LEN[2]	LEN[1]	LEN[0]
	SD_OPTION_1	WIDTH	-	-	-	-	-	-	-
		TOP[27]	TOP[26]	TOP[25]	TOP[24]	CTOP[24]	CTOP[23]	CTOP[22]	CTOP[21]
	SD_ERR_STS1_1	-	E[14]	E[13]	E[12]	E[11]	E[10]	E[9]	E[8]
		-	-	E[5]	E[4]	E[3]	E[2]	E[1]	E[0]
	SD_ERR_STS2_1	-	-	-	-	-	-	-	-
		-	E[6]	E[5]	E[4]	E[3]	E[2]	E[1]	E[0]
	SD_BUF0_1	BUF[31]	BUF[30]	BUF[29]	BUF[28]	BUF[27]	BUF[26]	BUF[25]	BUF[24]
		BUF[23]	BUF[22]	BUF[21]	BUF[20]	BUF[19]	BUF[18]	BUF[17]	BUF[16]
		BUF[15]	BUF[14]	BUF[13]	BUF[12]	BUF[11]	BUF[10]	BUF[9]	BUF[8]
		BUF[7]	BUF[6]	BUF[5]	BUF[4]	BUF[3]	BUF[2]	BUF[1]	BUF[0]
	SDIO_MODE_1	-	-	-	-	-	-	C52PUB	IOABT
		-	-	-	-	-	RWREQ	-	IOMOD
	SDIO_INFO1_1	EXWT	EXPUB52	-	-	-	-	-	-
		-	-	-	-	-	-	-	IOIRQ
	SDIO_INFO1_MASK_1	MEXWT	MEXPUB52	-	-	-	-	-	-
		-	-	-	-	-	-	-	IOMSK
	CC_EXT_MODE_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	DMASDRW	-
	SOFT_RST_1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	SDRST
	VERSION_1	-	-	-	-	UR[3]	UR[2]	UR[1]	UR[0]
		IP[7]	IP[6]	IP[5]	IP[4]	IP[3]	IP[2]	IP[1]	IP[0]
	EXT_SWAP_1	-	-	-	-	-	-	-	DMASEL
		SDBRSWAP	SDBWSWAP	-	-	-	-	-	-

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
MMC host interface	CE_CMD_SET	-	-	CMD[5]	CMD[4]	CMD[3]	CMD[2]	CMD[1]	CMD[0]
		RTYP[1]	RTYP[0]	RBSY	-	WDAT	DWEN	CMLTE	CMD12EN
		RIDXC[1]	RIDXC[0]	RCRC7C[1]	RCRC7C[0]	-	CRC16C	-	CRCSTE
		TBIT	OPDM	-	-	SBIT	-	DATW[1]	DATW[0]
	CE_ARG	ARG[31]	ARG[30]	ARG[29]	ARG[28]	ARG[27]	ARG[26]	ARG[25]	ARG[24]
		ARG[23]	ARG[22]	ARG[21]	ARG[20]	ARG[19]	ARG[18]	ARG[17]	ARG[16]
		ARG[15]	ARG[14]	ARG[13]	ARG[12]	ARG[11]	ARG[10]	ARG[9]	ARG[8]
		ARG[7]	ARG[6]	ARG[5]	ARG[4]	ARG[3]	ARG[2]	ARG[1]	ARG[0]
	CE_ARG_CMD12	C12ARG[31]	C12ARG[30]	C12ARG[29]	C12ARG[28]	C12ARG[27]	C12ARG[26]	C12ARG[25]	C12ARG[24]
		C12ARG[23]	C12ARG[22]	C12ARG[21]	C12ARG[20]	C12ARG[19]	C12ARG[18]	C12ARG[17]	C12ARG[16]
		C12ARG[15]	C12ARG[14]	C12ARG[13]	C12ARG[12]	C12ARG[11]	C12ARG[10]	C12ARG[9]	C12ARG[8]
		C12ARG[7]	C12ARG[6]	C12ARG[5]	C12ARG[4]	C12ARG[3]	C12ARG[2]	C12ARG[1]	C12ARG[0]
	CE_CMD_CTRL	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	BREAK

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
MMC host interface	CE_BLOCK_SET	BLKCNT[15]	BLKCNT[14]	BLKCNT[13]	BLKCNT[12]	BLKCNT[11]	BLKCNT[10]	BLKCNT[9]	BLKCNT[8]
		BLKCNT[7]	BLKCNT[6]	BLKCNT[5]	BLKCNT[4]	BLKCNT[3]	BLKCNT[2]	BLKCNT[1]	BLKCNT[0]
		BLKSIZ[15]	BLKSIZ[14]	BLKSIZ[13]	BLKSIZ[12]	BLKSIZ[11]	BLKSIZ[10]	BLKSIZ[9]	BLKSIZ[8]
		BLKSIZ[7]	BLKSIZ[6]	BLKSIZ[5]	BLKSIZ[4]	BLKSIZ[3]	BLKSIZ[2]	BLKSIZ[1]	BLKSIZ[0]
	CE_CLK_CTRL	-	-	-	-	-	-	-	CLKEN
		-	-	-	-	CLKDIV[3]	CLKDIV[2]	CLKDIV[1]	CLKDIV[0]
		-	-	SRSPTO[1]	SRSPTO[0]	SRBSYTO[3]	SRBSYTO[2]	SRBSYTO[1]	SRBSYTO[0]
		SRWDTO[3]	SRWDTO[2]	SRWDTO[1]	SRWDTO[0]	-	-	-	-
	CE_BUF_ACC	-	-	-	-	-	-	DMAWEN	DMAREN
		-	-	-	-	-	-	BUSW	ATYP
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	CE_RESP3	RSP[127]	RSP[126]	RSP[125]	RSP[124]	RSP[123]	RSP[122]	RSP[121]	RSP[120]
		RSP[119]	RSP[118]	RSP[117]	RSP[116]	RSP[115]	RSP[114]	RSP[113]	RSP[112]
		RSP[111]	RSP[110]	RSP[109]	RSP[108]	RSP[107]	RSP[106]	RSP[105]	RSP[104]
		RSP[103]	RSP[102]	RSP[101]	RSP[100]	RSP[99]	RSP[98]	RSP[97]	RSP[96]
	CE_RESP2	RSP[95]	RSP[94]	RSP[93]	RSP[92]	RSP[91]	RSP[90]	RSP[89]	RSP[88]
		RSP[87]	RSP[86]	RSP[85]	RSP[84]	RSP[83]	RSP[82]	RSP[81]	RSP[80]
		RSP[79]	RSP[78]	RSP[77]	RSP[76]	RSP[75]	RSP[74]	RSP[73]	RSP[72]
		RSP[71]	RSP[70]	RSP[69]	RSP[68]	RSP[67]	RSP[66]	RSP[65]	RSP[64]
	CE_RESP1	RSP[63]	RSP[62]	RSP[61]	RSP[60]	RSP[59]	RSP[58]	RSP[57]	RSP[56]
		RSP[55]	RSP[54]	RSP[53]	RSP[52]	RSP[51]	RSP[50]	RSP[49]	RSP[48]
		RSP[47]	RSP[46]	RSP[45]	RSP[44]	RSP[43]	RSP[42]	RSP[41]	RSP[40]
		RSP[39]	RSP[38]	RSP[37]	RSP[36]	RSP[35]	RSP[34]	RSP[33]	RSP[32]
	CE_RESP0	RSP[31]	RSP[30]	RSP[29]	RSP[28]	RSP[27]	RSP[26]	RSP[25]	RSP[24]
		RSP[23]	RSP[22]	RSP[21]	RSP[20]	RSP[19]	RSP[18]	RSP[17]	RSP[16]
		RSP[15]	RSP[14]	RSP[13]	RSP[12]	RSP[11]	RSP[10]	RSP[9]	RSP[8]
		RSP[7]	RSP[6]	RSP[5]	RSP[4]	RSP[3]	RSP[2]	RSP[1]	RSP[0]
	CE_RESP_CMD12	RSP12[31]	RSP12[30]	RSP12[29]	RSP12[28]	RSP12[27]	RSP12[26]	RSP12[25]	RSP12[24]
		RSP12[23]	RSP12[22]	RSP12[21]	RSP12[20]	RSP12[19]	RSP12[18]	RSP12[17]	RSP12[16]
		RSP12[15]	RSP12[14]	RSP12[13]	RSP12[12]	RSP12[11]	RSP12[10]	RSP12[9]	RSP12[8]
		RSP12[7]	RSP12[6]	RSP12[5]	RSP12[4]	RSP12[3]	RSP12[2]	RSP12[1]	RSP12[0]
	CE_DATA	DATA[31]	DATA[30]	DATA[29]	DATA[28]	DATA[27]	DATA[26]	DATA[25]	DATA[24]
		DATA[23]	DATA[22]	DATA[21]	DATA[20]	DATA[19]	DATA[18]	DATA[17]	DATA[16]
		DATA[15]	DATA[14]	DATA[13]	DATA[12]	DATA[11]	DATA[10]	DATA[9]	DATA[8]
		DATA[7]	DATA[6]	DATA[5]	DATA[4]	DATA[3]	DATA[2]	DATA[1]	DATA[0]
	CE_INT	-	-	-	-	-	CMD12DRE	CMD12RBE	CMD12CRE
		DTRANE	BUFRE	BUFWEN	BUFREN	-	-	RBSYE	CRSPE
		CMDVIO	BUFVIO	-	-	WDATERR	RDATERR	RIDXERR	RSPERR
		-	-	-	CRCSTO	WDATTO	RDATTO	RBSYTO	RSPTO
	CE_INT_EN	-	-	-	-	-	MCMD12DRE	MCMD12RBE	MCMD12CRE
		MDTRANE	MBUFRE	MBUFWEN	MBUFREN	-	-	MRBSYE	MCRSPE
		MCMDVIO	MBUFVIO	-	-	MWDATERR	MRDATERR	MRIDXERR	MRSPERR
		-	-	-	MCRCSTO	MWDATTO	MRDATTO	MRBSYTO	MRSPTO
	CE_HOST_STS1	CMDSEQ	CMDSIG	RSPIDX[5]	RSPIDX[4]	RSPIDX[3]	RSPIDX[2]	RSPIDX[1]	RSPIDX[0]
		DATSIG[7]	DATSIG[6]	DATSIG[5]	DATSIG[4]	DATSIG[3]	DATSIG[2]	DATSIG[1]	DATSIG[0]
		RCVBLK[15]	RCVBLK[14]	RCVBLK[13]	RCVBLK[12]	RCVBLK[11]	RCVBLK[10]	RCVBLK[9]	RCVBLK[8]
		RCVBLK[7]	RCVBLK[6]	RCVBLK[5]	RCVBLK[4]	RCVBLK[3]	RCVBLK[2]	RCVBLK[1]	RCVBLK[0]
	CE_HOST_STS2	CRCSTE	CRC16E	AC12CRCE	RSPCRC7E	CRCSTEBE	RDATEBE	AC12REBE	RSPEBE
		AC12IDXE	RSPIDXE	-	-	-	CRCST[2]	CRCST[1]	CRCST[0]
		-	STRDATTO	DATBSYTO	CRCSTTO	AC12BSYTO	RSPBSYTO	AC12RSPTO	STRSPTO
		-	-	-	-	-	-	-	-
	CE_DMA_MODE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	DMASEL
	CE_DETECT	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	CDSIG	CDRISE	CDFALL	-	-	-	-
		-	-	MCDRISE	MCDFALL	-	-	-	-

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
MMC host interface	CE_ADD_MODE	-	-	-	-	-	-	-	-
		-	-	-	-	CLKMAIN	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	CE_VERSION	SWRST	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		VERSION[15]	VERSION[14]	VERSION[13]	VERSION[12]	VERSION[11]	VERSION[10]	VERSION[9]	VERSION[8]
		VERSION[7]	VERSION[6]	VERSION[5]	VERSION[4]	VERSION[3]	VERSION[2]	VERSION[1]	VERSION[0]
Ports	P1	-	-	-	-	-	-	-	-
		P1[7]	P1[6]	P1[5]	P1[4]	P1[3]	P1[2]	P1[1]	P1[0]
	P2	-	-	-	-	-	-	P2[9]	P2[8]
		P2[7]	P2[6]	P2[5]	P2[4]	P2[3]	P2[2]	P2[1]	P2[0]
	P3	P3[15]	P3[14]	P3[13]	P3[12]	P3[11]	P3[10]	P3[9]	P3[8]
		P3[7]	P3[6]	P3[5]	P3[4]	P3[3]	P3[2]	P3[1]	P3[0]
	P4	-	-	-	-	-	-	-	-
		P4[7]	P4[6]	P4[5]	P4[4]	P4[3]	P4[2]	P4[1]	P4[0]
	P5	P5[15]	P5[14]	P5[13]	P5[12]	P5[11]	P5[10]	P5[9]	P5[8]
		P5[7]	P5[6]	P5[5]	P5[4]	P5[3]	P5[2]	P5[1]	P5[0]
	P6	P6[15]	P6[14]	P6[13]	P6[12]	P6[11]	P6[10]	P6[9]	P6[8]
		P6[7]	P6[6]	P6[5]	P6[4]	P6[3]	P6[2]	P6[1]	P6[0]
	P7	-	-	-	-	P7[11]	P7[10]	P7[9]	P7[8]
		P7[7]	P7[6]	P7[5]	P7[4]	P7[3]	P7[2]	P7[1]	P7[0]
	P8	P8[15]	P8[14]	P8[13]	P8[12]	P8[11]	P8[10]	P8[9]	P8[8]
		P8[7]	P8[6]	P8[5]	P8[4]	P8[3]	P8[2]	P8[1]	P8[0]
	P9	-	-	-	-	-	-	-	-
		-	-	P9[5]	P9[4]	P9[3]	P9[2]	P9[1]	P9[0]
	PSR1	-	-	-	-	-	-	-	-
		PSR1[23]	PSR1[22]	PSR1[21]	PSR1[20]	PSR1[19]	PSR1[18]	PSR1[17]	PSR1[16]
		-	-	-	-	-	-	-	-
		PSR1[7]	PSR1[6]	PSR1[5]	PSR1[4]	PSR1[3]	PSR1[2]	PSR1[1]	PSR1[0]
	PSR2	-	-	-	-	-	-	PSR2[25]	PSR2[24]
		PSR2[23]	PSR2[22]	PSR2[21]	PSR2[20]	PSR2[19]	PSR2[18]	PSR2[17]	PSR2[16]
		-	-	-	-	-	-	PSR2[9]	PSR2[8]
		PSR2[7]	PSR2[6]	PSR2[5]	PSR2[4]	PSR2[3]	PSR2[2]	PSR2[1]	PSR2[0]
	PSR3	PSR3[31]	PSR3[30]	PSR3[29]	PSR3[28]	PSR3[27]	PSR3[26]	PSR3[25]	PSR3[24]
		PSR3[23]	PSR3[22]	PSR3[21]	PSR3[20]	PSR3[19]	PSR3[18]	PSR3[17]	PSR3[16]
		PSR3[15]	PSR3[14]	PSR3[13]	PSR3[12]	PSR3[11]	PSR3[10]	PSR3[9]	PSR3[8]
		PSR3[7]	PSR3[6]	PSR3[5]	PSR3[4]	PSR3[3]	PSR3[2]	PSR3[1]	PSR3[0]
	PSR4	-	-	-	-	-	-	-	-
		PSR4[23]	PSR4[22]	PSR4[21]	PSR4[20]	PSR4[19]	PSR4[18]	PSR4[17]	PSR4[16]
		-	-	-	-	-	-	-	-
		PSR4[7]	PSR4[6]	PSR4[5]	PSR4[4]	PSR4[3]	PSR4[2]	PSR4[1]	PSR4[0]
	PSR5	PSR5[31]	PSR5[30]	PSR5[29]	PSR5[28]	PSR5[27]	PSR5[26]	PSR5[25]	PSR5[24]
		PSR5[23]	PSR5[22]	PSR5[21]	PSR5[20]	PSR5[19]	PSR5[18]	PSR5[17]	PSR5[16]
		PSR5[15]	PSR5[14]	PSR5[13]	PSR5[12]	PSR5[11]	PSR5[10]	PSR5[9]	PSR5[8]
		PSR5[7]	PSR5[6]	PSR5[5]	PSR5[4]	PSR5[3]	PSR5[2]	PSR5[1]	PSR5[0]
	PSR6	PSR6[31]	PSR6[30]	PSR6[29]	PSR6[28]	PSR6[27]	PSR6[26]	PSR6[25]	PSR6[24]
		PSR6[23]	PSR6[22]	PSR6[21]	PSR6[20]	PSR6[19]	PSR6[18]	PSR6[17]	PSR6[16]
		PSR6[15]	PSR6[14]	PSR6[13]	PSR6[12]	PSR6[11]	PSR6[10]	PSR6[9]	PSR6[8]
		PSR6[7]	PSR6[6]	PSR6[5]	PSR6[4]	PSR6[3]	PSR6[2]	PSR6[1]	PSR6[0]
	PSR7	-	-	-	-	PSR7[27]	PSR7[26]	PSR7[25]	PSR7[24]
		PSR7[23]	PSR7[22]	PSR7[21]	PSR7[20]	PSR7[19]	PSR7[18]	PSR7[17]	PSR7[16]
		-	-	-	-	PSR7[11]	PSR7[10]	PSR7[9]	PSR7[8]
		PSR7[7]	PSR7[6]	PSR7[5]	PSR7[4]	PSR7[3]	PSR7[2]	PSR7[1]	PSR7[0]
	PSR8	PSR8[31]	PSR8[30]	PSR8[29]	PSR8[28]	PSR8[27]	PSR8[26]	PSR8[25]	PSR8[24]
		PSR8[23]	PSR8[22]	PSR8[21]	PSR8[20]	PSR8[19]	PSR8[18]	PSR8[17]	PSR8[16]
		PSR8[15]	PSR8[14]	PSR8[13]	PSR8[12]	PSR8[11]	PSR8[10]	PSR8[9]	PSR8[8]
		PSR8[7]	PSR8[6]	PSR8[5]	PSR8[4]	PSR8[3]	PSR8[2]	PSR8[1]	PSR8[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Ports	PSR9	-	-	-	-	-	-	-	-
		-	-	PSR9[21]	PSR9[20]	PSR9[19]	PSR9[18]	PSR9[17]	PSR9[16]
		-	-	-	-	-	-	-	-
		-	-	PSR9[5]	PSR9[4]	PSR9[3]	PSR9[2]	PSR9[1]	PSR9[0]
	PPR0	-	-	-	-	-	-	-	-
		-	-	-	-	PPR0[3]	PPR0[2]	PPR0[1]	PPR0[0]
	PPR1	PPR1[15]	PPR1[14]	PPR1[13]	PPR1[12]	PPR1[11]	PPR1[10]	PPR1[9]	PPR1[8]
		PPR1[7]	PPR1[6]	PPR1[5]	PPR1[4]	PPR1[3]	PPR1[2]	PPR1[1]	PPR1[0]
	PPR2	-	-	-	-	-	-	PPR2[9]	PPR2[8]
		PPR2[7]	PPR2[6]	PPR2[5]	PPR2[4]	PPR2[3]	PPR2[2]	PPR2[1]	PPR2[0]
	PPR3	PPR3[15]	PPR3[14]	PPR3[13]	PPR3[12]	PPR3[11]	PPR3[10]	PPR3[9]	PPR3[8]
		PPR3[7]	PPR3[6]	PPR3[5]	PPR3[4]	PPR3[3]	PPR3[2]	PPR3[1]	PPR3[0]
	PPR4	-	-	-	-	-	-	-	-
		PPR4[7]	PPR4[6]	PPR4[5]	PPR4[4]	PPR4[3]	PPR4[2]	PPR4[1]	PPR4[0]
	PPR5	PPR5[15]	PPR5[14]	PPR5[13]	PPR5[12]	PPR5[11]	PPR5[10]	PPR5[9]	PPR5[8]
		PPR5[7]	PPR5[6]	PPR5[5]	PPR5[4]	PPR5[3]	PPR5[2]	PPR5[1]	PPR5[0]
	PPR6	PPR6[15]	PPR6[14]	PPR6[13]	PPR6[12]	PPR6[11]	PPR6[10]	PPR6[9]	PPR6[8]
		PPR6[7]	PPR6[6]	PPR6[5]	PPR6[4]	PPR6[3]	PPR6[2]	PPR6[1]	PPR6[0]
	PPR7	-	-	-	-	PPR7[11]	PPR7[10]	PPR7[9]	PPR7[8]
		PPR7[7]	PPR7[6]	PPR7[5]	PPR7[4]	PPR7[3]	PPR7[2]	PPR7[1]	PPR7[0]
	PPR8	PPR8[15]	PPR8[14]	PPR8[13]	PPR8[12]	PPR8[11]	PPR8[10]	PPR8[9]	PPR8[8]
		PPR8[7]	PPR8[6]	PPR8[5]	PPR8[4]	PPR8[3]	PPR8[2]	PPR8[1]	PPR8[0]
	PPR9	-	-	-	-	-	-	-	-
		-	-	PPR9[5]	PPR9[4]	PPR9[3]	PPR9[2]	PPR9[1]	PPR9[0]
	PM1	-	-	-	-	-	-	-	-
		PM1[7]	PM1[6]	PM1[5]	PM1[4]	PM1[3]	PM1[2]	PM1[1]	PM1[0]
	PM2	-	-	-	-	-	-	PM2[9]	PM2[8]
		PM2[7]	PM2[6]	PM2[5]	PM2[4]	PM2[3]	PM2[2]	PM2[1]	PM2[0]
	PM3	PM3[15]	PM3[14]	PM3[13]	PM3[12]	PM3[11]	PM3[10]	PM3[9]	PM3[8]
		PM3[7]	PM3[6]	PM3[5]	PM3[4]	PM3[3]	PM3[2]	PM3[1]	PM3[0]
	PM4	-	-	-	-	-	-	-	-
		PM4[7]	PM4[6]	PM4[5]	PM4[4]	PM4[3]	PM4[2]	PM4[1]	PM4[0]
	PM5	PM5[15]	PM5[14]	PM5[13]	PM5[12]	PM5[11]	PM5[10]	PM5[9]	PM5[8]
		PM5[7]	PM5[6]	PM5[5]	PM5[4]	PM5[3]	PM5[2]	PM5[1]	PM5[0]
	PM6	PM6[15]	PM6[14]	PM6[13]	PM6[12]	PM6[11]	PM6[10]	PM6[9]	PM6[8]
		PM6[7]	PM6[6]	PM6[5]	PM6[4]	PM6[3]	PM6[2]	PM6[1]	PM6[0]
	PM7	-	-	-	-	PM7[11]	PM7[10]	PM7[9]	PM7[8]
		PM7[7]	PM7[6]	PM7[5]	PM7[4]	PM7[3]	PM7[2]	PM7[1]	PM7[0]
	PM8	PM8[15]	PM8[14]	PM8[13]	PM8[12]	PM8[11]	PM8[10]	PM8[9]	PM8[8]
		PM8[7]	PM8[6]	PM8[5]	PM8[4]	PM8[3]	PM8[2]	PM8[1]	PM8[0]
	PM9	-	-	-	-	-	-	-	-
		-	-	PM9[5]	PM9[4]	PM9[3]	PM9[2]	PM9[1]	PM9[0]
	PMC0	-	-	-	-	-	-	-	-
		-	-	-	-	PMC0[3]	PMC0[2]	PMC0[1]	PMC0[0]
	PMC1	PMC1[15]	PMC1[14]	PMC1[13]	PMC1[12]	PMC1[11]	PMC1[10]	PMC1[9]	PMC1[8]
		PMC1[7]	PMC1[6]	PMC1[5]	PMC1[4]	PMC1[3]	PMC1[2]	PMC1[1]	PMC1[0]
	PMC2	-	-	-	-	-	-	PMC2[9]	PMC2[8]
		PMC2[7]	PMC2[6]	PMC2[5]	PMC2[4]	PMC2[3]	PMC2[2]	PMC2[1]	PMC2[0]
	PMC3	PMC3[15]	PMC3[14]	PMC3[13]	PMC3[12]	PMC3[11]	PMC3[10]	PMC3[9]	PMC3[8]
		PMC3[7]	PMC3[6]	PMC3[5]	PMC3[4]	PMC3[3]	PMC3[2]	PMC3[1]	PMC3[0]
	PMC4	-	-	-	-	-	-	-	-
		PMC4[7]	PMC4[6]	PMC4[5]	PMC4[4]	PMC4[3]	PMC4[2]	PMC4[1]	PMC4[0]
	PMC5	PMC5[15]	PMC5[14]	PMC5[13]	PMC5[12]	PMC5[11]	PMC5[10]	PMC5[9]	PMC5[8]
		PMC5[7]	PMC5[6]	PMC5[5]	PMC5[4]	PMC5[3]	PMC5[2]	PMC5[1]	PMC5[0]
	PMC6	PMC6[15]	PMC6[14]	PMC6[13]	PMC6[12]	PMC6[11]	PMC6[10]	PMC6[9]	PMC6[8]
		PMC6[7]	PMC6[6]	PMC6[5]	PMC6[4]	PMC6[3]	PMC6[2]	PMC6[1]	PMC6[0]
	PMC7	-	-	-	-	PMC7[11]	PMC7[10]	PMC7[9]	PMC7[8]
		PMC7[7]	PMC7[6]	PMC7[5]	PMC7[4]	PMC7[3]	PMC7[2]	PMC7[1]	PMC7[0]
	PMC8	PMC8[15]	PMC8[14]	PMC8[13]	PMC8[12]	PMC8[11]	PMC8[10]	PMC8[9]	PMC8[8]
		PMC8[7]	PMC8[6]	PMC8[5]	PMC8[4]	PMC8[3]	PMC8[2]	PMC8[1]	PMC8[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Ports	PMC9	-	-	-	-	-	-	-	-
		-	-	PMC9[5]	PMC9[4]	PMC9[3]	PMC9[2]	PMC9[1]	PMC9[0]
	PFC0	-	-	-	-	-	-	-	-
		-	-	-	-	PFC0[3]	PFC0[2]	PFC0[1]	PFC0[0]
	PFC1	PFC1[15]	PFC1[14]	PFC1[13]	PFC1[12]	PFC1[11]	PFC1[10]	PFC1[9]	PFC1[8]
		PFC1[7]	PFC1[6]	PFC1[5]	PFC1[4]	PFC1[3]	PFC1[2]	PFC1[1]	PFC1[0]
	PFC2	-	-	-	-	-	-	PFC2[9]	PFC2[8]
		PFC2[7]	PFC2[6]	PFC2[5]	PFC2[4]	PFC2[3]	PFC2[2]	PFC2[1]	PFC2[0]
	PFC3	PFC3[15]	PFC3[14]	PFC3[13]	PFC3[12]	PFC3[11]	PFC3[10]	PFC3[9]	PFC3[8]
		PFC3[7]	PFC3[6]	PFC3[5]	PFC3[4]	PFC3[3]	PFC3[2]	PFC3[1]	PFC3[0]
	PFC4	-	-	-	-	-	-	-	-
		PFC4[7]	PFC4[6]	PFC4[5]	PFC4[4]	PFC4[3]	PFC4[2]	PFC4[1]	PFC4[0]
	PFC5	PFC5[15]	PFC5[14]	PFC5[13]	PFC5[12]	PFC5[11]	PFC5[10]	PFC5[9]	PFC5[8]
		PFC5[7]	PFC5[6]	PFC5[5]	PFC5[4]	PFC5[3]	PFC5[2]	PFC5[1]	PFC5[0]
	PFC6	PFC6[15]	PFC6[14]	PFC6[13]	PFC6[12]	PFC6[11]	PFC6[10]	PFC6[9]	PFC6[8]
		PFC6[7]	PFC6[6]	PFC6[5]	PFC6[4]	PFC6[3]	PFC6[2]	PFC6[1]	PFC6[0]
	PFC7	-	-	-	-	PFC7[11]	PFC7[10]	PFC7[9]	PFC7[8]
		PFC7[7]	PFC7[6]	PFC7[5]	PFC7[4]	PFC7[3]	PFC7[2]	PFC7[1]	PFC7[0]
	PFC8	PFC8[15]	PFC8[14]	PFC8[13]	PFC8[12]	PFC8[11]	PFC8[10]	PFC8[9]	PFC8[8]
		PFC8[7]	PFC8[6]	PFC8[5]	PFC8[4]	PFC8[3]	PFC8[2]	PFC8[1]	PFC8[0]
	PFC9	-	-	-	-	-	-	-	-
		-	-	PFC9[5]	PFC9[4]	PFC9[3]	PFC9[2]	PFC9[1]	PFC9[0]
	PFCE0	-	-	-	-	-	-	-	-
		-	-	-	-	PFCE0[3]	PFCE0[2]	PFCE0[1]	PFCE0[0]
	PFCE1	PFCE1[15]	PFCE1[14]	PFCE1[13]	PFCE1[12]	PFCE1[11]	PFCE1[10]	PFCE1[9]	PFCE1[8]
		PFCE1[7]	PFCE1[6]	PFCE1[5]	PFCE1[4]	PFCE1[3]	PFCE1[2]	PFCE1[1]	PFCE1[0]
	PFCE2	-	-	-	-	-	-	PFCE2[9]	PFCE2[8]
		PFCE2[7]	PFCE2[6]	PFCE2[5]	PFCE2[4]	PFCE2[3]	PFCE2[2]	PFCE2[1]	PFCE2[0]
	PFCE3	PFCE3[15]	PFCE3[14]	PFCE3[13]	PFCE3[12]	PFCE3[11]	PFCE3[10]	PFCE3[9]	PFCE3[8]
		PFCE3[7]	PFCE3[6]	PFCE3[5]	PFCE3[4]	PFCE3[3]	PFCE3[2]	PFCE3[1]	PFCE3[0]
	PFCE4	-	-	-	-	-	-	-	-
		PFCE4[7]	PFCE4[6]	PFCE4[5]	PFCE4[4]	PFCE4[3]	PFCE4[2]	PFCE4[1]	PFCE4[0]
	PFCE5	PFCE5[15]	PFCE5[14]	PFCE5[13]	PFCE5[12]	PFCE5[11]	PFCE5[10]	PFCE5[9]	PFCE5[8]
		PFCE5[7]	PFCE5[6]	PFCE5[5]	PFCE5[4]	PFCE5[3]	PFCE5[2]	PFCE5[1]	PFCE5[0]
	PFCE6	PFCE6[15]	PFCE6[14]	PFCE6[13]	PFCE6[12]	PFCE6[11]	PFCE6[10]	PFCE6[9]	PFCE6[8]
		PFCE6[7]	PFCE6[6]	PFCE6[5]	PFCE6[4]	PFCE6[3]	PFCE6[2]	PFCE6[1]	PFCE6[0]
	PFCE7	-	-	-	-	PFCE7[11]	PFCE7[10]	PFCE7[9]	PFCE7[8]
		PFCE7[7]	PFCE7[6]	PFCE7[5]	PFCE7[4]	PFCE7[3]	PFCE7[2]	PFCE7[1]	PFCE7[0]
	PFCE8	PFCE8[15]	PFCE8[14]	PFCE8[13]	PFCE8[12]	PFCE8[11]	PFCE8[10]	PFCE8[9]	PFCE8[8]
		PFCE8[7]	PFCE8[6]	PFCE8[5]	PFCE8[4]	PFCE8[3]	PFCE8[2]	PFCE8[1]	PFCE8[0]
	PFCE9	-	-	-	-	-	-	-	-
		-	-	PFCE9[5]	PFCE9[4]	PFCE9[3]	PFCE9[2]	PFCE9[1]	PFCE9[0]
	PNOT1	-	-	-	-	-	-	-	-
		PNOT1[7]	PNOT1[6]	PNOT1[5]	PNOT1[4]	PNOT1[3]	PNOT1[2]	PNOT1[1]	PNOT1[0]
	PNOT2	-	-	-	-	-	-	PNOT2[9]	PNOT2[8]
		PNOT2[7]	PNOT2[6]	PNOT2[5]	PNOT2[4]	PNOT2[3]	PNOT2[2]	PNOT2[1]	PNOT2[0]
	PNOT3	PNOT3[15]	PNOT3[14]	PNOT3[13]	PNOT3[12]	PNOT3[11]	PNOT3[10]	PNOT3[9]	PNOT3[8]
		PNOT3[7]	PNOT3[6]	PNOT3[5]	PNOT3[4]	PNOT3[3]	PNOT3[2]	PNOT3[1]	PNOT3[0]
	PNOT4	-	-	-	-	-	-	-	-
		PNOT4[7]	PNOT4[6]	PNOT4[5]	PNOT4[4]	PNOT4[3]	PNOT4[2]	PNOT4[1]	PNOT4[0]
	PNOT5	PNOT5[15]	PNOT5[14]	PNOT5[13]	PNOT5[12]	PNOT5[11]	PNOT5[10]	PNOT5[9]	PNOT5[8]
		PNOT5[7]	PNOT5[6]	PNOT5[5]	PNOT5[4]	PNOT5[3]	PNOT5[2]	PNOT5[1]	PNOT5[0]
	PNOT6	PNOT6[15]	PNOT6[14]	PNOT6[13]	PNOT6[12]	PNOT6[11]	PNOT6[10]	PNOT6[9]	PNOT6[8]
		PNOT6[7]	PNOT6[6]	PNOT6[5]	PNOT6[4]	PNOT6[3]	PNOT6[2]	PNOT6[1]	PNOT6[0]
	PNOT7	-	-	-	-	PNOT7[11]	PNOT7[10]	PNOT7[9]	PNOT7[8]
		PNOT7[7]	PNOT7[6]	PNOT7[5]	PNOT7[4]	PNOT7[3]	PNOT7[2]	PNOT7[1]	PNOT7[0]
	PNOT8	PNOT8[15]	PNOT8[14]	PNOT8[13]	PNOT8[12]	PNOT8[11]	PNOT8[10]	PNOT8[9]	PNOT8[8]
		PNOT8[7]	PNOT8[6]	PNOT8[5]	PNOT8[4]	PNOT8[3]	PNOT8[2]	PNOT8[1]	PNOT8[0]
	PNOT9	-	-	-	-	-	-	-	-
		-	-	PNOT9[5]	PNOT9[4]	PNOT9[3]	PNOT9[2]	PNOT9[1]	PNOT9[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Ports	PMSR1	-	-	-	-	-	-	-	-
		PMSR1[23]	PMSR1[22]	PMSR1[21]	PMSR1[20]	PMSR1[19]	PMSR1[18]	PMSR1[17]	PMSR1[16]
		-	-	-	-	-	-	-	-
		PMSR1[7]	PMSR1[6]	PMSR1[5]	PMSR1[4]	PMSR1[3]	PMSR1[2]	PMSR1[1]	PMSR1[0]
	PMSR2	-	-	-	-	-	-	PMSR2[25]	PMSR2[24]
		PMSR2[23]	PMSR2[22]	PMSR2[21]	PMSR2[20]	PMSR2[19]	PMSR2[18]	PMSR2[17]	PMSR2[16]
		-	-	-	-	-	-	PMSR2[9]	PMSR2[8]
		PMSR2[7]	PMSR2[6]	PMSR2[5]	PMSR2[4]	PMSR2[3]	PMSR2[2]	PMSR2[1]	PMSR2[0]
	PMSR3	PMSR3[31]	PMSR3[30]	PMSR3[29]	PMSR3[28]	PMSR3[27]	PMSR3[26]	PMSR3[25]	PMSR3[24]
		PMSR3[23]	PMSR3[22]	PMSR3[21]	PMSR3[20]	PMSR3[19]	PMSR3[18]	PMSR3[17]	PMSR3[16]
		PMSR3[15]	PMSR3[14]	PMSR3[13]	PMSR3[12]	PMSR3[11]	PMSR3[10]	PMSR3[9]	PMSR3[8]
		PMSR3[7]	PMSR3[6]	PMSR3[5]	PMSR3[4]	PMSR3[3]	PMSR3[2]	PMSR3[1]	PMSR3[0]
	PMSR4	-	-	-	-	-	-	-	-
		PMSR4[23]	PMSR4[22]	PMSR4[21]	PMSR4[20]	PMSR4[19]	PMSR4[18]	PMSR4[17]	PMSR4[16]
		-	-	-	-	-	-	-	-
		PMSR4[7]	PMSR4[6]	PMSR4[5]	PMSR4[4]	PMSR4[3]	PMSR4[2]	PMSR4[1]	PMSR4[0]
	PMSR5	PMSR5[31]	PMSR5[30]	PMSR5[29]	PMSR5[28]	PMSR5[27]	PMSR5[26]	PMSR5[25]	PMSR5[24]
		PMSR5[23]	PMSR5[22]	PMSR5[21]	PMSR5[20]	PMSR5[19]	PMSR5[18]	PMSR5[17]	PMSR5[16]
		PMSR5[15]	PMSR5[14]	PMSR5[13]	PMSR5[12]	PMSR5[11]	PMSR5[10]	PMSR5[9]	PMSR5[8]
		PMSR5[7]	PMSR5[6]	PMSR5[5]	PMSR5[4]	PMSR5[3]	PMSR5[2]	PMSR5[1]	PMSR5[0]
	PMSR6	PMSR6[31]	PMSR6[30]	PMSR6[29]	PMSR6[28]	PMSR6[27]	PMSR6[26]	PMSR6[25]	PMSR6[24]
		PMSR6[23]	PMSR6[22]	PMSR6[21]	PMSR6[20]	PMSR6[19]	PMSR6[18]	PMSR6[17]	PMSR6[16]
		PMSR6[15]	PMSR6[14]	PMSR6[13]	PMSR6[12]	PMSR6[11]	PMSR6[10]	PMSR6[9]	PMSR6[8]
		PMSR6[7]	PMSR6[6]	PMSR6[5]	PMSR6[4]	PMSR6[3]	PMSR6[2]	PMSR6[1]	PMSR6[0]
	PMSR7	-	-	-	-	PMSR7[27]	PMSR7[26]	PMSR7[25]	PMSR7[24]
		PMSR7[23]	PMSR7[22]	PMSR7[21]	PMSR7[20]	PMSR7[19]	PMSR7[18]	PMSR7[17]	PMSR7[16]
		-	-	-	-	PMSR7[11]	PMSR7[10]	PMSR7[9]	PMSR7[8]
		PMSR7[7]	PMSR7[6]	PMSR7[5]	PMSR7[4]	PMSR7[3]	PMSR7[2]	PMSR7[1]	PMSR7[0]
	PMSR8	PMSR8[31]	PMSR8[30]	PMSR8[29]	PMSR8[28]	PMSR8[27]	PMSR8[26]	PMSR8[25]	PMSR8[24]
		PMSR8[23]	PMSR8[22]	PMSR8[21]	PMSR8[20]	PMSR8[19]	PMSR8[18]	PMSR8[17]	PMSR8[16]
		PMSR8[15]	PMSR8[14]	PMSR8[13]	PMSR8[12]	PMSR8[11]	PMSR8[10]	PMSR8[9]	PMSR8[8]
		PMSR8[7]	PMSR8[6]	PMSR8[5]	PMSR8[4]	PMSR8[3]	PMSR8[2]	PMSR8[1]	PMSR8[0]
	PMSR9	-	-	-	-	-	-	-	-
		-	-	PMSR9[21]	PMSR9[20]	PMSR9[19]	PMSR9[18]	PMSR9[17]	PMSR9[16]
		-	-	-	-	-	-	-	-
		-	-	PMSR9[5]	PMSR9[4]	PMSR9[3]	PMSR9[2]	PMSR9[1]	PMSR9[0]
	PMCSR0	-	-	-	-	-	-	-	-
		-	-	-	-	PMCSR0[19]	PMCSR0[18]	PMCSR0[17]	PMCSR0[16]
		-	-	-	-	-	-	-	-
		-	-	-	-	PMCSR0[3]	PMCSR0[2]	PMCSR0[1]	PMCSR0[0]
	PMCSR1	-	-	-	-	-	-	-	-
		PMCSR1[23]	PMCSR1[22]	PMCSR1[21]	PMCSR1[20]	PMCSR1[19]	PMCSR1[18]	PMCSR1[17]	PMCSR1[16]
		-	-	-	-	-	-	-	-
		PMCSR1[7]	PMCSR1[6]	PMCSR1[5]	PMCSR1[4]	PMCSR1[3]	PMCSR1[2]	PMCSR1[1]	PMCSR1[0]
	PMCSR2	-	-	-	-	-	-	PMCSR2[25]	PMCSR2[24]
		PMCSR2[23]	PMCSR2[22]	PMCSR2[21]	PMCSR2[20]	PMCSR2[19]	PMCSR2[18]	PMCSR2[17]	PMCSR2[16]
		-	-	-	-	-	-	PMCSR2[9]	PMCSR2[8]
		PMCSR2[7]	PMCSR2[6]	PMCSR2[5]	PMCSR2[4]	PMCSR2[3]	PMCSR2[2]	PMCSR2[1]	PMCSR2[0]
	PMCSR3	PMCSR3[31]	PMCSR3[30]	PMCSR3[29]	PMCSR3[28]	PMCSR3[27]	PMCSR3[26]	PMCSR3[25]	PMCSR3[24]
		PMCSR3[23]	PMCSR3[22]	PMCSR3[21]	PMCSR3[20]	PMCSR3[19]	PMCSR3[18]	PMCSR3[17]	PMCSR3[16]
		PMCSR3[15]	PMCSR3[14]	PMCSR3[13]	PMCSR3[12]	PMCSR3[11]	PMCSR3[10]	PMCSR3[9]	PMCSR3[8]
		PMCSR3[7]	PMCSR3[6]	PMCSR3[5]	PMCSR3[4]	PMCSR3[3]	PMCSR3[2]	PMCSR3[1]	PMCSR3[0]
	PMCSR4	-	-	-	-	-	-	-	-
		PMCSR4[23]	PMCSR4[22]	PMCSR4[21]	PMCSR4[20]	PMCSR4[19]	PMCSR4[18]	PMCSR4[17]	PMCSR4[16]
		-	-	-	-	-	-	-	-
		PMCSR4[7]	PMCSR4[6]	PMCSR4[5]	PMCSR4[4]	PMCSR4[3]	PMCSR4[2]	PMCSR4[1]	PMCSR4[0]
	PMCSR5	PMCSR5[31]	PMCSR5[30]	PMCSR5[29]	PMCSR5[28]	PMCSR5[27]	PMCSR5[26]	PMCSR5[25]	PMCSR5[24]
		PMCSR5[23]	PMCSR5[22]	PMCSR5[21]	PMCSR5[20]	PMCSR5[19]	PMCSR5[18]	PMCSR5[17]	PMCSR5[16]
		PMCSR5[15]	PMCSR5[14]	PMCSR5[13]	PMCSR5[12]	PMCSR5[11]	PMCSR5[10]	PMCSR5[9]	PMCSR5[8]
		PMCSR5[7]	PMCSR5[6]	PMCSR5[5]	PMCSR5[4]	PMCSR5[3]	PMCSR5[2]	PMCSR5[1]	PMCSR5[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Ports	PMCSR6	PMCSR6[31]	PMCSR6[30]	PMCSR6[29]	PMCSR6[28]	PMCSR6[27]	PMCSR6[26]	PMCSR6[25]	PMCSR6[24]
		PMCSR6[23]	PMCSR6[22]	PMCSR6[21]	PMCSR6[20]	PMCSR6[19]	PMCSR6[18]	PMCSR6[17]	PMCSR6[16]
		PMCSR6[15]	PMCSR6[14]	PMCSR6[13]	PMCSR6[12]	PMCSR6[11]	PMCSR6[10]	PMCSR6[9]	PMCSR6[8]
		PMCSR6[7]	PMCSR6[6]	PMCSR6[5]	PMCSR6[4]	PMCSR6[3]	PMCSR6[2]	PMCSR6[1]	PMCSR6[0]
	PMCSR7	-	-	-	-	PMCSR7[27]	PMCSR7[26]	PMCSR7[25]	PMCSR7[24]
		PMCSR7[23]	PMCSR7[22]	PMCSR7[21]	PMCSR7[20]	PMCSR7[19]	PMCSR7[18]	PMCSR7[17]	PMCSR7[16]
		-	-	-	-	PMCSR7[11]	PMCSR7[10]	PMCSR7[9]	PMCSR7[8]
		PMCSR7[7]	PMCSR7[6]	PMCSR7[5]	PMCSR7[4]	PMCSR7[3]	PMCSR7[2]	PMCSR7[1]	PMCSR7[0]
	PMCSR8	PMCSR8[31]	PMCSR8[30]	PMCSR8[29]	PMCSR8[28]	PMCSR8[27]	PMCSR8[26]	PMCSR8[25]	PMCSR8[24]
		PMCSR8[23]	PMCSR8[22]	PMCSR8[21]	PMCSR8[20]	PMCSR8[19]	PMCSR8[18]	PMCSR8[17]	PMCSR8[16]
		PMCSR8[15]	PMCSR8[14]	PMCSR8[13]	PMCSR8[12]	PMCSR8[11]	PMCSR8[10]	PMCSR8[9]	PMCSR8[8]
		PMCSR8[7]	PMCSR8[6]	PMCSR8[5]	PMCSR8[4]	PMCSR8[3]	PMCSR8[2]	PMCSR8[1]	PMCSR8[0]
	PMCSR9	-	-	-	-	-	-	-	-
		-	-	PMCSR9[21]	PMCSR9[20]	PMCSR9[19]	PMCSR9[18]	PMCSR9[17]	PMCSR9[16]
		-	-	-	-	-	-	-	-
		-	-	PMCSR9[5]	PMCSR9[4]	PMCSR9[3]	PMCSR9[2]	PMCSR9[1]	PMCSR9[0]
	PFCAE0	-	-	-	-	-	-	-	-
		-	-	-	-	PFCAE0[3]	PFCAE0[2]	PFCAE0[1]	PFCAE0[0]
	PFCAE1	PFCAE1[15]	PFCAE1[14]	PFCAE1[13]	PFCAE1[12]	PFCAE1[11]	PFCAE1[10]	PFCAE1[9]	PFCAE1[8]
		PFCAE1[7]	PFCAE1[6]	PFCAE1[5]	PFCAE1[4]	PFCAE1[3]	PFCAE1[2]	PFCAE1[1]	PFCAE1[0]
	PFCAE2	-	-	-	-	-	-	PFCAE2[9]	PFCAE2[8]
		PFCAE2[7]	PFCAE2[6]	PFCAE2[5]	PFCAE2[4]	PFCAE2[3]	PFCAE2[2]	PFCAE2[1]	PFCAE2[0]
	PFCAE3	PFCAE3[15]	PFCAE3[14]	PFCAE3[13]	PFCAE3[12]	PFCAE3[11]	PFCAE3[10]	PFCAE3[9]	PFCAE3[8]
		PFCAE3[7]	PFCAE3[6]	PFCAE3[5]	PFCAE3[4]	PFCAE3[3]	PFCAE3[2]	PFCAE3[1]	PFCAE3[0]
	PFCAE4	-	-	-	-	-	-	-	-
		PFCAE4[7]	PFCAE4[6]	PFCAE4[5]	PFCAE4[4]	PFCAE4[3]	PFCAE4[2]	PFCAE4[1]	PFCAE4[0]
	PFCAE5	PFCAE5[15]	PFCAE5[14]	PFCAE5[13]	PFCAE5[12]	PFCAE5[11]	PFCAE5[10]	PFCAE5[9]	PFCAE5[8]
		PFCAE5[7]	PFCAE5[6]	PFCAE5[5]	PFCAE5[4]	PFCAE5[3]	PFCAE5[2]	PFCAE5[1]	PFCAE5[0]
	PFCAE6	PFCAE6[15]	PFCAE6[14]	PFCAE6[13]	PFCAE6[12]	PFCAE6[11]	PFCAE6[10]	PFCAE6[9]	PFCAE6[8]
		PFCAE6[7]	PFCAE6[6]	PFCAE6[5]	PFCAE6[4]	PFCAE6[3]	PFCAE6[2]	PFCAE6[1]	PFCAE6[0]
	PFCAE7	-	-	-	-	PFCAE7[11]	PFCAE7[10]	PFCAE7[9]	PFCAE7[8]
		PFCAE7[7]	PFCAE7[6]	PFCAE7[5]	PFCAE7[4]	PFCAE7[3]	PFCAE7[2]	PFCAE7[1]	PFCAE7[0]
	PFCAE8	PFCAE8[15]	PFCAE8[14]	PFCAE8[13]	PFCAE8[12]	PFCAE8[11]	PFCAE8[10]	PFCAE8[9]	PFCAE8[8]
		PFCAE8[7]	PFCAE8[6]	PFCAE8[5]	PFCAE8[4]	PFCAE8[3]	PFCAE8[2]	PFCAE8[1]	PFCAE8[0]
	PFCAE9	-	-	-	-	-	-	-	-
		-	-	PFCAE9[5]	PFCAE9[4]	PFCAE9[3]	PFCAE9[2]	PFCAE9[1]	PFCAE9[0]
	PIBC0	-	-	-	-	-	-	-	-
		-	-	-	-	PIBC0[3]	PIBC0[2]	PIBC0[1]	PIBC0[0]
	PIBC1	PIBC1[15]	PIBC1[14]	PIBC1[13]	PIBC1[12]	PIBC1[11]	PIBC1[10]	PIBC1[9]	PIBC1[8]
		PIBC1[7]	PIBC1[6]	PIBC1[5]	PIBC1[4]	PIBC1[3]	PIBC1[2]	PIBC1[1]	PIBC1[0]
	PIBC2	-	-	-	-	-	-	PIBC2[9]	PIBC2[8]
		PIBC2[7]	PIBC2[6]	PIBC2[5]	PIBC2[4]	PIBC2[3]	PIBC2[2]	PIBC2[1]	PIBC2[0]
	PIBC3	PIBC3[15]	PIBC3[14]	PIBC3[13]	PIBC3[12]	PIBC3[11]	PIBC3[10]	PIBC3[9]	PIBC3[8]
		PIBC3[7]	PIBC3[6]	PIBC3[5]	PIBC3[4]	PIBC3[3]	PIBC3[2]	PIBC3[1]	PIBC3[0]
	PIBC4	-	-	-	-	-	-	-	-
		PIBC4[7]	PIBC4[6]	PIBC4[5]	PIBC4[4]	PIBC4[3]	PIBC4[2]	PIBC4[1]	PIBC4[0]
	PIBC5	PIBC5[15]	PIBC5[14]	PIBC5[13]	PIBC5[12]	PIBC5[11]	PIBC5[10]	PIBC5[9]	PIBC5[8]
		PIBC5[7]	PIBC5[6]	PIBC5[5]	PIBC5[4]	PIBC5[3]	PIBC5[2]	PIBC5[1]	PIBC5[0]
	PIBC6	PIBC6[15]	PIBC6[14]	PIBC6[13]	PIBC6[12]	PIBC6[11]	PIBC6[10]	PIBC6[9]	PIBC6[8]
		PIBC6[7]	PIBC6[6]	PIBC6[5]	PIBC6[4]	PIBC6[3]	PIBC6[2]	PIBC6[1]	PIBC6[0]
	PIBC7	-	-	-	-	PIBC7[11]	PIBC7[10]	PIBC7[9]	PIBC7[8]
		PIBC7[7]	PIBC7[6]	PIBC7[5]	PIBC7[4]	PIBC7[3]	PIBC7[2]	PIBC7[1]	PIBC7[0]
	PIBC8	PIBC8[15]	PIBC8[14]	PIBC8[13]	PIBC8[12]	PIBC8[11]	PIBC8[10]	PIBC8[9]	PIBC8[8]
		PIBC8[7]	PIBC8[6]	PIBC8[5]	PIBC8[4]	PIBC8[3]	PIBC8[2]	PIBC8[1]	PIBC8[0]
	PIBC9	-	-	-	-	-	-	-	-
		-	-	PIBC9[5]	PIBC9[4]	PIBC9[3]	PIBC9[2]	PIBC9[1]	PIBC9[0]
	PBDC1	PBDC1[15]	PBDC1[14]	PBDC1[13]	PBDC1[12]	PBDC1[11]	PBDC1[10]	PBDC1[9]	PBDC1[8]
		PBDC1[7]	PBDC1[6]	PBDC1[5]	PBDC1[4]	PBDC1[3]	PBDC1[2]	PBDC1[1]	PBDC1[0]
	PBDC2	-	-	-	-	-	-	PBDC2[9]	PBDC2[8]
		PBDC2[7]	PBDC2[6]	PBDC2[5]	PBDC2[4]	PBDC2[3]	PBDC2[2]	PBDC2[1]	PBDC2[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Ports	PBDC3	PBDC3[15]	PBDC3[14]	PBDC3[13]	PBDC3[12]	PBDC3[11]	PBDC3[10]	PBDC3[9]	PBDC3[8]
		PBDC3[7]	PBDC3[6]	PBDC3[5]	PBDC3[4]	PBDC3[3]	PBDC3[2]	PBDC3[1]	PBDC3[0]
	PBDC4	-	-	-	-	-	-	-	-
		PBDC4[7]	PBDC4[6]	PBDC4[5]	PBDC4[4]	PBDC4[3]	PBDC4[2]	PBDC4[1]	PBDC4[0]
	PBDC5	PBDC5[15]	PBDC5[14]	PBDC5[13]	PBDC5[12]	PBDC5[11]	PBDC5[10]	PBDC5[9]	PBDC5[8]
		PBDC5[7]	PBDC5[6]	PBDC5[5]	PBDC5[4]	PBDC5[3]	PBDC5[2]	PBDC5[1]	PBDC5[0]
	PBDC6	PBDC6[15]	PBDC6[14]	PBDC6[13]	PBDC6[12]	PBDC6[11]	PBDC6[10]	PBDC6[9]	PBDC6[8]
		PBDC6[7]	PBDC6[6]	PBDC6[5]	PBDC6[4]	PBDC6[3]	PBDC6[2]	PBDC6[1]	PBDC6[0]
	PBDC7	-	-	-	-	PBDC7[11]	PBDC7[10]	PBDC7[9]	PBDC7[8]
		PBDC7[7]	PBDC7[6]	PBDC7[5]	PBDC7[4]	PBDC7[3]	PBDC7[2]	PBDC7[1]	PBDC7[0]
	PBDC8	PBDC8[15]	PBDC8[14]	PBDC8[13]	PBDC8[12]	PBDC8[11]	PBDC8[10]	PBDC8[9]	PBDC8[8]
		PBDC8[7]	PBDC8[6]	PBDC8[5]	PBDC8[4]	PBDC8[3]	PBDC8[2]	PBDC8[1]	PBDC8[0]
	PBDC9	-	-	-	-	-	-	-	-
		-	-	PBDC9[5]	PBDC9[4]	PBDC9[3]	PBDC9[2]	PBDC9[1]	PBDC9[0]
	PIPC1	-	-	-	-	-	-	-	-
		PIPC1[7]	PIPC1[6]	PIPC1[5]	PIPC1[4]	PIPC1[3]	PIPC1[2]	PIPC1[1]	PIPC1[0]
	PIPC2	-	-	-	-	-	-	PIPC2[9]	PIPC2[8]
		PIPC2[7]	PIPC2[6]	PIPC2[5]	PIPC2[4]	PIPC2[3]	PIPC2[2]	PIPC2[1]	PIPC2[0]
	PIPC3	PIPC3[15]	PIPC3[14]	PIPC3[13]	PIPC3[12]	PIPC3[11]	PIPC3[10]	PIPC3[9]	PIPC3[8]
		PIPC3[7]	PIPC3[6]	PIPC3[5]	PIPC3[4]	PIPC3[3]	PIPC3[2]	PIPC3[1]	PIPC3[0]
	PIPC4	-	-	-	-	-	-	-	-
		PIPC4[7]	PIPC4[6]	PIPC4[5]	PIPC4[4]	PIPC4[3]	PIPC4[2]	PIPC4[1]	PIPC4[0]
	PIPC5	PIPC5[15]	PIPC5[14]	PIPC5[13]	PIPC5[12]	PIPC5[11]	PIPC5[10]	PIPC5[9]	PIPC5[8]
		PIPC5[7]	PIPC5[6]	PIPC5[5]	PIPC5[4]	PIPC5[3]	PIPC5[2]	PIPC5[1]	PIPC5[0]
	PIPC6	PIPC6[15]	PIPC6[14]	PIPC6[13]	PIPC6[12]	PIPC6[11]	PIPC6[10]	PIPC6[9]	PIPC6[8]
		PIPC6[7]	PIPC6[6]	PIPC6[5]	PIPC6[4]	PIPC6[3]	PIPC6[2]	PIPC6[1]	PIPC6[0]
	PIPC7	-	-	-	-	PIPC7[11]	PIPC7[10]	PIPC7[9]	PIPC7[8]
		PIPC7[7]	PIPC7[6]	PIPC7[5]	PIPC7[4]	PIPC7[3]	PIPC7[2]	PIPC7[1]	PIPC7[0]
	PIPC8	PIPC8[15]	PIPC8[14]	PIPC8[13]	PIPC8[12]	PIPC8[11]	PIPC8[10]	PIPC8[9]	PIPC8[8]
		PIPC8[7]	PIPC8[6]	PIPC8[5]	PIPC8[4]	PIPC8[3]	PIPC8[2]	PIPC8[1]	PIPC8[0]
	PIPC9	-	-	-	-	-	-	-	-
		-	-	PIPC9[5]	PIPC9[4]	PIPC9[3]	PIPC9[2]	PIPC9[1]	PIPC9[0]
	JPPR0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	JPPR0[1]	JPPR0[0]
	JPMC0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	JPMC0[1]	JPMC0[0]
	JPMCSR0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	JPMCSR0[17]	JPMCSR0[16]
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	JPMCSR0[1]	JPMCSR0[0]
	JPIBC0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	JPIBC0[1]	JPIBC0[0]
	SNCR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	ETSEL*4	-	-	SSI3NCE	SSI2NCE	SSI1NCE	SSI0NCE
Power-down modes	STBCR1	STBY	DEEP	-	-	-	-	-	-
	STBCR2	HIZ	-	-	-	-	-	-	MSTP20
	STBCR3	MSTP37*3	MSTP36	MSTP35*3	-	MSTP33	MSTP32	MSTP31	-
	STBCR4	MSTP47	MSTP46	MSTP45	MSTP44	MSTP43	-	-	-
	STBCR5	MSTP57	MSTP56	-	-	-	-	MSTP51	MSTP50
	STBCR6	MSTP67	MSTP66	-	-	-	-	MSTP61*4	MSTP60
	STBCR7	-	-	-	MSTP74	-	-	MSTP71	MSTP70
	STBCR8	-	-	-	MSTP84	MSTP83*3	MSTP82*4	MSTP81	-
	STBCR9	MSTP97	MSTP96	MSTP95	MSTP94	MSTP93	-	MSTP91	-
	STBCR10	MSTP107	MSTP106	MSTP105	-	-	MSTP102*3	MSTP101	-
	STBCR11	-	-	MSTP115	MSTP114	MSTP113	MSTP112	-	-
	STBCR12	-	-	-	-	MSTP123	MSTP122	MSTP121	MSTP120
	SWRSTCR1	AXTALE	SRST16	SRST15	SRST14	SRST13	-	-	-
	SWRSTCR2*2	-	-	-	-	-	-	SRST21	-

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Power-down modes	SYSCR1	-	-	-	VRAME4*5	VRAME3	VRAME2	VRAME1	VRAME0
	SYSCR2	-	-	-	VRAMWE4*5	VRAMWE3	VRAMWE2	VRAMWE1	VRAMWE0
	SYSCR3	-	-	-	-	RRAMWE3	RRAMWE2	RRAMWE1	RRAMWE0
	CPUSTS	-	-	-	ISBUSY	-	-	-	-
	STBREQ1	-	-	STBRQ15	-	STBRQ13*4	STBRQ12*4	-	STBRQ10
	STBREQ2	STBRQ27*3	STBRQ26	STBRQ25	-	-	-	-	-
	STBACK1	-	-	STBAK15	-	STBAK13*4	STBAK12*4	-	STBAK10
	STBACK2	STBAK27*3	STBAK26	STBAK25	-	-	-	-	-
	RRAMKP	-	-	-	-	RRAMKP3	RRAMKP2	RRAMKP1	RRAMKP0
	DSCTR	EBUSKEEPE	RAMBOOT	-	-	-	-	-	-
	DSSSR	-	P2_0	P7_6	P7_9	P5_10	P2_2	P7_2	NMI
		-	RTCAR	P5_9	P7_3	P5_8	P2_7	P2_9	P6_7
	DSESR	-	P2_0E	P7_6E	P7_9E	P5_10E	P2_2E	P7_2E	NMIE
		-	-	P5_9E	P7_3E	P5_8E	P2_7E	P2_9E	P6_7E
	DSFR	IOKEEP	P2_0F	P7_6F	P7_9F	P5_10F	P2_2F	P7_2F	NMIF
		-	RTCARF	P5_9F	P7_3F	P5_8F	P2_7F	P2_9F	P6_7F
	XTALCTR	-	-	-	-	-	-	-	GAIN0
Debugger interface	DAPROM_ PERIPHID4	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		4KB_count[3]	4KB_count[2]	4KB_count[1]	4KB_count[0]	JEP106_code[3]	JEP106_code[2]	JEP106_code[1]	JEP106_code[0]
	DAPROM_ PERIPHID0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Part_Number[7]	Part_Number[6]	Part_Number[5]	Part_Number[4]	Part_Number[3]	Part_Number[2]	Part_Number[1]	Part_Number[0]
	DAPROM_ PERIPHID1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		JEP106_id[3]	JEP106_id[2]	JEP106_id[1]	JEP106_id[0]	Part_Number[11]	Part_Number[10]	Part_Number[9]	Part_Number[8]
	DAPROM_ PERIPHID2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Revision[3]	Revision[2]	Revision[1]	Revision[0]	JEDEC	JEP106_id[6]	JEP106_id[5]	JEP106_id[4]
	DAPROM_ PERIPHID3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		RevAnd[3]	RevAnd[2]	RevAnd[1]	RevAnd[0]	CUSTOM[3]	CUSTOM[2]	CUSTOM[1]	CUSTOM[0]
	DAPROM_COMPID0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	DAPROM_COMPID1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Class[3]	Class[2]	Class[1]	Class[0]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	DAPROM_COMPID2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	DAPROM_COMPID3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	ICEREGMDRSTCTL	-	-	-	-	-	-	-	-
		-	-	-	NIDEN_CPU0	-	-	-	DBGEN_CPU0
		-	-	-	RSTRB_CPU0_ DERSTZ	-	-	-	RSTRB_CPU0_ CPURSTZ
		-	-	-	RSTRB_SYS_ SYSRSTZ	-	RSTRB_CPU_ PRSTDBGZ	RSTRB_CPU_ SYSRSTZ	-

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Debugger interface	ICEREGJTTRCSEL	-	-	-	-	-	-	-	-
		PINSETEN	-	-	TRCMUX_SEL	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	ICEREGCLKPWR CTRL	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		FAKEDBGCTRL	-	-	-	-	-	-	FAKEDBG
	ICEREGLOCK ACCES	ICEREGLOCKACCESS [31]	ICEREGLOCKACCESS [30]	ICEREGLOCKACCESS [29]	ICEREGLOCKACCESS [28]	ICEREGLOCKACCESS [27]	ICEREGLOCKACCESS [26]	ICEREGLOCKACCESS [25]	ICEREGLOCKACCESS [24]
		ICEREGLOCKACCESS [23]	ICEREGLOCKACCESS [22]	ICEREGLOCKACCESS [21]	ICEREGLOCKACCESS [20]	ICEREGLOCKACCESS [19]	ICEREGLOCKACCESS [18]	ICEREGLOCKACCESS [17]	ICEREGLOCKACCESS [16]
		ICEREGLOCKACCESS [15]	ICEREGLOCKACCESS [14]	ICEREGLOCKACCESS [13]	ICEREGLOCKACCESS [12]	ICEREGLOCKACCESS [11]	ICEREGLOCKACCESS [10]	ICEREGLOCKACCESS [9]	ICEREGLOCKACCESS [8]
		ICEREGLOCKACCESS [7]	ICEREGLOCKACCESS [6]	ICEREGLOCKACCESS [5]	ICEREGLOCKACCESS [4]	ICEREGLOCKACCESS [3]	ICEREGLOCKACCESS [2]	ICEREGLOCKACCESS [1]	ICEREGLOCKACCESS [0]
	2ndDAPROM_ PERIPHID4	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		4KB_count[3]	4KB_count[2]	4KB_count[1]	4KB_count[0]	JEP106_code[3]	JEP106_code[2]	JEP106_code[1]	JEP106_code[0]
	2ndDAPROM_ PERIPHID0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Part_Number[7]	Part_Number[6]	Part_Number[5]	Part_Number[4]	Part_Number[3]	Part_Number[2]	Part_Number[1]	Part_Number[0]
	2ndDAPROM_ PERIPHID1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		JEP106_id[3]	JEP106_id[2]	JEP106_id[1]	JEP106_id[0]	Part_Number [11]	Part_Number [10]	Part_Number[9]	Part_Number[8]
	2ndDAPROM_ PERIPHID2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Revision[3]	Revision[2]	Revision[1]	Revision[0]	JEDEC	JEP106_id[6]	JEP106_id[5]	JEP106_id[4]
	2ndDAPROM_ PERIPHID3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		RevAnd[3]	RevAnd[2]	RevAnd[1]	RevAnd[0]	CUSTOM[3]	CUSTOM[2]	CUSTOM[1]	CUSTOM[0]
	2ndDAPROM_ COMPID0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	2ndDAPROM_ COMPID1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Class[3]	Class[2]	Class[1]	Class[0]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	2ndDAPROM_ COMPID2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	2ndDAPROM_ COMPID3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	CPU_ETF_RSZ	-	RSZ[30]	RSZ[29]	RSZ[28]	RSZ[27]	RSZ[26]	RSZ[25]	RSZ[24]
		RSZ[23]	RSZ[22]	RSZ[21]	RSZ[20]	RSZ[19]	RSZ[18]	RSZ[17]	RSZ[16]
		RSZ[15]	RSZ[14]	RSZ[13]	RSZ[12]	RSZ[11]	RSZ[10]	RSZ[9]	RSZ[8]
		RSZ[7]	RSZ[6]	RSZ[5]	RSZ[4]	RSZ[3]	RSZ[2]	RSZ[1]	RSZ[0]
	CPU_ETF_STS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	MemErr	Empty	FtEmpty	TMCReady	Triggered	Full

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Debugger interface	CPU_ETF_RRD	RRD[31]	RRD[30]	RRD[29]	RRD[28]	RRD[27]	RRD[26]	RRD[25]	RRD[24]
		RRD[23]	RRD[22]	RRD[21]	RRD[20]	RRD[19]	RRD[18]	RRD[17]	RRD[16]
		RRD[15]	RRD[14]	RRD[13]	RRD[12]	RRD[11]	RRD[10]	RRD[9]	RRD[8]
		RRD[7]	RRD[6]	RRD[5]	RRD[4]	RRD[3]	RRD[2]	RRD[1]	RRD[0]
	CPU_ETF_RRP	RRP[31]	RRP[30]	RRP[29]	RRP[28]	RRP[27]	RRP[26]	RRP[25]	RRP[24]
		RRP[23]	RRP[22]	RRP[21]	RRP[20]	RRP[19]	RRP[18]	RRP[17]	RRP[16]
		RRP[15]	RRP[14]	RRP[13]	RRP[12]	RRP[11]	RRP[10]	RRP[9]	RRP[8]
		RRP[7]	RRP[6]	RRP[5]	RRP[4]	RRP[3]	RRP[2]	RRP[1]	RRP[0]
	CPU_ETF_RWP	RWP[31]	RWP[30]	RWP[29]	RWP[28]	RWP[27]	RWP[26]	RWP[25]	RWP[24]
		RWP[23]	RWP[22]	RWP[21]	RWP[20]	RWP[19]	RWP[18]	RWP[17]	RWP[16]
		RWP[15]	RWP[14]	RWP[13]	RWP[12]	RWP[11]	RWP[10]	RWP[9]	RWP[8]
		RWP[7]	RWP[6]	RWP[5]	RWP[4]	RWP[3]	RWP[2]	RWP[1]	RWP[0]
	CPU_ETF_TRG	TRG[31]	TRG[30]	TRG[29]	TRG[28]	TRG[27]	TRG[26]	TRG[25]	TRG[24]
		TRG[23]	TRG[22]	TRG[21]	TRG[20]	TRG[19]	TRG[18]	TRG[17]	TRG[16]
		TRG[15]	TRG[14]	TRG[13]	TRG[12]	TRG[11]	TRG[10]	TRG[9]	TRG[8]
		TRG[7]	TRG[6]	TRG[5]	TRG[4]	TRG[3]	TRG[2]	TRG[1]	TRG[0]
	CPU_ETF_CTL	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	TraceCaptEn
	CPU_ETF_RWD	RWD[31]	RWD[30]	RWD[29]	RWD[28]	RWD[27]	RWD[26]	RWD[25]	RWD[24]
		RWD[23]	RWD[22]	RWD[21]	RWD[20]	RWD[19]	RWD[18]	RWD[17]	RWD[16]
		RWD[15]	RWD[14]	RWD[13]	RWD[12]	RWD[11]	RWD[10]	RWD[9]	RWD[8]
		RWD[7]	RWD[6]	RWD[5]	RWD[4]	RWD[3]	RWD[2]	RWD[1]	RWD[0]
	CPU_ETF_MODE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	MODE[1]	MODE[0]
	CPU_ETF_LBUFLEVEL	LBUFLEVEL[31]	LBUFLEVEL[30]	LBUFLEVEL[29]	LBUFLEVEL[28]	LBUFLEVEL[27]	LBUFLEVEL[26]	LBUFLEVEL[25]	LBUFLEVEL[24]
		LBUFLEVEL[23]	LBUFLEVEL[22]	LBUFLEVEL[21]	LBUFLEVEL[20]	LBUFLEVEL[19]	LBUFLEVEL[18]	LBUFLEVEL[17]	LBUFLEVEL[16]
		LBUFLEVEL[15]	LBUFLEVEL[14]	LBUFLEVEL[13]	LBUFLEVEL[12]	LBUFLEVEL[11]	LBUFLEVEL[10]	LBUFLEVEL[9]	LBUFLEVEL[8]
		LBUFLEVEL[7]	LBUFLEVEL[6]	LBUFLEVEL[5]	LBUFLEVEL[4]	LBUFLEVEL[3]	LBUFLEVEL[2]	LBUFLEVEL[1]	LBUFLEVEL[0]
	CPU_ETF_CBUFLEVEL	CBUFLEVEL[31]	CBUFLEVEL[30]	CBUFLEVEL[29]	CBUFLEVEL[28]	CBUFLEVEL[27]	CBUFLEVEL[26]	CBUFLEVEL[25]	CBUFLEVEL[24]
		CBUFLEVEL[23]	CBUFLEVEL[22]	CBUFLEVEL[21]	CBUFLEVEL[20]	CBUFLEVEL[19]	CBUFLEVEL[18]	CBUFLEVEL[17]	CBUFLEVEL[16]
		CBUFLEVEL[15]	CBUFLEVEL[14]	CBUFLEVEL[13]	CBUFLEVEL[12]	CBUFLEVEL[11]	CBUFLEVEL[10]	CBUFLEVEL[9]	CBUFLEVEL[8]
		CBUFLEVEL[7]	CBUFLEVEL[6]	CBUFLEVEL[5]	CBUFLEVEL[4]	CBUFLEVEL[3]	CBUFLEVEL[2]	CBUFLEVEL[1]	CBUFLEVEL[0]
	CPU_ETF_BUFWM	BUFWM[31]	BUFWM[30]	BUFWM[29]	BUFWM[28]	BUFWM[27]	BUFWM[26]	BUFWM[25]	BUFWM[24]
		BUFWM[23]	BUFWM[22]	BUFWM[21]	BUFWM[20]	BUFWM[19]	BUFWM[18]	BUFWM[17]	BUFWM[16]
		BUFWM[15]	BUFWM[14]	BUFWM[13]	BUFWM[12]	BUFWM[11]	BUFWM[10]	BUFWM[9]	BUFWM[8]
		BUFWM[7]	BUFWM[6]	BUFWM[5]	BUFWM[4]	BUFWM[3]	BUFWM[2]	BUFWM[1]	BUFWM[0]
	CPU_ETF_RRPHI	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		RRPHI[7]	RRPHI[6]	RRPHI[5]	RRPHI[4]	RRPHI[3]	RRPHI[2]	RRPHI[1]	RRPHI[0]
	CPU_ETF_RWPHI	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		RWPHI[7]	RWPHI[6]	RWPHI[5]	RWPHI[4]	RWPHI[3]	RWPHI[2]	RWPHI[1]	RWPHI[0]
	CPU_ETF_FFSR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	FtStopped	FlInProg
	CPU_ETF_FFCR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	DrainBuffer	StopOnTrigEvt	StopOnFl	-	TrigOnFl	TrigOnTrigEvt	TrigOnTrigIn
		-	FlushMan	FOnTrigEvt	FOnFlIn	-	-	EnTI	EnFt
	CPU_ETF_PSCR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	PSCount[4]	PSCount[3]	PSCount[2]	PSCount[1]	PSCount[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Debugger interface	CPU_ETF_ CLAIMSET	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CLAIMSET[3]	CLAIMSET[2]	CLAIMSET[1]	CLAIMSET[0]
	CPU_ETF_ CLAIMCLR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CLAIMCLR[3]	CLAIMCLR[2]	CLAIMCLR[1]	CLAIMCLR[0]
	CPU_ETF_LAR	ACCESS_W[31]	ACCESS_W[30]	ACCESS_W[29]	ACCESS_W[28]	ACCESS_W[27]	ACCESS_W[26]	ACCESS_W[25]	ACCESS_W[24]
		ACCESS_W[23]	ACCESS_W[22]	ACCESS_W[21]	ACCESS_W[20]	ACCESS_W[19]	ACCESS_W[18]	ACCESS_W[17]	ACCESS_W[16]
		ACCESS_W[15]	ACCESS_W[14]	ACCESS_W[13]	ACCESS_W[12]	ACCESS_W[11]	ACCESS_W[10]	ACCESS_W[9]	ACCESS_W[8]
		ACCESS_W[7]	ACCESS_W[6]	ACCESS_W[5]	ACCESS_W[4]	ACCESS_W[3]	ACCESS_W[2]	ACCESS_W[1]	ACCESS_W[0]
	CPU_ETF_LSR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	LOCKTYPE	LOCKGRANT	LOCKEXIST
	CPU_ETF_ AUTHSTATUS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		SNID[1]	SNID[0]	SID[1]	SID[0]	NSNID[1]	NSNID[0]	NSID[1]	NSID[0]
	CPU_ETF_DEVID	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	WBUF_DEPTH[2]	WBUF_DEPTH[1]	WBUF_DEPTH[0]	MEMWIDTH[2]	MEMWIDTH[1]	MEMWIDTH[0]
		CONFIGTYPE[1]	CONFIGTYPE[0]	CLKSCHEME	ATBINPORT COUNT[4]	ATBINPORT COUNT[3]	ATBINPORT COUNT[2]	ATBINPORT COUNT[1]	ATBINPORT COUNT[0]
	CPU_ETF_ DEVTYPE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Sub_type[3]	Sub_type[2]	Sub_type[1]	Sub_type[0]	Major_type[3]	Major_type[2]	Major_type[1]	Major_type[0]
	CPU_ETF_ PERIPHID4	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		4KB_count[3]	4KB_count[2]	4KB_count[1]	4KB_count[0]	JEP106_code[3]	JEP106_code[2]	JEP106_code[1]	JEP106_code[0]
	CPU_ETF_ PERIPHID0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Part_Number[7]	Part_Number[6]	Part_Number[5]	Part_Number[4]	Part_Number[3]	Part_Number[2]	Part_Number[1]	Part_Number[0]
	CPU_ETF_ PERIPHID1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		JEP106_id[3]	JEP106_id[2]	JEP106_id[1]	JEP106_id[0]	Part_Number[11]	Part_Number[10]	Part_Number[9]	Part_Number[8]
	CPU_ETF_ PERIPHID2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Revision[3]	Revision[2]	Revision[1]	Revision[0]	JEDEC	JEP106_id[6]	JEP106_id[5]	JEP106_id[4]
	CPU_ETF_ PERIPHID3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		RevAnd[3]	RevAnd[2]	RevAnd[1]	RevAnd[0]	Customer Modified[3]	Customer Modified[2]	Customer Modified[1]	Customer Modified[0]
	CPU_ETF_ COMPID0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	CPU_ETF_ COMPID1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Class[3]	Class[2]	Class[1]	Class[0]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Debugger interface	CPU_ETF_COMPID2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	CPU_ETF_COMPID3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	CPU_CTICS_CTICONTROL	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	GLBEN
	CPU_CTICS_CTIINTACK	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		INTACK[7]	INTACK[6]	INTACK[5]	INTACK[4]	INTACK[3]	INTACK[2]	INTACK[1]	INTACK[0]
	CPU_CTICS_CTIAPPSET	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	APPSET[3]	APPSET[2]	APPSET[1]	APPSET[0]
	CPU_CTICS_CTIAPPCLEAR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	APPCLEAR[3]	APPCLEAR[2]	APPCLEAR[1]	APPCLEAR[0]
	CPU_CTICS_CTIAPPPULSE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	APPULSE[3]	APPULSE[2]	APPULSE[1]	APPULSE[0]
	CPU_CTICS_CTIINEN0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGINEN[3]	TRIGINEN[2]	TRIGINEN[1]	TRIGINEN[0]
	CPU_CTICS_CTIINEN1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGINEN[3]	TRIGINEN[2]	TRIGINEN[1]	TRIGINEN[0]
	CPU_CTICS_CTIINEN2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGINEN[3]	TRIGINEN[2]	TRIGINEN[1]	TRIGINEN[0]
	CPU_CTICS_CTIINEN3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGINEN[3]	TRIGINEN[2]	TRIGINEN[1]	TRIGINEN[0]
	CPU_CTICS_CTIINEN4	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGINEN[3]	TRIGINEN[2]	TRIGINEN[1]	TRIGINEN[0]
	CPU_CTICS_CTIINEN5	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGINEN[3]	TRIGINEN[2]	TRIGINEN[1]	TRIGINEN[0]
	CPU_CTICS_CTIINEN6	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGINEN[3]	TRIGINEN[2]	TRIGINEN[1]	TRIGINEN[0]
	CPU_CTICS_CTIINEN7	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGINEN[3]	TRIGINEN[2]	TRIGINEN[1]	TRIGINEN[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Debugger interface	CPU_CTICS_CTIOUTEN0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGOUTEN[3]	TRIGOUTEN[2]	TRIGOUTEN[1]	TRIGOUTEN[0]
	CPU_CTICS_CTIOUTEN1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGOUTEN[3]	TRIGOUTEN[2]	TRIGOUTEN[1]	TRIGOUTEN[0]
	CPU_CTICS_CTIOUTEN2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGOUTEN[3]	TRIGOUTEN[2]	TRIGOUTEN[1]	TRIGOUTEN[0]
	CPU_CTICS_CTIOUTEN3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGOUTEN[3]	TRIGOUTEN[2]	TRIGOUTEN[1]	TRIGOUTEN[0]
	CPU_CTICS_CTIOUTEN4	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGOUTEN[3]	TRIGOUTEN[2]	TRIGOUTEN[1]	TRIGOUTEN[0]
	CPU_CTICS_CTIOUTEN5	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGOUTEN[3]	TRIGOUTEN[2]	TRIGOUTEN[1]	TRIGOUTEN[0]
	CPU_CTICS_CTIOUTEN6	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGOUTEN[3]	TRIGOUTEN[2]	TRIGOUTEN[1]	TRIGOUTEN[0]
	CPU_CTICS_CTIOUTEN7	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGOUTEN[3]	TRIGOUTEN[2]	TRIGOUTEN[1]	TRIGOUTEN[0]
	CPU_CTICS_CTITRIGINSTATUS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		TRIGINSTATUS[7]	TRIGINSTATUS[6]	TRIGINSTATUS[5]	TRIGINSTATUS[4]	TRIGINSTATUS[3]	TRIGINSTATUS[2]	TRIGINSTATUS[1]	TRIGINSTATUS[0]
	CPU_CTICS_CTITRIGOUTSTATUS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		TRIGOUTSTATUS[7]	TRIGOUTSTATUS[6]	TRIGOUTSTATUS[5]	TRIGOUTSTATUS[4]	TRIGOUTSTATUS[3]	TRIGOUTSTATUS[2]	TRIGOUTSTATUS[1]	TRIGOUTSTATUS[0]
	CPU_CTICS_CTICHINSTATUS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CTCHINSTATUS[3]	CTCHINSTATUS[2]	CTCHINSTATUS[1]	CTCHINSTATUS[0]
	CPU_CTICS_CTICHOUTSTATUS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CTCHOUTSTATUS[3]	CTCHOUTSTATUS[2]	CTCHOUTSTATUS[1]	CTCHOUTSTATUS[0]
	CPU_CTICS_CTIGATE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CTIGATEEN3	CTIGATEEN2	CTIGATEEN1	CTIGATEEN0
	CPU_CTICS_ASICCTL	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	ASICCTL[3]	ASICCTL[2]	ASICCTL[1]	ASICCTL[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Debugger interface	CPU_CTICS_ CLAIMSET	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CLAIMSET[3]	CLAIMSET[2]	CLAIMSET[1]	CLAIMSET[0]
	CPU_CTICS_ CLAIMCLR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CLAIMCLR[3]	CLAIMCLR[2]	CLAIMCLR[1]	CLAIMCLR[0]
	CPU_CTICS_LAR	ACCESS_W[31]	ACCESS_W[30]	ACCESS_W[29]	ACCESS_W[28]	ACCESS_W[27]	ACCESS_W[26]	ACCESS_W[25]	ACCESS_W[24]
		ACCESS_W[23]	ACCESS_W[22]	ACCESS_W[21]	ACCESS_W[20]	ACCESS_W[19]	ACCESS_W[18]	ACCESS_W[17]	ACCESS_W[16]
		ACCESS_W[15]	ACCESS_W[14]	ACCESS_W[13]	ACCESS_W[12]	ACCESS_W[11]	ACCESS_W[10]	ACCESS_W[9]	ACCESS_W[8]
		ACCESS_W[7]	ACCESS_W[6]	ACCESS_W[5]	ACCESS_W[4]	ACCESS_W[3]	ACCESS_W[2]	ACCESS_W[1]	ACCESS_W[0]
	CPU_CTICS_LSR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	LOCKTYPE	LOCKGRANT	LOCKEXIST
	CPU_CTICS_ AUTHSTATUS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		SNID[1]	SNID[0]	SID[1]	SID[0]	NSNID[1]	NSNID[0]	NSID[1]	NSID[0]
	CPU_CTICS_DEVID	-	-	-	-	-	-	-	-
		-	-	-	-	CHANWIDTH[3]	CHANWIDTH[2]	CHANWIDTH[1]	CHANWIDTH[0]
		TRIGWIDTH[7]	TRIGWIDTH[6]	TRIGWIDTH[5]	TRIGWIDTH[4]	TRIGWIDTH[3]	TRIGWIDTH[2]	TRIGWIDTH[1]	TRIGWIDTH[0]
		-	-	-	-	EXTMUXNUM[3]	EXTMUXNUM[2]	EXTMUXNUM[1]	EXTMUXNUM[0]
	CPU_CTICS_ DEVTYPE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Sub_type[3]	Sub_type[2]	Sub_type[1]	Sub_type[0]	Major_type[3]	Major_type[2]	Major_type[1]	Major_type[0]
	CPU_CTICS_ PERIPHID4	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		4KB_count[3]	4KB_count[2]	4KB_count[1]	4KB_count[0]	JEP106_code[3]	JEP106_code[2]	JEP106_code[1]	JEP106_code[0]
	CPU_CTICS_ PERIPHID0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Part_Number[7]	Part_Number[6]	Part_Number[5]	Part_Number[4]	Part_Number[3]	Part_Number[2]	Part_Number[1]	Part_Number[0]
	CPU_CTICS_ PERIPHID1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		JEP106_id[3]	JEP106_id[2]	JEP106_id[1]	JEP106_id[0]	Part_Number [11]	Part_Number [10]	Part_Number[9]	Part_Number[8]
	CPU_CTICS_ PERIPHID2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Revision[3]	Revision[2]	Revision[1]	Revision[0]	JEDEC	JEP106_id[6]	JEP106_id[5]	JEP106_id[4]
	CPU_CTICS_ PERIPHID3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		RevAnd[3]	RevAnd[2]	RevAnd[1]	RevAnd[0]	CUSTOM[3]	CUSTOM[2]	CUSTOM[1]	CUSTOM[0]
	CPU_CTICS_ COMPID0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	CPU_CTICS_ COMPID1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Class[3]	Class[2]	Class[1]	Class[0]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	CPU_CTICS_ COMPID2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Debugger interface	CPU_CTICS_ COMPID3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	CPU_TPIU_ Supportedportsizes	SIZE[31]	SIZE[30]	SIZE[29]	SIZE[28]	SIZE[27]	SIZE[26]	SIZE[25]	SIZE[24]
		SIZE[23]	SIZE[22]	SIZE[21]	SIZE[20]	SIZE[19]	SIZE[18]	SIZE[17]	SIZE[16]
		SIZE[15]	SIZE[14]	SIZE[13]	SIZE[12]	SIZE[11]	SIZE[10]	SIZE[9]	SIZE[8]
		SIZE[7]	SIZE[6]	SIZE[5]	SIZE[4]	SIZE[3]	SIZE[2]	SIZE[1]	SIZE[0]
	CPU_TPIU_ Currentportsize	SIZE[31]	SIZE[30]	SIZE[29]	SIZE[28]	SIZE[27]	SIZE[26]	SIZE[25]	SIZE[24]
		SIZE[23]	SIZE[22]	SIZE[21]	SIZE[20]	SIZE[19]	SIZE[18]	SIZE[17]	SIZE[16]
		SIZE[15]	SIZE[14]	SIZE[13]	SIZE[12]	SIZE[11]	SIZE[10]	SIZE[9]	SIZE[8]
		SIZE[7]	SIZE[6]	SIZE[5]	SIZE[4]	SIZE[3]	SIZE[2]	SIZE[1]	SIZE[0]
	CPU_TPIU_ Supportedtriggermodes	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	TrgRun	Triggered
		-	-	-	-	-	-	-	TCount8
		-	-	-	Multipliers[4]	Multipliers[3]	Multipliers[2]	Multipliers[1]	Multipliers[0]
	CPU_TPIU_ TriggerCounter value	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		TrigCount[7]	TrigCount[6]	TrigCount[5]	TrigCount[4]	TrigCount[3]	TrigCount[2]	TrigCount[1]	TrigCount[0]
	CPU_TPIU_ Triggermultiplier	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	TrgRun	Triggered
		-	-	-	-	-	-	-	TCount8
		-	-	-	Multipliers[4]	Multipliers[3]	Multipliers[2]	Multipliers[1]	Multipliers[0]
	CPU_TPIU_ Supportedtestpattern/ modes	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	Mode[1]	Mode[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	Pattern[3]	Pattern[2]	Pattern[1]	Pattern[0]
	CPU_TPIU_ Currenttestpattern/ mode	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	Mode[1]	Mode[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	Pattern[3]	Pattern[2]	Pattern[1]	Pattern[0]
	CPU_TPIU_ Testpatternrepeat counter	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		PattCount[7]	PattCount[6]	PattCount[5]	PattCount[4]	PattCount[3]	PattCount[2]	PattCount[1]	PattCount[0]
	CPU_TPIU_ Formatterandflush status	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	Mode[1]	Mode[0]
		-	-	-	-	-	-	-	-
		-	-	StopTrig	StopFI	-	TCP	FTS	FIP
	CPU_TPIU_ Formatterandflush control	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	TrigFI	TrigEvt	TrigIn
		-	FOnMan	FOnTrig	FOnFIIn	-	-	EnFCont	EnFTC
	CPU_TPIU_ Formatter synchronization counter	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CycCount[11]	CycCount[10]	CycCount[9]	CycCount[8]
		CycCount[7]	CycCount[6]	CycCount[5]	CycCount[4]	CycCount[3]	CycCount[2]	CycCount[1]	CycCount[0]
	CPU_TPIU_ CLAIMSET	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CLAIMSET[3]	CLAIMSET[2]	CLAIMSET[1]	CLAIMSET[0]
	CPU_TPIU_ CLAIMCLR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CLAIMCLR[3]	CLAIMCLR[2]	CLAIMCLR[1]	CLAIMCLR[0]
	CPU_TPIU_LAR	ACCESS_W[31]	ACCESS_W[30]	ACCESS_W[29]	ACCESS_W[28]	ACCESS_W[27]	ACCESS_W[26]	ACCESS_W[25]	ACCESS_W[24]
		ACCESS_W[23]	ACCESS_W[22]	ACCESS_W[21]	ACCESS_W[20]	ACCESS_W[19]	ACCESS_W[18]	ACCESS_W[17]	ACCESS_W[16]
		ACCESS_W[15]	ACCESS_W[14]	ACCESS_W[13]	ACCESS_W[12]	ACCESS_W[11]	ACCESS_W[10]	ACCESS_W[9]	ACCESS_W[8]
		ACCESS_W[7]	ACCESS_W[6]	ACCESS_W[5]	ACCESS_W[4]	ACCESS_W[3]	ACCESS_W[2]	ACCESS_W[1]	ACCESS_W[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Debugger interface	CPU_TPIU_LSR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	LOCKTYPE	LOCKGRANT	LOCKEXIST
	CPU_TPIU_ AUTHSTATUS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		SNID[1]	SNID[0]	SID[1]	SID[0]	NSNID[1]	NSNID[0]	NSID[1]	NSID[0]
	CPU_TPIU_DEVID	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	ID[12]	ID[11]	ID[10]	ID[9]	ID[8]
		ID[7]	ID[6]	ID[5]	ID[4]	ID[3]	ID[2]	ID[1]	ID[0]
	CPU_TPIU_ DEVTYPE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Sub_type[3]	Sub_type[2]	Sub_type[1]	Sub_type[0]	Main_type[3]	Main_type[2]	Main_type[1]	Main_type[0]
	CPU_TPIU_ PERIPHID4	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		4KB_count[3]	4KB_count[2]	4KB_count[1]	4KB_count[0]	JEP106_code[3]	JEP106_code[2]	JEP106_code[1]	JEP106_code[0]
	CPU_TPIU_ PERIPHID0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Part_Number[7]	Part_Number[6]	Part_Number[5]	Part_Number[4]	Part_Number[3]	Part_Number[2]	Part_Number[1]	Part_Number[0]
	CPU_TPIU_ PERIPHID1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		JEP106_id[3]	JEP106_id[2]	JEP106_id[1]	JEP106_id[0]	Part_Number [11]	Part_Number [10]	Part_Number[9]	Part_Number[8]
	CPU_TPIU_ PERIPHID2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Revision[3]	Revision[2]	Revision[1]	Revision[0]	JEDEC	JEP106_id[6]	JEP106_id[5]	JEP106_id[4]
	CPU_TPIU_ PERIPHID3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		RevAnd[3]	RevAnd[2]	RevAnd[1]	RevAnd[0]	Customer Modified[3]	Customer Modified[2]	Customer Modified[1]	Customer Modified[0]
	CPU_TPIU_COMPID 0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	CPU_TPIU_COMPID 1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Class[3]	Class[2]	Class[1]	Class[0]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	CPU_TPIU_COMPID 2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	CPU_TPIU_COMPID 3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	CPU_TraceFunnel_ FUNCTL	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	HT[3]	HT[2]	HT[1]	HT[0]
		EnS7	EnS6	EnS5	EnS4	EnS3	EnS2	EnS1	EnS0

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Debugger interface	CPU_TraceFunnel_PRICTL	-	-	-	-	-	-	-	-
		PriPort7[2]	PriPort7[1]	PriPort7[0]	PriPort6[2]	PriPort6[1]	PriPort6[0]	PriPort5[2]	PriPort5[1]
		PriPort5[0]	PriPort4[2]	PriPort4[1]	PriPort4[0]	PriPort3[2]	PriPort3[1]	PriPort3[0]	PriPort2[2]
		PriPort2[1]	PriPort2[0]	PriPort1[2]	PriPort1[1]	PriPort1[0]	PriPort0[2]	PriPort0[1]	PriPort0[0]
	CPU_TraceFunnel_CLAIMSET	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CLAIMSET[3]	CLAIMSET[2]	CLAIMSET[1]	CLAIMSET[0]
	CPU_TraceFunnel_CLAIMCLR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CLAIMCLR[3]	CLAIMCLR[2]	CLAIMCLR[1]	CLAIMCLR[0]
	CPU_TraceFunnel_LAR	ACCESS_W[31]	ACCESS_W[30]	ACCESS_W[29]	ACCESS_W[28]	ACCESS_W[27]	ACCESS_W[26]	ACCESS_W[25]	ACCESS_W[24]
		ACCESS_W[23]	ACCESS_W[22]	ACCESS_W[21]	ACCESS_W[20]	ACCESS_W[19]	ACCESS_W[18]	ACCESS_W[17]	ACCESS_W[16]
		ACCESS_W[15]	ACCESS_W[14]	ACCESS_W[13]	ACCESS_W[12]	ACCESS_W[11]	ACCESS_W[10]	ACCESS_W[9]	ACCESS_W[8]
		ACCESS_W[7]	ACCESS_W[6]	ACCESS_W[5]	ACCESS_W[4]	ACCESS_W[3]	ACCESS_W[2]	ACCESS_W[1]	ACCESS_W[0]
	CPU_TraceFunnel_LSR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	LOCKTYPE	LOCKGRANT	LOCKEXIST
	CPU_TraceFunnel_AUTHSTATUS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		SNID[1]	SNID[0]	SID[1]	SID[0]	NSNID[1]	NSNID[0]	NSID[1]	NSID[0]
	CPU_TraceFunnel_DEVID	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ID[7]	ID[6]	ID[5]	ID[4]	ID[3]	ID[2]	ID[1]	ID[0]
	CPU_TraceFunnel_DEVTYPE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		TYPE[7]	TYPE[6]	TYPE[5]	TYPE[4]	TYPE[3]	TYPE[2]	TYPE[1]	TYPE[0]
	CPU_TraceFunnel_PERIPHID4	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		4KB_count[3]	4KB_count[2]	4KB_count[1]	4KB_count[0]	JEP106_code[3]	JEP106_code[2]	JEP106_code[1]	JEP106_code[0]
	CPU_TraceFunnel_PERIPHID0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Part_Number[7]	Part_Number[6]	Part_Number[5]	Part_Number[4]	Part_Number[3]	Part_Number[2]	Part_Number[1]	Part_Number[0]
	CPU_TraceFunnel_PERIPHID1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		JEP106_id[3]	JEP106_id[2]	JEP106_id[1]	JEP106_id[0]	Part_Number[11]	Part_Number[10]	Part_Number[9]	Part_Number[8]
	CPU_TraceFunnel_PERIPHID2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Revision[3]	Revision[2]	Revision[1]	Revision[0]	JEDEC	JEP106_id[6]	JEP106_id[5]	JEP106_id[4]
	CPU_TraceFunnel_PERIPHID3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		RevAnd[3]	RevAnd[2]	RevAnd[1]	RevAnd[0]	CUSTOM[3]	CUSTOM[2]	CUSTOM[1]	CUSTOM[0]
	CPU_TraceFunnel_COMPID0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	CPU_TraceFunnel_COMPID1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Class[3]	Class[2]	Class[1]	Class[0]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Debugger interface	CPU_TraceFunnel_ COMPID2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	CPU_TraceFunnel_ COMPID3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	CA9_DBG_ DBGDIDR	WRPs[3]	WRPs[2]	WRPs[1]	WRPs[0]	BRPs[3]	BRPs[2]	BRPs[1]	BRPs[0]
		CTX_CMPs[3]	CTX_CMPs[2]	CTX_CMPs[1]	CTX_CMPs[0]	Version[3]	Version[2]	Version[1]	Version[0]
		DEVID_imp	nSUHD_imp	PCSR_imp	SE_imp	-	-	-	-
		Variant[3]	Variant[2]	Variant[1]	Variant[0]	Revision[3]	Revision[2]	Revision[1]	Revision[0]
	CA9_DBG_ DBGWFAR	ADDRESS[31]	ADDRESS[30]	ADDRESS[29]	ADDRESS[28]	ADDRESS[27]	ADDRESS[26]	ADDRESS[25]	ADDRESS[24]
		ADDRESS[23]	ADDRESS[22]	ADDRESS[21]	ADDRESS[20]	ADDRESS[19]	ADDRESS[18]	ADDRESS[17]	ADDRESS[16]
		ADDRESS[15]	ADDRESS[14]	ADDRESS[13]	ADDRESS[12]	ADDRESS[11]	ADDRESS[10]	ADDRESS[9]	ADDRESS[8]
		ADDRESS[7]	ADDRESS[6]	ADDRESS[5]	ADDRESS[4]	ADDRESS[3]	ADDRESS[2]	ADDRESS[1]	ADDRESS[0]
	CA9_DBG_ DBGVCR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		FIQ	IRQ	-	DataAbort	PrefetchAbort	SVC	Undef	Reset
	CA9_DBG_ DBGDTRRXext	Host_to_target_data [31]	Host_to_target_data [30]	Host_to_target_data [29]	Host_to_target_data [28]	Host_to_target_data [27]	Host_to_target_data [26]	Host_to_target_data [25]	Host_to_target_data [24]
		Host_to_target_data [23]	Host_to_target_data [22]	Host_to_target_data [21]	Host_to_target_data [20]	Host_to_target_data [19]	Host_to_target_data [18]	Host_to_target_data [17]	Host_to_target_data [16]
		Host_to_target_data [15]	Host_to_target_data [14]	Host_to_target_data [13]	Host_to_target_data [12]	Host_to_target_data [11]	Host_to_target_data [10]	Host_to_target_data [9]	Host_to_target_data [8]
		Host_to_target_data [7]	Host_to_target_data [6]	Host_to_target_data [5]	Host_to_target_data [4]	Host_to_target_data [3]	Host_to_target_data [2]	Host_to_target_data [1]	Host_to_target_data [0]
	CA9_DBG_DBGITR	Instruction[31]	Instruction[30]	Instruction[29]	Instruction[28]	Instruction[27]	Instruction[26]	Instruction[25]	Instruction[24]
		Instruction[23]	Instruction[22]	Instruction[21]	Instruction[20]	Instruction[19]	Instruction[18]	Instruction[17]	Instruction[16]
		Instruction[15]	Instruction[14]	Instruction[13]	Instruction[12]	Instruction[11]	Instruction[10]	Instruction[9]	Instruction[8]
		Instruction[7]	Instruction[6]	Instruction[5]	Instruction[4]	Instruction[3]	Instruction[2]	Instruction[1]	Instruction[0]
	CA9_DBG_ DBGPCSR	PC[29]	PC[28]	PC[27]	PC[26]	PC[25]	PC[24]	PC[23]	PC[22]
		PC[21]	PC[20]	PC[19]	PC[18]	PC[17]	PC[16]	PC[15]	PC[14]
		PC[13]	PC[12]	PC[11]	PC[10]	PC[9]	PC[8]	PC[7]	PC[6]
		PC[5]	PC[4]	PC[3]	PC[2]	PC[1]	PC[0]	Meaning_of_PC[1]	Meaning_of_PC[0]
	CA9_DBG_ DBGDSCRext	-	Rxfull	Txfull	-	RXfull_I	TXfull_I	PipeAdv	InstrCompl_I
		-	-	ExtDCCmode[1]	ExtDCCmode[0]	ADAdiscard,	NS	SPNIDdis	SPIDdis
		MDBGen	HDBGen	ITREn	UDCCdis	INTdis	DBGack	-	UND_I
		ADABORT_I	SDABORT_I	MOE[3]	MOE[2]	MOE[1]	MOE[0]	RESTARTED	HALTED
	CA9_DBG_ DBGDTRTXext	Target_to_host_data [31]	Target_to_host_data [30]	Target_to_host_data [29]	Target_to_host_data [28]	Target_to_host_data [27]	Target_to_host_data [26]	Target_to_host_data [25]	Target_to_host_data [24]
		Target_to_host_data [23]	Target_to_host_data [22]	Target_to_host_data [21]	Target_to_host_data [20]	Target_to_host_data [19]	Target_to_host_data [18]	Target_to_host_data [17]	Target_to_host_data [16]
		Target_to_host_data [15]	Target_to_host_data [14]	Target_to_host_data [13]	Target_to_host_data [12]	Target_to_host_data [11]	Target_to_host_data [10]	Target_to_host_data [9]	Target_to_host_data [8]
		Target_to_host_data [7]	Target_to_host_data [6]	Target_to_host_data [5]	Target_to_host_data [4]	Target_to_host_data [3]	Target_to_host_data [2]	Target_to_host_data [1]	Target_to_host_data [0]
	CA9_DBG_ DBGDRCR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	Cancel BIU Requests	Clear Sticky Pipeline Advanceflag	Clear Sticky Exceptions flags	Restart request	Halt request
	CA9_DBG_DBGBVR 0	Breakpoint[31]	Breakpoint[30]	Breakpoint[29]	Breakpoint[28]	Breakpoint[27]	Breakpoint[26]	Breakpoint[25]	Breakpoint[24]
		Breakpoint[23]	Breakpoint[22]	Breakpoint[21]	Breakpoint[20]	Breakpoint[19]	Breakpoint[18]	Breakpoint[17]	Breakpoint[16]
		Breakpoint[15]	Breakpoint[14]	Breakpoint[13]	Breakpoint[12]	Breakpoint[11]	Breakpoint[10]	Breakpoint[9]	Breakpoint[8]
		Breakpoint[7]	Breakpoint[6]	Breakpoint[5]	Breakpoint[4]	Breakpoint[3]	Breakpoint[2]	Breakpoint[1]	Breakpoint[0]
	CA9_DBG_DBGBVR 1	Breakpoint[31]	Breakpoint[30]	Breakpoint[29]	Breakpoint[28]	Breakpoint[27]	Breakpoint[26]	Breakpoint[25]	Breakpoint[24]
		Breakpoint[23]	Breakpoint[22]	Breakpoint[21]	Breakpoint[20]	Breakpoint[19]	Breakpoint[18]	Breakpoint[17]	Breakpoint[16]
		Breakpoint[15]	Breakpoint[14]	Breakpoint[13]	Breakpoint[12]	Breakpoint[11]	Breakpoint[10]	Breakpoint[9]	Breakpoint[8]
		Breakpoint[7]	Breakpoint[6]	Breakpoint[5]	Breakpoint[4]	Breakpoint[3]	Breakpoint[2]	Breakpoint[1]	Breakpoint[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Debugger interface	CA9_DBG_DBGBVR2	Breakpoint[31]	Breakpoint[30]	Breakpoint[29]	Breakpoint[28]	Breakpoint[27]	Breakpoint[26]	Breakpoint[25]	Breakpoint[24]
		Breakpoint[23]	Breakpoint[22]	Breakpoint[21]	Breakpoint[20]	Breakpoint[19]	Breakpoint[18]	Breakpoint[17]	Breakpoint[16]
		Breakpoint[15]	Breakpoint[14]	Breakpoint[13]	Breakpoint[12]	Breakpoint[11]	Breakpoint[10]	Breakpoint[9]	Breakpoint[8]
		Breakpoint[7]	Breakpoint[6]	Breakpoint[5]	Breakpoint[4]	Breakpoint[3]	Breakpoint[2]	Breakpoint[1]	Breakpoint[0]
	CA9_DBG_DBGBVR3	Breakpoint[31]	Breakpoint[30]	Breakpoint[29]	Breakpoint[28]	Breakpoint[27]	Breakpoint[26]	Breakpoint[25]	Breakpoint[24]
		Breakpoint[23]	Breakpoint[22]	Breakpoint[21]	Breakpoint[20]	Breakpoint[19]	Breakpoint[18]	Breakpoint[17]	Breakpoint[16]
		Breakpoint[15]	Breakpoint[14]	Breakpoint[13]	Breakpoint[12]	Breakpoint[11]	Breakpoint[10]	Breakpoint[9]	Breakpoint[8]
		Breakpoint[7]	Breakpoint[6]	Breakpoint[5]	Breakpoint[4]	Breakpoint[3]	Breakpoint[2]	Breakpoint[1]	Breakpoint[0]
	CA9_DBG_DBGBVR4	Breakpoint[31]	Breakpoint[30]	Breakpoint[29]	Breakpoint[28]	Breakpoint[27]	Breakpoint[26]	Breakpoint[25]	Breakpoint[24]
		Breakpoint[23]	Breakpoint[22]	Breakpoint[21]	Breakpoint[20]	Breakpoint[19]	Breakpoint[18]	Breakpoint[17]	Breakpoint[16]
		Breakpoint[15]	Breakpoint[14]	Breakpoint[13]	Breakpoint[12]	Breakpoint[11]	Breakpoint[10]	Breakpoint[9]	Breakpoint[8]
		Breakpoint[7]	Breakpoint[6]	Breakpoint[5]	Breakpoint[4]	Breakpoint[3]	Breakpoint[2]	Breakpoint[1]	Breakpoint[0]
	CA9_DBG_DBGBVR5	Breakpoint[31]	Breakpoint[30]	Breakpoint[29]	Breakpoint[28]	Breakpoint[27]	Breakpoint[26]	Breakpoint[25]	Breakpoint[24]
		Breakpoint[23]	Breakpoint[22]	Breakpoint[21]	Breakpoint[20]	Breakpoint[19]	Breakpoint[18]	Breakpoint[17]	Breakpoint[16]
		Breakpoint[15]	Breakpoint[14]	Breakpoint[13]	Breakpoint[12]	Breakpoint[11]	Breakpoint[10]	Breakpoint[9]	Breakpoint[8]
		Breakpoint[7]	Breakpoint[6]	Breakpoint[5]	Breakpoint[4]	Breakpoint[3]	Breakpoint[2]	Breakpoint[1]	Breakpoint[0]
	CA9_DBG_DBGBCR0	-	-	-	address_mask[4]	address_mask[3]	address_mask[2]	address_mask[1]	address_mask[0]
		-	M[2]	M[1]	M[0]	Linked BRP[3]	Linked BRP[2]	Linked BRP[1]	Linked BRP[0]
		Sec_state_cont[1]	Sec_state_cont[0]	-	-	-	-	-	Byte Add_sel[3]
		Byte Add_sel[2]	Byte Add_sel[1]	Byte Add_sel[0]	-	-	SP[1]	SP[0]	B
	CA9_DBG_DBGBCR1	-	-	-	address_mask[4]	address_mask[3]	address_mask[2]	address_mask[1]	address_mask[0]
		-	M[2]	M[1]	M[0]	Linked BRP[3]	Linked BRP[2]	Linked BRP[1]	Linked BRP[0]
		Sec_state_cont[1]	Sec_state_cont[0]	-	-	-	-	-	Byte Add_sel[3]
		Byte Add_sel[2]	Byte Add_sel[1]	Byte Add_sel[0]	-	-	SP[1]	SP[0]	B
	CA9_DBG_DBGBCR2	-	-	-	address_mask[4]	address_mask[3]	address_mask[2]	address_mask[1]	address_mask[0]
		-	M[2]	M[1]	M[0]	Linked BRP[3]	Linked BRP[2]	Linked BRP[1]	Linked BRP[0]
		Sec_state_cont[1]	Sec_state_cont[0]	-	-	-	-	-	Byte Add_sel[3]
		Byte Add_sel[2]	Byte Add_sel[1]	Byte Add_sel[0]	-	-	SP[1]	SP[0]	B
	CA9_DBG_DBGBCR3	-	-	-	address_mask[4]	address_mask[3]	address_mask[2]	address_mask[1]	address_mask[0]
		-	M[2]	M[1]	M[0]	Linked BRP[3]	Linked BRP[2]	Linked BRP[1]	Linked BRP[0]
		Sec_state_cont[1]	Sec_state_cont[0]	-	-	-	-	-	Byte Add_sel[3]
		Byte Add_sel[2]	Byte Add_sel[1]	Byte Add_sel[0]	-	-	SP[1]	SP[0]	B
	CA9_DBG_DBGBCR4	-	-	-	address_mask[4]	address_mask[3]	address_mask[2]	address_mask[1]	address_mask[0]
		-	M[2]	M[1]	M[0]	Linked BRP[3]	Linked BRP[2]	Linked BRP[1]	Linked BRP[0]
		Sec_state_cont[1]	Sec_state_cont[0]	-	-	-	-	-	Byte Add_sel[3]
		Byte Add_sel[2]	Byte Add_sel[1]	Byte Add_sel[0]	-	-	SP[1]	SP[0]	B
	CA9_DBG_DBGBCR5	-	-	-	address_mask[4]	address_mask[3]	address_mask[2]	address_mask[1]	address_mask[0]
		-	M[2]	M[1]	M[0]	Linked BRP[3]	Linked BRP[2]	Linked BRP[1]	Linked BRP[0]
		Sec_state_cont[1]	Sec_state_cont[0]	-	-	-	-	-	Byte Add_sel[3]
		Byte Add_sel[2]	Byte Add_sel[1]	Byte Add_sel[0]	-	-	SP[1]	SP[0]	B
	CA9_DBG_DBGWVR0	Watchpoint[31]	Watchpoint[30]	Watchpoint[29]	Watchpoint[28]	Watchpoint[27]	Watchpoint[26]	Watchpoint[25]	Watchpoint[24]
		Watchpoint[23]	Watchpoint[22]	Watchpoint[21]	Watchpoint[20]	Watchpoint[19]	Watchpoint[18]	Watchpoint[17]	Watchpoint[16]
		Watchpoint[15]	Watchpoint[14]	Watchpoint[13]	Watchpoint[12]	Watchpoint[11]	Watchpoint[10]	Watchpoint[9]	Watchpoint[8]
		Watchpoint[7]	Watchpoint[6]	Watchpoint[5]	Watchpoint[4]	Watchpoint[3]	Watchpoint[2]	Watchpoint[1]	Watchpoint[0]
	CA9_DBG_DBGWVR1	Watchpoint[31]	Watchpoint[30]	Watchpoint[29]	Watchpoint[28]	Watchpoint[27]	Watchpoint[26]	Watchpoint[25]	Watchpoint[24]
		Watchpoint[23]	Watchpoint[22]	Watchpoint[21]	Watchpoint[20]	Watchpoint[19]	Watchpoint[18]	Watchpoint[17]	Watchpoint[16]
		Watchpoint[15]	Watchpoint[14]	Watchpoint[13]	Watchpoint[12]	Watchpoint[11]	Watchpoint[10]	Watchpoint[9]	Watchpoint[8]
		Watchpoint[7]	Watchpoint[6]	Watchpoint[5]	Watchpoint[4]	Watchpoint[3]	Watchpoint[2]	Watchpoint[1]	Watchpoint[0]
	CA9_DBG_DBGWVR2	Watchpoint[31]	Watchpoint[30]	Watchpoint[29]	Watchpoint[28]	Watchpoint[27]	Watchpoint[26]	Watchpoint[25]	Watchpoint[24]
		Watchpoint[23]	Watchpoint[22]	Watchpoint[21]	Watchpoint[20]	Watchpoint[19]	Watchpoint[18]	Watchpoint[17]	Watchpoint[16]
		Watchpoint[15]	Watchpoint[14]	Watchpoint[13]	Watchpoint[12]	Watchpoint[11]	Watchpoint[10]	Watchpoint[9]	Watchpoint[8]
		Watchpoint[7]	Watchpoint[6]	Watchpoint[5]	Watchpoint[4]	Watchpoint[3]	Watchpoint[2]	Watchpoint[1]	Watchpoint[0]
	CA9_DBG_DBGWVR3	Watchpoint[31]	Watchpoint[30]	Watchpoint[29]	Watchpoint[28]	Watchpoint[27]	Watchpoint[26]	Watchpoint[25]	Watchpoint[24]
		Watchpoint[23]	Watchpoint[22]	Watchpoint[21]	Watchpoint[20]	Watchpoint[19]	Watchpoint[18]	Watchpoint[17]	Watchpoint[16]
		Watchpoint[15]	Watchpoint[14]	Watchpoint[13]	Watchpoint[12]	Watchpoint[11]	Watchpoint[10]	Watchpoint[9]	Watchpoint[8]
		Watchpoint[7]	Watchpoint[6]	Watchpoint[5]	Watchpoint[4]	Watchpoint[3]	Watchpoint[2]	Watchpoint[1]	Watchpoint[0]
	CA9_DBG_DBGWCR0	-	-	-	address_mask[4]	address_mask[3]	address_mask[2]	address_mask[1]	address_mask[0]
		-	-	-	E	Linked BRP[3]	Linked BRP[2]	Linked BRP[1]	Linked BRP[0]
		Sec_state_cont[1]	Sec_state_cont[0]	-	-	-	-	-	Byte Add_sel[3]
		Byte add_sel[2]	Byte Add_sel[1]	Byte Add_sel[0]	L/S[1]	L/S[0]	SP[1]	SP[0]	W

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Debugger interface	CA9_DBG_ DBGWCR1	-	-	-	address_mask[4]	address_mask[3]	address_mask[2]	address_mask[1]	address_mask[0]
		-	-	-	E	Linked BRP[3]	Linked BRP[2]	Linked BRP[1]	Linked BRP[0]
		Sec_state_cont[1]	Sec_state_cont[0]	-	-	-	-	-	Byte Add_sel[3]
		Byte Add_sel[2]	Byte Add_sel[1]	Byte Add_sel[0]	L/S[1]	L/S[0]	SP[1]	SP[0]	W
	CA9_DBG_ DBGWCR2	-	-	-	address_mask[4]	address_mask[3]	address_mask[2]	address_mask[1]	address_mask[0]
		-	-	-	E	Linked BRP[3]	Linked BRP[2]	Linked BRP[1]	Linked BRP[0]
		Sec_state_cont[1]	Sec_state_cont[0]	-	-	-	-	-	Byte Add_sel[3]
		Byte Add_sel[2]	Byte Add_sel[1]	Byte Add_sel[0]	L/S[1]	L/S[0]	SP[1]	SP[0]	W
	CA9_DBG_ DBGWCR3	-	-	-	address_mask[4]	address_mask[3]	address_mask[2]	address_mask[1]	address_mask[0]
		-	-	-	E	Linked BRP[3]	Linked BRP[2]	Linked BRP[1]	Linked BRP[0]
		Sec_state_cont[1]	Sec_state_cont[0]	-	-	-	-	-	Byte Add_sel[3]
		Byte Add_sel[2]	Byte Add_sel[1]	Byte Add_sel[0]	L/S[1]	L/S[0]	SP[1]	SP[0]	W
	CA9_DBG_MIDR	Implementer[7]	Implementer[6]	Implementer[5]	Implementer[4]	Implementer[3]	Implementer[2]	Implementer[1]	Implementer[0]
		Variant[3]	Variant[2]	Variant[1]	Variant[0]	Architecture[3]	Architecture[2]	Architecture[1]	Architecture[0]
		part number[11]	part number[10]	part number[9]	part number[8]	part number[7]	part number[6]	part number[5]	part number[4]
		part number[3]	part number[2]	part number[1]	part number[0]	Revision[3]	Revision[2]	Revision[1]	Revision[0]
	CA9_DBG_CTR	-	-	-	-	CWG[3]	CWG[2]	CWG[1]	CWG[0]
		ERG[3]	ERG[2]	ERG[1]	ERG[0]	DminLine[3]	DminLine[2]	DminLine[1]	DminLine[0]
		-	-	-	-	-	-	-	-
		-	-	-	-	IminLine[3]	IminLine[2]	IminLine[1]	IminLine[0]
	CA9_DBG_TLBTR	-	-	-	-	-	-	-	-
		ILSize[7]	ILSize[6]	ILSize[5]	ILSize[4]	ILSize[3]	ILSize[2]	ILSize[1]	ILSize[0]
		DLSize[7]	DLSize[6]	DLSize[5]	DLSize[4]	DLSize[3]	DLSize[2]	DLSize[1]	DLSize[0]
		-	-	-	-	-	-	TLB_size	uU
	CA9_DBG_MPIDR	-	U	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	ClusterID[3]	ClusterID[2]	ClusterID[1]	ClusterID[0]
		-	-	-	-	-	-	CPUID[1]	CPUID[0]
	CA9_DBG_REVIDR	ID[31]	ID[30]	ID[29]	ID[28]	ID[27]	ID[26]	ID[25]	ID[24]
		ID[23]	ID[22]	ID[21]	ID[20]	ID[19]	ID[18]	ID[17]	ID[16]
		ID[15]	ID[14]	ID[13]	ID[12]	ID[11]	ID[10]	ID[9]	ID[8]
		ID[7]	ID[6]	ID[5]	ID[4]	ID[3]	ID[2]	ID[1]	ID[0]
	CA9_DBG_ID_PFR0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		State3[3]	State3[2]	State3[1]	State3[0]	State2[3]	State2[2]	State2[1]	State2[0]
		State1[3]	State1[2]	State1[1]	State1[0]	State0[3]	State0[2]	State0[1]	State0[0]
	CA9_DBG_ID_PFR1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	Mprofile[3]	Mprofile[2]	Mprofile[1]	Mprofile[0]
		Security[3]	Security[2]	Security[1]	Security[0]	model[3]	model[2]	model[1]	model[0]
	CA9_DBG_ID_DFR0	-	-	-	-	-	-	-	-
		Debug Model, M profile[3]	Debug Model, M profile[2]	Debug Model, M profile[1]	Debug Model, M profile[0]	Memory-mapped trace model[3]	Memory-mapped trace model[2]	Memory-mapped trace model[1]	Memory-mapped trace model[0]
		Coprocessor trace model[3]	Coprocessor trace model[2]	Coprocessor trace model[1]	Coprocessor trace model[0]	Memory-mapped debug model[3]	Memory-mapped debug model[2]	Memory-mapped debug model[1]	Memory-mapped debug model[0]
		Coprocessor Secure debug model[3]	Coprocessor Secure debug model[2]	Coprocessor Secure debug model[1]	Coprocessor Secure debug model[0]	Coprocessor debug model[3]	Coprocessor debug model[2]	Coprocessor debug model[1]	Coprocessor debug model[0]
	CA9_DBG_ID_ MMFR0	Innermost shareability[3]	Innermost shareability2]	Innermost shareability[1]	Innermost shareability[0]	FCSE support[3]	FCSE support[2]	FCSE support[1]	FCSE support[0]
		Auxiliary registers[3]	Auxiliary registers[2]	Auxiliary registers[1]	Auxiliary registers[0]	TCM support[3]	TCM support[2]	TCM support[1]	TCM support[0]
		Share ability levels[3]	Share ability levels[2]	Share ability levels[1]	Share ability levels[0]	Outermost shareability[3]	Outermost shareability[2]	Outermost shareability[1]	Outermost shareability[0]
		PMSA support[3]	PMSA support[2]	PMSA support[1]	PMSA support[0]	VMSA support[3]	VMSA support[2]	VMSA support[1]	VMSA support[0]
	CA9_DBG_ID_ MMFR1	Branch Predictor[3]	Branch Predictor[2]	Branch Predictor[1]	Branch Predictor[0]	L1 cache Testand Clean[3]	L1 cache Testand Clean[2]	L1 cache Testand Clean[1]	L1 cache Testand Clean[0]
		L1 unified cache[3]	L1 unified cache[2]	L1 unified cache1]	L1 unified cache[0]	L1 Harvard cache[3]	L1 Harvard cache[2]	L1 Harvard cache[1]	L1 Harvard cache[0]
		L1 unified cache s/w[3]	L1 unified cache s/w[2]	L1 unified cache s/w[1]	L1 unified cache s/w[0]	L1 Harvard cache s/w[3]	L1 Harvard cache s/w[2]	L1 Harvard cache s/w[1]	L1 Harvard cache s/w[0]
		L1 unified cache VA[3]	L1 unified cache VA[2]	L1 unified cache VA[1]	L1 unified cache VA[0]	L1 Harvard cache VA[3]	L1 Harvard cache VA[2]	L1 Harvard cache VA[1]	L1 Harvard cache VA[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Debugger interface	CA9_DBG_ID_MMFR2	HW access flag[3]	HW access flag[2]	HW access flag[1]	HW access flag[0]	WFI stall[3]	WFI stall[2]	WFI stall[1]	WFI stall[0]
		Mem barrier[3]	Mem barrier[2]	Mem barrier[1]	Mem barrier[0]	Unified TLB[3]	Unified TLB[2]	Unified TLB[1]	Unified TLB[0]
		Harvard TLB[3]	Harvard TLB[2]	Harvard TLB[1]	Harvard TLB[0]	L1 Harvard range[3]	L1 Harvard range[2]	L1 Harvard range[1]	L1 Harvard range[0]
		L1 Harvard bg prefetch[3]	L1 Harvard bg prefetch[2]	L1 Harvard bg prefetch[1]	L1 Harvard bg prefetch[0]	L1 Harvard fg prefetch[3]	L1 Harvard fg prefetch[2]	L1 Harvard fg prefetch[1]	L1 Harvard fg prefetch[0]
	CA9_DBG_ID_MMFR3	Supersection support[3]	Supersection support[2]	Supersection support[1]	Supersection support[0]	-	-	-	-
		Coherent walk[3]	Coherent walk[2]	Coherent walk[1]	Coherent walk[0]	-	-	-	-
		Maintenance broadcast[3]	Maintenance broadcast[2]	Maintenance broadcast[1]	Maintenance broadcast[0]	BP maintain[3]	BP maintain[2]	BP maintain[1]	BP maintain[0]
		Cache maintenance s/w[3]	Cache maintenance s/w[2]	Cache maintenance s/w[1]	Cache maintenance s/w[0]	Cache maintenance MVA[3]	Cache maintenance MVA[2]	Cache maintenance MVA[1]	Cache maintenance MVA[0]
	CA9_DBG_ID_ISAR0	-	-	-	-	Divide_instrs[3]	Divide_instrs[2]	Divide_instrs[1]	Divide_instrs[0]
		Debug_instrs[3]	Debug_instrs[2]	Debug_instrs[1]	Debug_instrs[0]	Coproc_instrs[3]	Coproc_instrs[2]	Coproc_instrs[1]	Coproc_instrs[0]
		CmpBranch_instrs[3]	CmpBranch_instrs[2]	CmpBranch_instrs[1]	CmpBranch_instrs[0]	Bitfield_instrs[3]	Bitfield_instrs[2]	Bitfield_instrs[1]	Bitfield_instrs[0]
		BitCount_instrs[3]	BitCount_instrs[2]	BitCount_instrs[1]	BitCount_instrs[0]	Swap_instrs[3]	Swap_instrs[2]	Swap_instrs[1]	Swap_instrs[0]
	CA9_DBG_ID_ISAR1	Jazelle_instrs[3]	Jazelle_instrs[2]	Jazelle_instrs[1]	Jazelle_instrs[0]	Interwork_instrs[3]	Interwork_instrs[2]	Interwork_instrs[1]	Interwork_instrs[0]
		Immediate_instrs[3]	Immediate_instrs[2]	Immediate_instrs[1]	Immediate_instrs[0]	IfThen_instrs[3]	IfThen_instrs[2]	IfThen_instrs[1]	IfThen_instrs[0]
		Extend_instrs[3]	Extend_instrs[2]	Extend_instrs[1]	Extend_instrs[0]	Except_AR_instrs[3]	Except_AR_instrs[2]	Except_AR_instrs[1]	Except_AR_instrs[0]
		Except_instrs[3]	Except_instrs[2]	Except_instrs[1]	Except_instrs[0]	Endian_instrs[3]	Endian_instrs[2]	Endian_instrs[1]	Endian_instrs[0]
	CA9_DBG_ID_ISAR2	Reversal_instrs[3]	Reversal_instrs[2]	Reversal_instrs[1]	Reversal_instrs[0]	PSR_AR_instrs[3]	PSR_AR_instrs[2]	PSR_AR_instrs[1]	PSR_AR_instrs[0]
		MultU_instrs[3]	MultU_instrs[2]	MultU_instrs[1]	MultU_instrs[0]	Mult_instrs[3]	Mult_instrs[2]	Mult_instrs[1]	Mult_instrs[0]
		Mult_instrs[3]	Mult_instrs[2]	Mult_instrs[1]	Mult_instrs[0]	MultiAccessInt_instrs[3]	MultiAccessInt_instrs[2]	MultiAccessInt_instrs[1]	MultiAccessInt_instrs[0]
		MemHint_instrs[3]	MemHint_instrs[2]	MemHint_instrs[1]	MemHint_instrs[0]	LoadStore_instrs[3]	LoadStore_instrs[2]	LoadStore_instrs[1]	LoadStore_instrs[0]
	CA9_DBG_ID_ISAR3	ThumbEE_extn_instrs[3]	ThumbEE_extn_instrs[2]	ThumbEE_extn_instrs[1]	ThumbEE_extn_instrs[0]	TrueNOP_instrs[3]	TrueNOP_instrs[2]	TrueNOP_instrs[1]	TrueNOP_instrs[0]
		ThumbCopy_instrs[3]	ThumbCopy_instrs[2]	ThumbCopy_instrs[1]	ThumbCopy_instrs[0]	TabBranch_instrs[3]	TabBranch_instrs[2]	TabBranch_instrs[1]	TabBranch_instrs[0]
		SynchPrim_instrs[3]	SynchPrim_instrs[2]	SynchPrim_instrs[1]	SynchPrim_instrs[0]	SVC_instrs[3]	SVC_instrs[2]	SVC_instrs[1]	SVC_instrs[0]
		SIMD_instrs[3]	SIMD_instrs[2]	SIMD_instrs[1]	SIMD_instrs[0]	Saturate_instrs[3]	Saturate_instrs[2]	Saturate_instrs[1]	Saturate_instrs[0]
	CA9_DBG_ID_ISAR4	SWP_frac[3]	SWP_frac[2]	SWP_frac[1]	SWP_frac[0]	PSR_M_instrs[3]	PSR_M_instrs[2]	PSR_M_instrs[1]	PSR_M_instrs[0]
		SynchPrim_instrs_frac[3]	SynchPrim_instrs_frac[2]	SynchPrim_instrs_frac[1]	SynchPrim_instrs_frac[0]	Barrier_instrs[3]	Barrier_instrs[2]	Barrier_instrs[1]	Barrier_instrs[0]
		SMC_instrs[3]	SMC_instrs[2]	SMC_instrs[1]	SMC_instrs[0]	Writeback_instrs[3]	Writeback_instrs[2]	Writeback_instrs[1]	Writeback_instrs[0]
		WithShifts_instrs[3]	WithShifts_instrs[2]	WithShifts_instrs[1]	WithShifts_instrs[0]	Unpriv_instrs[3]	Unpriv_instrs[2]	Unpriv_instrs[1]	Unpriv_instrs[0]
	CA9_DBG_CLAIMSET	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		CLAIMSET[7]	CLAIMSET[6]	CLAIMSET[5]	CLAIMSET[4]	CLAIMSET[3]	CLAIMSET[2]	CLAIMSET[1]	CLAIMSET[0]
	CA9_DBG_CLAIMCLR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		CLAIMCLR[7]	CLAIMCLR[6]	CLAIMCLR[5]	CLAIMCLR[4]	CLAIMCLR[3]	CLAIMCLR[2]	CLAIMCLR[1]	CLAIMCLR[0]
	CA9_DBG_LAR	ACCESS_W[31]	ACCESS_W[30]	ACCESS_W[29]	ACCESS_W[28]	ACCESS_W[27]	ACCESS_W[26]	ACCESS_W[25]	ACCESS_W[24]
		ACCESS_W[23]	ACCESS_W[22]	ACCESS_W[21]	ACCESS_W[20]	ACCESS_W[19]	ACCESS_W[18]	ACCESS_W[17]	ACCESS_W[16]
		ACCESS_W[15]	ACCESS_W[14]	ACCESS_W[13]	ACCESS_W[12]	ACCESS_W[11]	ACCESS_W[10]	ACCESS_W[9]	ACCESS_W[8]
		ACCESS_W[7]	ACCESS_W[6]	ACCESS_W[5]	ACCESS_W[4]	ACCESS_W[3]	ACCESS_W[2]	ACCESS_W[1]	ACCESS_W[0]
	CA9_DBG_LSR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	LOCKTYPE	LOCKGRANT	LOCKEXIST
	CA9_DBG_AUTHSTATUS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		SNID[1]	SNID[0]	SID[1]	SID[0]	NSNID[1]	NSNID[0]	NSID[1]	NSID[0]
	CA9_DBG_DEVID	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	PCsample[3]	PCsample[2]	PCsample[1]	PCsample[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Debugger interface	CA9_DBG_ DEVTYPE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Sub_type[3]	Sub_type[2]	Sub_type[1]	Sub_type[0]	Main_class[3]	Main_class[2]	Main_class[1]	Main_class[0]
	CA9_DBG_ PERIPHID4	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		4KB_count[3]	4KB_count[2]	4KB_count[1]	4KB_count[0]	JEP106_code[3]	JEP106_code[2]	JEP106_code[1]	JEP106_code[0]
	CA9_DBG_ PERIPHID0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Part_Number[7]	Part_Number[6]	Part_Number[5]	Part_Number[4]	Part_Number[3]	Part_Number[2]	Part_Number[1]	Part_Number[0]
	CA9_DBG_ PERIPHID1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		JEP106_id[3]	JEP106_id[2]	JEP106_id[1]	JEP106_id[0]	Part_Number [11]	Part_Number [10]	Part_Number[9]	Part_Number[8]
	CA9_DBG_ PERIPHID2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Revision[3]	Revision[2]	Revision[1]	Revision[0]	JEDEC	JEP106_id[6]	JEP106_id[5]	JEP106_id[4]
	CA9_DBG_ PERIPHID3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		RevAnd[3]	RevAnd[2]	RevAnd[1]	RevAnd[0]	CUSTOM[3]	CUSTOM[2]	CUSTOM[1]	CUSTOM[0]
	CA9_DBG_ COMPID0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	CA9_DBG_ COMPID1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Class[3]	Class[2]	Class[1]	Class[0]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	CA9_DBG_ COMPID2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	CA9_DBG_ COMPID3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	CA9_PMU_ PMXEVCNTR0	PMNX[31]	PMNX[30]	PMNX[29]	PMNX[28]	PMNX[27]	PMNX[26]	PMNX[25]	PMNX[24]
		PMNX[23]	PMNX[22]	PMNX[21]	PMNX[20]	PMNX[19]	PMNX[18]	PMNX[17]	PMNX[16]
		PMNX[15]	PMNX[14]	PMNX[13]	PMNX[12]	PMNX[11]	PMNX[10]	PMNX[9]	PMNX[8]
		PMNX[7]	PMNX[6]	PMNX[5]	PMNX[4]	PMNX[3]	PMNX[2]	PMNX[1]	PMNX[0]
	CA9_PMU_ PMXEVCNTR1	PMNX[31]	PMNX[30]	PMNX[29]	PMNX[28]	PMNX[27]	PMNX[26]	PMNX[25]	PMNX[24]
		PMNX[23]	PMNX[22]	PMNX[21]	PMNX[20]	PMNX[19]	PMNX[18]	PMNX[17]	PMNX[16]
		PMNX[15]	PMNX[14]	PMNX[13]	PMNX[12]	PMNX[11]	PMNX[10]	PMNX[9]	PMNX[8]
		PMNX[7]	PMNX[6]	PMNX[5]	PMNX[4]	PMNX[3]	PMNX[2]	PMNX[1]	PMNX[0]
	CA9_PMU_ PMXEVCNTR2	PMNX[31]	PMNX[30]	PMNX[29]	PMNX[28]	PMNX[27]	PMNX[26]	PMNX[25]	PMNX[24]
		PMNX[23]	PMNX[22]	PMNX[21]	PMNX[20]	PMNX[19]	PMNX[18]	PMNX[17]	PMNX[16]
		PMNX[15]	PMNX[14]	PMNX[13]	PMNX[12]	PMNX[11]	PMNX[10]	PMNX[9]	PMNX[8]
		PMNX[7]	PMNX[6]	PMNX[5]	PMNX[4]	PMNX[3]	PMNX[2]	PMNX[1]	PMNX[0]
	CA9_PMU_ PMXEVCNTR3	PMNX[31]	PMNX[30]	PMNX[29]	PMNX[28]	PMNX[27]	PMNX[26]	PMNX[25]	PMNX[24]
		PMNX[23]	PMNX[22]	PMNX[21]	PMNX[20]	PMNX[19]	PMNX[18]	PMNX[17]	PMNX[16]
		PMNX[15]	PMNX[14]	PMNX[13]	PMNX[12]	PMNX[11]	PMNX[10]	PMNX[9]	PMNX[8]
		PMNX[7]	PMNX[6]	PMNX[5]	PMNX[4]	PMNX[3]	PMNX[2]	PMNX[1]	PMNX[0]
	CA9_PMU_ PMXEVCNTR4	PMNX[31]	PMNX[30]	PMNX[29]	PMNX[28]	PMNX[27]	PMNX[26]	PMNX[25]	PMNX[24]
		PMNX[23]	PMNX[22]	PMNX[21]	PMNX[20]	PMNX[19]	PMNX[18]	PMNX[17]	PMNX[16]
		PMNX[15]	PMNX[14]	PMNX[13]	PMNX[12]	PMNX[11]	PMNX[10]	PMNX[9]	PMNX[8]
		PMNX[7]	PMNX[6]	PMNX[5]	PMNX[4]	PMNX[3]	PMNX[2]	PMNX[1]	PMNX[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Debugger interface	CA9_PMU_ PMXEVCNTR5	PMNX[31]	PMNX[30]	PMNX[29]	PMNX[28]	PMNX[27]	PMNX[26]	PMNX[25]	PMNX[24]
		PMNX[23]	PMNX[22]	PMNX[21]	PMNX[20]	PMNX[19]	PMNX[18]	PMNX[17]	PMNX[16]
		PMNX[15]	PMNX[14]	PMNX[13]	PMNX[12]	PMNX[11]	PMNX[10]	PMNX[9]	PMNX[8]
		PMNX[7]	PMNX[6]	PMNX[5]	PMNX[4]	PMNX[3]	PMNX[2]	PMNX[1]	PMNX[0]
	CA9_PMU_ PMCCNTR	CCNT[31]	CCNT[30]	CCNT[29]	CCNT[28]	CCNT[27]	CCNT[26]	CCNT[25]	CCNT[24]
		CCNT[23]	CCNT[22]	CCNT[21]	CCNT[20]	CCNT[19]	CCNT[18]	CCNT[17]	CCNT[16]
		CCNT[15]	CCNT[14]	CCNT[13]	CCNT[12]	CCNT[11]	CCNT[10]	CCNT[9]	CCNT[8]
		CCNT[7]	CCNT[6]	CCNT[5]	CCNT[4]	CCNT[3]	CCNT[2]	CCNT[1]	CCNT[0]
	CA9_PMU_ PMXEVTYPER0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	SEL[4]	SEL[3]	SEL[2]	SEL[1]	SEL[0]
	CA9_PMU_ PMXEVTYPER1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	SEL[4]	SEL[3]	SEL[2]	SEL[1]	SEL[0]
	CA9_PMU_ PMXEVTYPER2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	SEL[4]	SEL[3]	SEL[2]	SEL[1]	SEL[0]
	CA9_PMU_ PMXEVTYPER3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	SEL[4]	SEL[3]	SEL[2]	SEL[1]	SEL[0]
	CA9_PMU_ PMXEVTYPER4	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	SEL[4]	SEL[3]	SEL[2]	SEL[1]	SEL[0]
	CA9_PMU_ PMXEVTYPER5	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	SEL[4]	SEL[3]	SEL[2]	SEL[1]	SEL[0]
	CA9_PMU_ PMCNTENSET	C	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	P5	P4	P3	P2	P1	P0
	CA9_PMU_ PMCNTENCLR	C	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	P5	P4	P3	P2	P1	P0
	CA9_PMU_ PMINTENSET	C	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	P5	P4	P3	P2	P1	P0
	CA9_PMU_ PMINTENCLR	C	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	P5	P4	P3	P2	P1	P0
	CA9_PMU_ PMINTENCLR	C	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	P5	P4	P3	P2	P1	P0
	CA9_PMU_PMOVSR	C	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	P5	P4	P3	P2	P1	P0
	CA9_PMU_ PMSWINC	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	P5	P4	P3	P2	P1	P0

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Debugger interface	CA9_PMU_PMCR	IMP[7]	IMP[6]	IMP[5]	IMP[4]	IMP[3]	IMP[2]	IMP[1]	IMP[0]
		IDCODE[7]	IDCODE[6]	IDCODE[5]	IDCODE[4]	IDCODE[3]	IDCODE[2]	IDCODE[1]	IDCODE[0]
		N[4]	N[3]	N[2]	N[1]	N[0]	-	-	-
		-	-	DP	X	D	C	P	E
	CA9_PMU_ CLAIMSET	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		CLAIMSET[7]	CLAIMSET[6]	CLAIMSET[5]	CLAIMSET[4]	CLAIMSET[3]	CLAIMSET[2]	CLAIMSET[1]	CLAIMSET[0]
	CA9_PMU_ CLAIMCLR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		CLAIMCLR[7]	CLAIMCLR[6]	CLAIMCLR[5]	CLAIMCLR[4]	CLAIMCLR[3]	CLAIMCLR[2]	CLAIMCLR[1]	CLAIMCLR[0]
	CA9_PMU_LAR	ACCESS_W[31]	ACCESS_W[30]	ACCESS_W[29]	ACCESS_W[28]	ACCESS_W[27]	ACCESS_W[26]	ACCESS_W[25]	ACCESS_W[24]
		ACCESS_W[23]	ACCESS_W[22]	ACCESS_W[21]	ACCESS_W[20]	ACCESS_W[19]	ACCESS_W[18]	ACCESS_W[17]	ACCESS_W[16]
		ACCESS_W[15]	ACCESS_W[14]	ACCESS_W[13]	ACCESS_W[12]	ACCESS_W[11]	ACCESS_W[10]	ACCESS_W[9]	ACCESS_W[8]
		ACCESS_W[7]	ACCESS_W[6]	ACCESS_W[5]	ACCESS_W[4]	ACCESS_W[3]	ACCESS_W[2]	ACCESS_W[1]	ACCESS_W[0]
	CA9_PMU_LSR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	LOCKTYPE	LOCKGRANT	LOCKEXIST
	CA9_PMU_ AUTHSTATUS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		SNID[1]	SNID[0]	SID[1]	SID[0]	NSNID[1]	NSNID[0]	NSID[1]	NSID[0]
	CA9_PMU_DEVID	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ID[7]	ID[6]	ID[5]	ID[4]	ID[3]	ID[2]	ID[1]	ID[0]
	CA9_PMU_ DEVTYPE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Sub_type[3]	Sub_type[2]	Sub_type[1]	Sub_type[0]	Main_class[3]	Main_class[2]	Main_class[1]	Main_class[0]
	CA9_PMU_ PERIPHID4	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		4KB_count[3]	4KB_count[2]	4KB_count[1]	4KB_count[0]	JEP106_code[3]	JEP106_code[2]	JEP106_code[1]	JEP106_code[0]
	CA9_PMU_ PERIPHID0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Part_Number[7]	Part_Number[6]	Part_Number[5]	Part_Number[4]	Part_Number[3]	Part_Number[2]	Part_Number[1]	Part_Number[0]
	CA9_PMU_ PERIPHID1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		JEP106_id[3]	JEP106_id[2]	JEP106_id[1]	JEP106_id[0]	Part_Number[11]	Part_Number[10]	Part_Number[9]	Part_Number[8]
	CA9_PMU_ PERIPHID2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Revision[3]	Revision[2]	Revision[1]	Revision[0]	JEDEC	JEP106_id[6]	JEP106_id[5]	JEP106_id[4]
	CA9_PMU_ PERIPHID3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		RevAnd[3]	RevAnd[2]	RevAnd[1]	RevAnd[0]	CUSTOM[3]	CUSTOM[2]	CUSTOM[1]	CUSTOM[0]
	CA9_PMU_ COMPID0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	CA9_PMU_ COMPID1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Class[3]	Class[2]	Class[1]	Class[0]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Debugger interface	CA9_PMU_ COMPID2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	CA9_PMU_ COMPID3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	CA9_CTI_ CTICONTROL	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	GLBEN
	CA9_CTI_ CTIINTACK	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		INTACK[7]	INTACK[6]	INTACK[5]	INTACK[4]	INTACK[3]	INTACK[2]	INTACK[1]	INTACK[0]
	CA9_CTI_ CTIAPPSET	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	APPSET[3]	APPSET[2]	APPSET[1]	APPSET[0]
	CA9_CTI_ CTIAPPCLEAR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	APPCLEAR[3]	APPCLEAR[2]	APPCLEAR[1]	APPCLEAR[0]
	CA9_CTI_ CTIAPPPULSE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	APPULSE[3]	APPULSE[2]	APPULSE[1]	APPULSE[0]
	CA9_CTI_CTIINEN0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGINEN[3]	TRIGINEN[2]	TRIGINEN[1]	TRIGINEN[0]
	CA9_CTI_CTIINEN1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGINEN[3]	TRIGINEN[2]	TRIGINEN[1]	TRIGINEN[0]
	CA9_CTI_CTIINEN2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGINEN[3]	TRIGINEN[2]	TRIGINEN[1]	TRIGINEN[0]
	CA9_CTI_CTIINEN3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGINEN[3]	TRIGINEN[2]	TRIGINEN[1]	TRIGINEN[0]
	CA9_CTI_CTIINEN4	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGINEN[3]	TRIGINEN[2]	TRIGINEN[1]	TRIGINEN[0]
	CA9_CTI_CTIINEN5	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGINEN[3]	TRIGINEN[2]	TRIGINEN[1]	TRIGINEN[0]
	CA9_CTI_CTIINEN6	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGINEN[3]	TRIGINEN[2]	TRIGINEN[1]	TRIGINEN[0]
	CA9_CTI_CTIINEN7	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGINEN[3]	TRIGINEN[2]	TRIGINEN[1]	TRIGINEN[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Debugger interface	CA9_CTI_CTIOUTEN0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGOUTEN[3]	TRIGOUTEN[2]	TRIGOUTEN[1]	TRIGOUTEN[0]
	CA9_CTI_CTIOUTEN1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGOUTEN[3]	TRIGOUTEN[2]	TRIGOUTEN[1]	TRIGOUTEN[0]
	CA9_CTI_CTIOUTEN2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGOUTEN[3]	TRIGOUTEN[2]	TRIGOUTEN[1]	TRIGOUTEN[0]
	CA9_CTI_CTIOUTEN3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGOUTEN[3]	TRIGOUTEN[2]	TRIGOUTEN[1]	TRIGOUTEN[0]
	CA9_CTI_CTIOUTEN4	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGOUTEN[3]	TRIGOUTEN[2]	TRIGOUTEN[1]	TRIGOUTEN[0]
	CA9_CTI_CTIOUTEN5	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGOUTEN[3]	TRIGOUTEN[2]	TRIGOUTEN[1]	TRIGOUTEN[0]
	CA9_CTI_CTIOUTEN6	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGOUTEN[3]	TRIGOUTEN[2]	TRIGOUTEN[1]	TRIGOUTEN[0]
	CA9_CTI_CTIOUTEN7	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	TRIGOUTEN[3]	TRIGOUTEN[2]	TRIGOUTEN[1]	TRIGOUTEN[0]
	CA9_CTI_CTITRIGINSTATUS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		TRIGINSTATUS[7]	TRIGINSTATUS[6]	TRIGINSTATUS[5]	TRIGINSTATUS[4]	TRIGINSTATUS[3]	TRIGINSTATUS[2]	TRIGINSTATUS[1]	TRIGINSTATUS[0]
	CA9_CTI_CTITRIGOUTSTATUS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		TRIGOUTSTATUS[7]	TRIGOUTSTATUS[6]	TRIGOUTSTATUS[5]	TRIGOUTSTATUS[4]	TRIGOUTSTATUS[3]	TRIGOUTSTATUS[2]	TRIGOUTSTATUS[1]	TRIGOUTSTATUS[0]
	CA9_CTI_CTICHINSTATUS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CTCHINSTATUS[3]	CTCHINSTATUS[2]	CTCHINSTATUS[1]	CTCHINSTATUS[0]
	CA9_CTI_CTICHOUTSTATUS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CTCHOUTSTATUS[3]	CTCHOUTSTATUS[2]	CTCHOUTSTATUS[1]	CTCHOUTSTATUS[0]
	CA9_CTI_CTIGATE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CTIGATEEN3	CTIGATEEN2	CTIGATEEN1	CTIGATEEN0
	CA9_CTI_ASICCTL	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	ASICCTL[3]	ASICCTL[2]	ASICCTL[1]	ASICCTL[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Debugger interface	CA9_CTI_CLAIMSET	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CLAIMSET[3]	CLAIMSET[2]	CLAIMSET[1]	CLAIMSET[0]
	CA9_CTI_CLAIMCLR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	CLAIMCLR[3]	CLAIMCLR[2]	CLAIMCLR[1]	CLAIMCLR[0]
	CA9_CTI_LAR	ACCESS_W[31]	ACCESS_W[30]	ACCESS_W[29]	ACCESS_W[28]	ACCESS_W[27]	ACCESS_W[26]	ACCESS_W[25]	ACCESS_W[24]
		ACCESS_W[23]	ACCESS_W[22]	ACCESS_W[21]	ACCESS_W[20]	ACCESS_W[19]	ACCESS_W[18]	ACCESS_W[17]	ACCESS_W[16]
		ACCESS_W[15]	ACCESS_W[14]	ACCESS_W[13]	ACCESS_W[12]	ACCESS_W[11]	ACCESS_W[10]	ACCESS_W[9]	ACCESS_W[8]
		ACCESS_W[7]	ACCESS_W[6]	ACCESS_W[5]	ACCESS_W[4]	ACCESS_W[3]	ACCESS_W[2]	ACCESS_W[1]	ACCESS_W[0]
	CA9_CTI_LSR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	LOCKTYPE	LOCKGRANT	LOCKEXIST
	CA9_CTI_AUTHSTATUS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		SNID[1]	SNID[0]	SID[1]	SID[0]	NSNID[1]	NSNID[0]	NSID[1]	NSID[0]
	CA9_CTI_DEVID	-	-	-	-	-	-	-	-
		-	-	-	-	CHANWIDTH[3]	CHANWIDTH[2]	CHANWIDTH[1]	CHANWIDTH[0]
		TRIGWIDTH[7]	TRIGWIDTH[6]	TRIGWIDTH[5]	TRIGWIDTH[4]	TRIGWIDTH[3]	TRIGWIDTH[2]	TRIGWIDTH[1]	TRIGWIDTH[0]
		-	-	-	-	EXTMUXNUM[3]	EXTMUXNUM[2]	EXTMUXNUM[1]	EXTMUXNUM[0]
	CA9_CTI_DEVTYPE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Sub_type[3]	Sub_type[2]	Sub_type[1]	Sub_type[0]	Major_type[3]	Major_type[2]	Major_type[1]	Major_type[0]
	CA9_CTI_PERIPHID4	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		4KB_count[3]	4KB_count[2]	4KB_count[1]	4KB_count[0]	JEP106_code[3]	JEP106_code[2]	JEP106_code[1]	JEP106_code[0]
	CA9_CTI_PERIPHID0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Part_Number[7]	Part_Number[6]	Part_Number[5]	Part_Number[4]	Part_Number[3]	Part_Number[2]	Part_Number[1]	Part_Number[0]
	CA9_CTI_PERIPHID1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		JEP106_id[3]	JEP106_id[2]	JEP106_id[1]	JEP106_id[0]	Part_Number [11]	Part_Number [10]	Part_Number[9]	Part_Number[8]
	CA9_CTI_PERIPHID2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Revision[3]	Revision[2]	Revision[1]	Revision[0]	JEDEC	JEP106_id[6]	JEP106_id[5]	JEP106_id[4]
	CA9_CTI_PERIPHID3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		RevAnd[3]	RevAnd[2]	RevAnd[1]	RevAnd[0]	CUSTOM[3]	CUSTOM[2]	CUSTOM[1]	CUSTOM[0]
	CA9_CTI_COMPID0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	CA9_CTI_COMPID1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Class[3]	Class[2]	Class[1]	Class[0]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	CA9_CTI_COMPID2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Debugger interface	CA9_CTI_COMPID3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	PTM_A9_ETMCR	-	-	Return stack enable	Timestamp enable	Processor select[2]	Processor select [1]	Processor select [0]	-
		-	-	-	-	-	-	-	-
		Context IDsize[1]	Context IDsize[0]	-	CycleAccurate	-	ProgBit	Debug request control	Branch Broadcast
		Stall processor	-	-	-	-	-	-	PowerDown
	PTM_A9_ETMCCR	ID Register present	-	-	-	Software access support	Trace stop/start block present	Number of Context ID comparators[1]	Number of Context ID comparators[0]
		FIFOFULL logic	Number of external outputs[2]	Number of external outputs[1]	Number of external outputs[0]	Number of external inputs[2]	Number of external inputs[1]	Number of external inputs[0]	Sequencer
		Numberofcounters[2]	Numberofcounters[1]	Numberofcounters[0]	-	-	-	-	-
		-	-	-	-	Number of pairs of address comparators[3]	Number of pairs of address comparators[2]	Number of pairs of address comparators[1]	Number of pairs of address comparators[0]
	PTM_A9_ETMTRIGGER	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Trigger event[15]	Trigger Event[14]	Trigger Event[13]	Trigger Event[12]	Trigger Event[11]	Trigger Event[10]	Trigger Event[9]	Trigger Event[8]
		Trigger Event[7]	Trigger Event[6]	Trigger Event[5]	Trigger Event[4]	Trigger Event[3]	Trigger Event[2]	Trigger Event[1]	Trigger Event[0]
	PTM_A9_ETMSR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	Trigger flag	Trace start/stop resource status	ProgBit value	Untraced overflow flag
	PTM_A9_ETMSCR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	MAXCORES[2]	MAXCORES[1]	MAXCORES[0]	-	-	-	FIFOFULL supported
		-	-	-	-	-	-	-	-
	PTM_A9_ETMTSSCR	-	-	-	-	-	-	-	-
		Stop addresses[7]	Stop addresses[6]	Stop addresses[5]	Stop addresses[4]	Stop addresses[3]	Stop addresses[2]	Stop addresses[1]	Stop addresses[0]
		-	-	-	-	-	-	-	-
		Start addresses[7]	Start addresses[6]	Start addresses[5]	Start addresses[4]	Start addresses[3]	Start addresses[2]	Start addresses[1]	Start addresses[0]
	PTM_A9_ETMTEEVR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Trace Enable event[15]	Trace Enable event[14]	Trace Enable event[13]	Trace Enable event[12]	Trace Enable event[11]	Trace Enable event[10]	Trace Enable event[9]	Trace Enable event[8]
		Trace Enable event[7]	Trace Enable event[6]	Trace Enable event[5]	Trace Enable event[4]	Trace Enable event[3]	Trace Enable event[2]	Trace Enable event[1]	Trace Enable event[0]
	PTM_A9_ETMTECR1	-	-	-	-	-	-	Trace control enable	Exclude/include flag
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
						Addresscom parators[3]	Addresscom parators[2]	Addresscom parators[1]	Addresscom parators[0]
	PTM_A9_ETMACVR1	Address[31]	Address[30]	Address[29]	Address[28]	Address[27]	Address[26]	Address[25]	Address[24]
		Address[23]	Address[22]	Address[21]	Address[20]	Address[19]	Address[18]	Address[17]	Address[16]
		Address[15]	Address[14]	Address[13]	Address[12]	Address[11]	Address[10]	Address[9]	Address[8]
		Address[7]	Address[6]	Address[5]	Address[4]	Address[3]	Address[2]	Address[1]	Address[0]
	PTM_A9_ETMACVR2	Address[31]	Address[30]	Address[29]	Address[28]	Address[27]	Address[26]	Address[25]	Address[24]
		Address[23]	Address[22]	Address[21]	Address[20]	Address[19]	Address[18]	Address[17]	Address[16]
		Address[15]	Address[14]	Address[13]	Address[12]	Address[11]	Address[10]	Address[9]	Address[8]
		Address[7]	Address[6]	Address[5]	Address[4]	Address[3]	Address[2]	Address[1]	Address[0]
	PTM_A9_ETMACVR3	Address[31]	Address[30]	Address[29]	Address[28]	Address[27]	Address[26]	Address[25]	Address[24]
		Address[23]	Address[22]	Address[21]	Address[20]	Address[19]	Address[18]	Address[17]	Address[16]
		Address[15]	Address[14]	Address[13]	Address[12]	Address[11]	Address[10]	Address[9]	Address[8]
		Address[7]	Address[6]	Address[5]	Address[4]	Address[3]	Address[2]	Address[1]	Address[0]
	PTM_A9_ETMACVR4	Address[31]	Address[30]	Address[29]	Address[28]	Address[27]	Address[26]	Address[25]	Address[24]
		Address[23]	Address[22]	Address[21]	Address[20]	Address[19]	Address[18]	Address[17]	Address[16]
		Address[15]	Address[14]	Address[13]	Address[12]	Address[11]	Address[10]	Address[9]	Address[8]
		Address[7]	Address[6]	Address[5]	Address[4]	Address[3]	Address[2]	Address[1]	Address[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Debugger interface	PTM_A9_ ETMACVR5	Address[31]	Address[30]	Address[29]	Address[28]	Address[27]	Address[26]	Address[25]	Address[24]
		Address[23]	Address[22]	Address[21]	Address[20]	Address[19]	Address[18]	Address[17]	Address[16]
		Address[15]	Address[14]	Address[13]	Address[12]	Address[11]	Address[10]	Address[9]	Address[8]
		Address[7]	Address[6]	Address[5]	Address[4]	Address[3]	Address[2]	Address[1]	Address[0]
	PTM_A9_ ETMACVR6	Address[31]	Address[30]	Address[29]	Address[28]	Address[27]	Address[26]	Address[25]	Address[24]
		Address[23]	Address[22]	Address[21]	Address[20]	Address[19]	Address[18]	Address[17]	Address[16]
		Address[15]	Address[14]	Address[13]	Address[12]	Address[11]	Address[10]	Address[9]	Address[8]
		Address[7]	Address[6]	Address[5]	Address[4]	Address[3]	Address[2]	Address[1]	Address[0]
	PTM_A9_ ETMACVR7	Address[31]	Address[30]	Address[29]	Address[28]	Address[27]	Address[26]	Address[25]	Address[24]
		Address[23]	Address[22]	Address[21]	Address[20]	Address[19]	Address[18]	Address[17]	Address[16]
		Address[15]	Address[14]	Address[13]	Address[12]	Address[11]	Address[10]	Address[9]	Address[8]
		Address[7]	Address[6]	Address[5]	Address[4]	Address[3]	Address[2]	Address[1]	Address[0]
	PTM_A9_ ETMACVR8	Address[31]	Address[30]	Address[29]	Address[28]	Address[27]	Address[26]	Address[25]	Address[24]
		Address[23]	Address[22]	Address[21]	Address[20]	Address[19]	Address[18]	Address[17]	Address[16]
		Address[15]	Address[14]	Address[13]	Address[12]	Address[11]	Address[10]	Address[9]	Address[8]
		Address[7]	Address[6]	Address[5]	Address[4]	Address[3]	Address[2]	Address[1]	Address[0]
	PTM_A9_ ETMACTR1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	Secure Level[1]	Secure Level[0]	Context ID[1]	Context ID[0]
		-	-	-	-	-	Access Type[2]	Access Type[1]	Access Type[0]
	PTM_A9_ ETMACTR2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	Secure Level[1]	Secure Level[0]	Context ID[1]	Context ID[0]
		-	-	-	-	-	Access Type[2]	Access Type[1]	Access Type[0]
	PTM_A9_ ETMACTR3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	Secure Level[1]	Secure Level[0]	Context ID[1]	Context ID[0]
		-	-	-	-	-	Access Type[2]	Access Type[1]	Access Type[0]
	PTM_A9_ ETMACTR4	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	Secure Level[1]	Secure Level[0]	Context ID[1]	Context ID[0]
		-	-	-	-	-	Access Type[2]	Access Type[1]	Access Type[0]
	PTM_A9_ ETMACTR5	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	Secure Level[1]	Secure Level[0]	Context ID[1]	Context ID[0]
		-	-	-	-	-	Access Type[2]	Access Type[1]	Access Type[0]
	PTM_A9_ ETMACTR6	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	Secure Level[1]	Secure Level[0]	Context ID[1]	Context ID[0]
		-	-	-	-	-	Access Type[2]	Access Type[1]	Access Type[0]
	PTM_A9_ ETMACTR7	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	Secure Level[1]	Secure Level[0]	Context ID[1]	Context ID[0]
		-	-	-	-	-	Access Type[2]	Access Type[1]	Access Type[0]
	PTM_A9_ ETMACTR8	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	Secure Level[1]	Secure Level[0]	Context ID[1]	Context ID[0]
		-	-	-	-	-	Access Type[2]	Access Type[1]	Access Type[0]
	PTM_A9_ETMCNTR LDVR1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		value[15]	value[14]	value[13]	value[12]	value[11]	value[10]	value[9]	value[8]
		value[7]	value[6]	value[5]	value[4]	value[3]	value[2]	value[1]	value[0]
	PTM_A9_ ETMCNTRLDVR2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		value[15]	value[14]	value[13]	value[12]	value[11]	value[10]	value[9]	value[8]
		value[7]	value[6]	value[5]	value[4]	value[3]	value[2]	value[1]	value[0]
	PTM_A9_ ETMCNTENR1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	event[16]
		event[15]	event[14]	event[13]	event[12]	event[11]	event[10]	event[9]	event[8]
		event[7]	event[6]	event[5]	event[4]	event[3]	event[2]	event[1]	event[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Debugger interface	PTM_A9_ ETMCNTENR2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	event[16]
		event[15]	event[14]	event[13]	event[12]	event[11]	event[10]	event[9]	event[8]
		event[7]	event[6]	event[5]	event[4]	event[3]	event[2]	event[1]	event[0]
	PTM_A9_ ETMCNTRLDEVR1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	event[16]
		event[15]	event[14]	event[13]	event[12]	event[11]	event[10]	event[9]	event[8]
		event[7]	event[6]	event[5]	event[4]	event[3]	event[2]	event[1]	event[0]
	PTM_A9_ ETMCNTRLDEVR2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	event[16]
		event[15]	event[14]	event[13]	event[12]	event[11]	event[10]	event[9]	event[8]
		event[7]	event[6]	event[5]	event[4]	event[3]	event[2]	event[1]	event[0]
	PTM_A9_ ETMCNTVR1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		value[15]	value[14]	value[13]	value[12]	value[11]	value[10]	value[9]	value[8]
		value[7]	value[6]	value[5]	value[4]	value[3]	value[2]	value[1]	value[0]
	PTM_A9_ ETMCNTVR2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		value[15]	value[14]	value[13]	value[12]	value[11]	value[10]	value[9]	value[8]
		value[7]	value[6]	value[5]	value[4]	value[3]	value[2]	value[1]	value[0]
	PTM_A9_ ETMSQ12EVR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	event[16]
		event[15]	event[14]	event[13]	event[12]	event[11]	event[10]	event[9]	event[8]
		event[7]	event[6]	event[5]	event[4]	event[3]	event[2]	event[1]	event[0]
	PTM_A9_ ETMSQ21EVR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	event[16]
		event[15]	event[14]	event[13]	event[12]	event[11]	event[10]	event[9]	event[8]
		event[7]	event[6]	event[5]	event[4]	event[3]	event[2]	event[1]	event[0]
	PTM_A9_ ETMSQ23EVR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	event[16]
		event[15]	event[14]	event[13]	event[12]	event[11]	event[10]	event[9]	event[8]
		event[7]	event[6]	event[5]	event[4]	event[3]	event[2]	event[1]	event[0]
	PTM_A9_ ETMSQ31EVR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	event[16]
		event[15]	event[14]	event[13]	event[12]	event[11]	event[10]	event[9]	event[8]
		event[7]	event[6]	event[5]	event[4]	event[3]	event[2]	event[1]	event[0]
	PTM_A9_ ETMSQ32EVR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	event[16]
		event[15]	event[14]	event[13]	event[12]	event[11]	event[10]	event[9]	event[8]
		event[7]	event[6]	event[5]	event[4]	event[3]	event[2]	event[1]	event[0]
	PTM_A9_ ETMSQ13EVR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	event[16]
		event[15]	event[14]	event[13]	event[12]	event[11]	event[10]	event[9]	event[8]
		event[7]	event[6]	event[5]	event[4]	event[3]	event[2]	event[1]	event[0]
	PTM_A9_ETMSQR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	state[1]	state[0]
	PTM_A9_ ETMEXTOUTEVR1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	event[16]
		event[15]	event[14]	event[13]	event[12]	event[11]	event[10]	event[9]	event[8]
		event[7]	event[6]	event[5]	event[4]	event[3]	event[2]	event[1]	event[0]
	PTM_A9_ ETMEXTOUTEVR2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	event[16]
		event[15]	event[14]	event[13]	event[12]	event[11]	event[10]	event[9]	event[8]
		event[7]	event[6]	event[5]	event[4]	event[3]	event[2]	event[1]	event[0]
	PTM_A9_ ETMCIDCVR1	value[31]	value[30]	value[29]	value[28]	value[27]	value[26]	value[25]	value[24]
		value[23]	value[22]	value[21]	value[20]	value[19]	value[18]	value[17]	value[16]
		value[15]	value[14]	value[13]	value[12]	value[11]	value[10]	value[9]	value[8]
		value[7]	value[6]	value[5]	value[4]	value[3]	value[2]	value[1]	value[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Debugger interface	PTM_A9_ ETMCIDCMR	mask[31]	mask[30]	mask[29]	mask[28]	mask[27]	mask[26]	mask[25]	mask[24]
		mask[23]	mask[22]	mask[21]	mask[20]	mask[19]	mask[18]	mask[17]	mask[16]
		mask[15]	mask[14]	mask[13]	mask[12]	mask[11]	mask[10]	mask[9]	mask[8]
		mask[7]	mask[6]	mask[5]	mask[4]	mask[3]	mask[2]	mask[1]	mask[0]
	PTM_A9_ ETMSYNCFR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	frequency[11]	frequency[10]	frequency[9]	frequency[8]
		frequency[7]	frequency[6]	frequency[5]	frequency[4]	frequency[3]	frequency[2]	frequency[1]	frequency[0]
	PTM_A9_ETMIDR	code[7]	code[6]	code[5]	code[4]	code[3]	code[2]	code[1]	code[0]
		-	-	-	-	Security	32-bitThumb	-	-
		-	-	-	-	Major[7]	Major[6]	Major[5]	Major[4]
		Minor[3]	Minor[2]	Minor[1]	Minor[0]	revision[3]	revision[2]	revision[1]	revision[0]
	PTM_A9_ETMCCER	-	-	-	-	-	-	Timestamps for DMB/DSB	DMB/DSB treated as waypoint
		Return stack implemented	Timestamping implemented	-	-	-	-	-	-
		Number of instrumentation resources[2]	Number of instrumentation resources[1]	Number of instrumentation resources[0]	-	-	Extended external input bus size[7]	Extended external input bus size[6]	Extended external input bus size[5]
		Extended external input bus size[4]	Extended external input bus size[3]	Extended external input bus size[2]	Extended external input bus size[1]	Extended external input bus size[0]	Number of extended external input selectors[2]	Number of extended external input selectors[1]	Number of extended external input selectors[0]
	PTM_A9_ ETMEXTINSELR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	Second extended external input selector[5]	Second extended external input selector[4]	Second extended external input selector[3]	Second extended external input selector[2]	Second extended external input selector[1]	Second extended external input selector[0]
		-	-	First extended external input selector[5]	First extended external input selector[4]	First extended external input selector[3]	First extended external input selector[2]	First extended external input selector[1]	First extended external input selector[0]
	PTM_A9_ ETMTSEVR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	event[16]
		event[15]	event[14]	event[13]	event[12]	event[11]	event[10]	event[9]	event[8]
		event[7]	event[6]	event[5]	event[4]	event[3]	event[2]	event[1]	event[0]
	PTM_A9_ ETMAUXCR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	Force synchronization packet insertion	Disable waypoint update packet	Disable timestamps on barriers	Disable forced overflow
	PTM_A9_ ETMTRACEIDR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ID[7]	ID[6]	ID[5]	ID[4]	ID[3]	ID[2]	ID[1]	ID[0]
	PTM_A9_OSLSR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	Lock access mechanism indicator	32-bit access required	PTM trace registers locked/ unlocked	Lock access mechanism indicator
	PTM_A9_CLAIMSET	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		CLAIMSET[7]	CLAIMSET[6]	CLAIMSET[5]	CLAIMSET[4]	CLAIMSET[3]	CLAIMSET[2]	CLAIMSET[1]	CLAIMSET[0]
	PTM_A9_CLAIMCLR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		CLAIMCLR[7]	CLAIMCLR[6]	CLAIMCLR[5]	CLAIMCLR[4]	CLAIMCLR[3]	CLAIMCLR[2]	CLAIMCLR[1]	CLAIMCLR[0]
	PTM_A9_LAR	ACCESS_W[31]	ACCESS_W[30]	ACCESS_W[29]	ACCESS_W[28]	ACCESS_W[27]	ACCESS_W[26]	ACCESS_W[25]	ACCESS_W[24]
		ACCESS_W[23]	ACCESS_W[22]	ACCESS_W[21]	ACCESS_W[20]	ACCESS_W[19]	ACCESS_W[18]	ACCESS_W[17]	ACCESS_W[16]
		ACCESS_W[15]	ACCESS_W[14]	ACCESS_W[13]	ACCESS_W[12]	ACCESS_W[11]	ACCESS_W[10]	ACCESS_W[9]	ACCESS_W[8]
		ACCESS_W[7]	ACCESS_W[6]	ACCESS_W[5]	ACCESS_W[4]	ACCESS_W[3]	ACCESS_W[2]	ACCESS_W[1]	ACCESS_W[0]
	PTM_A9_LSR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	LOCKTYPE	LOCKGRANT	LOCKEXIST

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
Debugger interface	PTM_A9_ AUTHSTATUS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		SNID[1]	SNID[0]	SID[1]	SID[0]	NSNID[1]	NSNID[0]	NSID[1]	NSID[0]
	PTM_A9_DEVTYPE	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Sub_type[3]	Sub_type[2]	Sub_type[1]	Sub_type[0]	Main_type[3]	Main_type[2]	Main_type[1]	Main_type[0]
	PTM_A9_ PERIPHID4	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		4KB_count[3]	4KB_count[2]	4KB_count[1]	4KB_count[0]	JEP106_code[3]	JEP106_code[2]	JEP106_code[1]	JEP106_code[0]
	PTM_A9_ PERIPHID0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Part_Number[7]	Part_Number[6]	Part_Number[5]	Part_Number[4]	Part_Number[3]	Part_Number[2]	Part_Number[1]	Part_Number[0]
	PTM_A9_ PERIPHID1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		JEP106_id[3]	JEP106_id[2]	JEP106_id[1]	JEP106_id[0]	Part_Number[11]	Part_Number[10]	Part_Number[9]	Part_Number[8]
	PTM_A9_ PERIPHID2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Revision[3]	Revision[2]	Revision[1]	Revision[0]	JEDEC	JEP106_id[6]	JEP106_id[5]	JEP106_id[4]
	PTM_A9_ PERIPHID3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		RevAnd[3]	RevAnd[2]	RevAnd[1]	RevAnd[0]	CUSTOM[3]	CUSTOM[2]	CUSTOM[1]	CUSTOM[0]
	PTM_A9_COMPID0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	PTM_A9_COMPID1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Class[3]	Class[2]	Class[1]	Class[0]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	PTM_A9_COMPID2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
	PTM_A9_COMPID3	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		Preamble[7]	Preamble[6]	Preamble[5]	Preamble[4]	Preamble[3]	Preamble[2]	Preamble[1]	Preamble[0]
JPEG codec unit*2	JCMOD	-	-	-	-	DSP	REDU[2]	REDU[1]	REDU[0]
	JCCMD	BRST	-	-	-	-	JEND	JRST	JSRT
	JCQTN	-	-	QT3[1]	QT3[0]	QT2[1]	QT2[0]	QT1[1]	QT1[0]
	JCHTN	-	-	HTA3	HTD3	HTA2	HTD2	HTA1	HTD1
	JCDRIU	DRIU[7]	DRIU[6]	DRIU[5]	DRIU[4]	DRIU[3]	DRIU[2]	DRIU[1]	DRIU[0]
	JCDRID	DRID[7]	DRID[6]	DRID[5]	DRID[4]	DRID[3]	DRID[2]	DRID[1]	DRID[0]
	JCVSZU	VSZU[7]	VSZU[6]	VSZU[5]	VSZU[4]	VSZU[3]	VSZU[2]	VSZU[1]	VSZU[0]
	JCVSZD	VSZD[7]	VSZD[6]	VSZD[5]	VSZD[4]	VSZD[3]	VSZD[2]	VSZD[1]	VSZD[0]
	JCHSZU	HSZU[7]	HSZU[6]	HSZU[5]	HSZU[4]	HSZU[3]	HSZU[2]	HSZU[1]	HSZU[0]
	JCHSZD	HSZD[7]	HSZD[6]	HSZD[5]	HSZD[4]	HSZD[3]	HSZD[2]	HSZD[1]	HSZD[0]
	JCDTCU	DCU[7]	DCU[6]	DCU[5]	DCU[4]	DCU[3]	DCU[2]	DCU[1]	DCU[0]
	JCDTCM	DCM[7]	DCM[6]	DCM[5]	DCM[4]	DCM[3]	DCM[2]	DCM[1]	DCM[0]
	JCDTCD	DCD[7]	DCD[6]	DCD[5]	DCD[4]	DCD[3]	DCD[2]	DCD[1]	DCD[0]
	JINTE0	INT7	INT6	INT5	-	INT3	-	-	-
	JINTS0	-	INS6	INS5	-	INS3	-	-	-
	JCDERR	-	-	-	-	ERR[3]	ERR[2]	ERR[1]	ERR[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
JPEG codec unit*2	JCRST	-	-	-	-	-	-	-	RST
	JIFECNT	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	JOUTRINI	JOUTRCMD	JOUTC	-	JOUTSWAP[2]	JOUTSWAP[1]	JOUTSWAP[0]
		-	DINRINI	DINRCMD	DINLC	-	DINSWAP[2]	DINSWAP[1]	DINSWAP[0]
	JIFESA	ESA[31]	ESA[30]	ESA[29]	ESA[28]	ESA[27]	ESA[26]	ESA[25]	ESA[24]
		ESA[23]	ESA[22]	ESA[21]	ESA[20]	ESA[19]	ESA[18]	ESA[17]	ESA[16]
		ESA[15]	ESA[14]	ESA[13]	ESA[12]	ESA[11]	ESA[10]	ESA[9]	ESA[8]
		ESA[7]	ESA[6]	ESA[5]	ESA[4]	ESA[3]	ESA[2]	ESA[1]	ESA[0]
	JIFESOFST	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	ESMW[14]	ESMW[13]	ESMW[12]	ESMW[11]	ESMW[10]	ESMW[9]	ESMW[8]
		ESMW[7]	ESMW[6]	ESMW[5]	ESMW[4]	ESMW[3]	ESMW[2]	ESMW[1]	ESMW[0]
	JIFEDA	EDA[31]	EDA[30]	EDA[29]	EDA[28]	EDA[27]	EDA[26]	EDA[25]	EDA[24]
		EDA[23]	EDA[22]	EDA[21]	EDA[20]	EDA[19]	EDA[18]	EDA[17]	EDA[16]
		EDA[15]	EDA[14]	EDA[13]	EDA[12]	EDA[11]	EDA[10]	EDA[9]	EDA[8]
		EDA[7]	EDA[6]	EDA[5]	EDA[4]	EDA[3]	EDA[2]	EDA[1]	EDA[0]
	JIFESLC	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		LINES[15]	LINES[14]	LINES[13]	LINES[12]	LINES[11]	LINES[10]	LINES[9]	LINES[8]
		LINES[7]	LINES[6]	LINES[5]	LINES[4]	LINES[3]	LINES[2]	LINES[1]	LINES[0]
	JIFEDDC	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		JDATAS[15]	JDATAS[14]	JDATAS[13]	JDATAS[12]	JDATAS[11]	JDATAS[10]	JDATAS[9]	JDATAS[8]
		JDATAS[7]	JDATAS[6]	JDATAS[5]	JDATAS[4]	JDATAS[3]	JDATAS[2]	JDATAS[1]	JDATAS[0]
	JIFDCNT	-	-	VINTER[1]	VINTER[0]	HINTER[1]	HINTER[0]	OPF[1]	OPF[0]
		-	-	-	-	-	-	-	-
		-	JINRINI	JINRCMD	JINC	-	JINSWAP[2]	JINSWAP[1]	JINSWAP[0]
		-	DOUTRINI	DOUTRCMD	DOUTLC	-	DOUTSWAP[2]	DOUTSWAP[1]	DOUTSWAP[0]
	JIFDSA	DSA[31]	DSA[30]	DSA[29]	DSA[28]	DSA[27]	DSA[26]	DSA[25]	DSA[24]
		DSA[23]	DSA[22]	DSA[21]	DSA[20]	DSA[19]	DSA[18]	DSA[17]	DSA[16]
		DSA[15]	DSA[14]	DSA[13]	DSA[12]	DSA[11]	DSA[10]	DSA[9]	DSA[8]
		DSA[7]	DSA[6]	DSA[5]	DSA[4]	DSA[3]	DSA[2]	DSA[1]	DSA[0]
	JIFDDOFST	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	DDMW[14]	DDMW[13]	DDMW[12]	DDMW[11]	DDMW[10]	DDMW[9]	DDMW[8]
		DDMW[7]	DDMW[6]	DDMW[5]	DDMW[4]	DDMW[3]	DDMW[2]	DDMW[1]	DDMW[0]
	JIFDDA	DDA[31]	DDA[30]	DDA[29]	DDA[28]	DDA[27]	DDA[26]	DDA[25]	DDA[24]
		DDA[23]	DDA[22]	DDA[21]	DDA[20]	DDA[19]	DDA[18]	DDA[17]	DDA[16]
		DDA[15]	DDA[14]	DDA[13]	DDA[12]	DDA[11]	DDA[10]	DDA[9]	DDA[8]
		DDA[7]	DDA[6]	DDA[5]	DDA[4]	DDA[3]	DDA[2]	DDA[1]	DDA[0]
	JIFDSDC	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		JDATAS[15]	JDATAS[14]	JDATAS[13]	JDATAS[12]	JDATAS[11]	JDATAS[10]	JDATAS[9]	JDATAS[8]
		JDATAS[7]	JDATAS[6]	JDATAS[5]	JDATAS[4]	JDATAS[3]	JDATAS[2]	JDATAS[1]	JDATAS[0]
	JIFDDLC	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		LINES[15]	LINES[14]	LINES[13]	LINES[12]	LINES[11]	LINES[10]	LINES[9]	LINES[8]
		LINES[7]	LINES[6]	LINES[5]	LINES[4]	LINES[3]	LINES[2]	LINES[1]	LINES[0]
	JIFDADT	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		ALPHA[7]	ALPHA[6]	ALPHA[5]	ALPHA[4]	ALPHA[3]	ALPHA[2]	ALPHA[1]	ALPHA[0]
	JINTE1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	CBTEN	DINLEN	JOUTEN	-	DBTEN	JINEN	DOUTLEN

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
JPEG codec unit*2	JINTS1	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	CBTF	DINLF	JOUTF	-	DBTF	JINF	DOUTLF
	JIFESVSZ	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DINYCHG	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	JIFESHSZ	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DOUTYCHG	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
EthernetAVB*2	CCC	-	-	-	-	-	-	FCE	LBME
		-	-	-	BOC	-	-	CSEL[1]	CSEL[0]
		-	-	-	-	-	-	-	DTSR
		-	-	-	-	-	-	OPC[1]	OPC[0]
	DBAT	TA[31]	TA[30]	TA[29]	TA[28]	TA[27]	TA[26]	TA[25]	TA[24]
		TA[23]	TA[22]	TA[21]	TA[20]	TA[19]	TA[18]	TA[17]	TA[16]
		TA[15]	TA[14]	TA[13]	TA[12]	TA[11]	TA[10]	TA[9]	TA[8]
		TA[7]	TA[6]	TA[5]	TA[4]	TA[3]	TA[2]	TA[1]	TA[0]
	DLR	-	-	-	-	-	-	-	-
		-	-	LBA21	LBA20	LBA19	LBA18	LBA17	LBA16
		LBA15	LBA14	LBA13	LBA12	LBA11	LBA10	LBA9	LBA8
		LBA7	LBA6	LBA5	LBA4	LBA3	LBA2	LBA1	LBA0
	CSR	-	-	-	-	-	-	-	-
		-	-	-	RPO	TPO3	TPO2	TPO1	TPO0
		-	-	-	-	-	-	-	DTS
		-	-	-	-	OPS[3]	OPS[2]	OPS[1]	OPS[0]
	CDAR0	CDA[31]	CDA[30]	CDA[29]	CDA[28]	CDA[27]	CDA[26]	CDA[25]	CDA[24]
		CDA[23]	CDA[22]	CDA[21]	CDA[20]	CDA[19]	CDA[18]	CDA[17]	CDA[16]
		CDA[15]	CDA[14]	CDA[13]	CDA[12]	CDA[11]	CDA[10]	CDA[9]	CDA[8]
		CDA[7]	CDA[6]	CDA[5]	CDA[4]	CDA[3]	CDA[2]	CDA[1]	CDA[0]
	CDAR1	CDA[31]	CDA[30]	CDA[29]	CDA[28]	CDA[27]	CDA[26]	CDA[25]	CDA[24]
		CDA[23]	CDA[22]	CDA[21]	CDA[20]	CDA[19]	CDA[18]	CDA[17]	CDA[16]
		CDA[15]	CDA[14]	CDA[13]	CDA[12]	CDA[11]	CDA[10]	CDA[9]	CDA[8]
		CDA[7]	CDA[6]	CDA[5]	CDA[4]	CDA[3]	CDA[2]	CDA[1]	CDA[0]
	CDAR2	CDA[31]	CDA[30]	CDA[29]	CDA[28]	CDA[27]	CDA[26]	CDA[25]	CDA[24]
		CDA[23]	CDA[22]	CDA[21]	CDA[20]	CDA[19]	CDA[18]	CDA[17]	CDA[16]
		CDA[15]	CDA[14]	CDA[13]	CDA[12]	CDA[11]	CDA[10]	CDA[9]	CDA[8]
		CDA[7]	CDA[6]	CDA[5]	CDA[4]	CDA[3]	CDA[2]	CDA[1]	CDA[0]
	CDAR3	CDA[31]	CDA[30]	CDA[29]	CDA[28]	CDA[27]	CDA[26]	CDA[25]	CDA[24]
		CDA[23]	CDA[22]	CDA[21]	CDA[20]	CDA[19]	CDA[18]	CDA[17]	CDA[16]
		CDA[15]	CDA[14]	CDA[13]	CDA[12]	CDA[11]	CDA[10]	CDA[9]	CDA[8]
		CDA[7]	CDA[6]	CDA[5]	CDA[4]	CDA[3]	CDA[2]	CDA[1]	CDA[0]
	CDAR4	CDA[31]	CDA[30]	CDA[29]	CDA[28]	CDA[27]	CDA[26]	CDA[25]	CDA[24]
		CDA[23]	CDA[22]	CDA[21]	CDA[20]	CDA[19]	CDA[18]	CDA[17]	CDA[16]
		CDA[15]	CDA[14]	CDA[13]	CDA[12]	CDA[11]	CDA[10]	CDA[9]	CDA[8]
		CDA[7]	CDA[6]	CDA[5]	CDA[4]	CDA[3]	CDA[2]	CDA[1]	CDA[0]
	CDAR5	CDA[31]	CDA[30]	CDA[29]	CDA[28]	CDA[27]	CDA[26]	CDA[25]	CDA[24]
		CDA[23]	CDA[22]	CDA[21]	CDA[20]	CDA[19]	CDA[18]	CDA[17]	CDA[16]
		CDA[15]	CDA[14]	CDA[13]	CDA[12]	CDA[11]	CDA[10]	CDA[9]	CDA[8]
		CDA[7]	CDA[6]	CDA[5]	CDA[4]	CDA[3]	CDA[2]	CDA[1]	CDA[0]
	CDAR6	CDA[31]	CDA[30]	CDA[29]	CDA[28]	CDA[27]	CDA[26]	CDA[25]	CDA[24]
		CDA[23]	CDA[22]	CDA[21]	CDA[20]	CDA[19]	CDA[18]	CDA[17]	CDA[16]
		CDA[15]	CDA[14]	CDA[13]	CDA[12]	CDA[11]	CDA[10]	CDA[9]	CDA[8]
		CDA[7]	CDA[6]	CDA[5]	CDA[4]	CDA[3]	CDA[2]	CDA[1]	CDA[0]
	CDAR7	CDA[31]	CDA[30]	CDA[29]	CDA[28]	CDA[27]	CDA[26]	CDA[25]	CDA[24]
		CDA[23]	CDA[22]	CDA[21]	CDA[20]	CDA[19]	CDA[18]	CDA[17]	CDA[16]
		CDA[15]	CDA[14]	CDA[13]	CDA[12]	CDA[11]	CDA[10]	CDA[9]	CDA[8]
		CDA[7]	CDA[6]	CDA[5]	CDA[4]	CDA[3]	CDA[2]	CDA[1]	CDA[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
EthernetAVB*2	CDAR8	CDA[31]	CDA[30]	CDA[29]	CDA[28]	CDA[27]	CDA[26]	CDA[25]	CDA[24]
		CDA[23]	CDA[22]	CDA[21]	CDA[20]	CDA[19]	CDA[18]	CDA[17]	CDA[16]
		CDA[15]	CDA[14]	CDA[13]	CDA[12]	CDA[11]	CDA[10]	CDA[9]	CDA[8]
		CDA[7]	CDA[6]	CDA[5]	CDA[4]	CDA[3]	CDA[2]	CDA[1]	CDA[0]
	CDAR9	CDA[31]	CDA[30]	CDA[29]	CDA[28]	CDA[27]	CDA[26]	CDA[25]	CDA[24]
		CDA[23]	CDA[22]	CDA[21]	CDA[20]	CDA[19]	CDA[18]	CDA[17]	CDA[16]
		CDA[15]	CDA[14]	CDA[13]	CDA[12]	CDA[11]	CDA[10]	CDA[9]	CDA[8]
		CDA[7]	CDA[6]	CDA[5]	CDA[4]	CDA[3]	CDA[2]	CDA[1]	CDA[0]
	CDAR10	CDA[31]	CDA[30]	CDA[29]	CDA[28]	CDA[27]	CDA[26]	CDA[25]	CDA[24]
		CDA[23]	CDA[22]	CDA[21]	CDA[20]	CDA[19]	CDA[18]	CDA[17]	CDA[16]
		CDA[15]	CDA[14]	CDA[13]	CDA[12]	CDA[11]	CDA[10]	CDA[9]	CDA[8]
		CDA[7]	CDA[6]	CDA[5]	CDA[4]	CDA[3]	CDA[2]	CDA[1]	CDA[0]
	CDAR11	CDA[31]	CDA[30]	CDA[29]	CDA[28]	CDA[27]	CDA[26]	CDA[25]	CDA[24]
		CDA[23]	CDA[22]	CDA[21]	CDA[20]	CDA[19]	CDA[18]	CDA[17]	CDA[16]
		CDA[15]	CDA[14]	CDA[13]	CDA[12]	CDA[11]	CDA[10]	CDA[9]	CDA[8]
		CDA[7]	CDA[6]	CDA[5]	CDA[4]	CDA[3]	CDA[2]	CDA[1]	CDA[0]
	CDAR12	CDA[31]	CDA[30]	CDA[29]	CDA[28]	CDA[27]	CDA[26]	CDA[25]	CDA[24]
		CDA[23]	CDA[22]	CDA[21]	CDA[20]	CDA[19]	CDA[18]	CDA[17]	CDA[16]
		CDA[15]	CDA[14]	CDA[13]	CDA[12]	CDA[11]	CDA[10]	CDA[9]	CDA[8]
		CDA[7]	CDA[6]	CDA[5]	CDA[4]	CDA[3]	CDA[2]	CDA[1]	CDA[0]
	CDAR13	CDA[31]	CDA[30]	CDA[29]	CDA[28]	CDA[27]	CDA[26]	CDA[25]	CDA[24]
		CDA[23]	CDA[22]	CDA[21]	CDA[20]	CDA[19]	CDA[18]	CDA[17]	CDA[16]
		CDA[15]	CDA[14]	CDA[13]	CDA[12]	CDA[11]	CDA[10]	CDA[9]	CDA[8]
		CDA[7]	CDA[6]	CDA[5]	CDA[4]	CDA[3]	CDA[2]	CDA[1]	CDA[0]
	CDAR14	CDA[31]	CDA[30]	CDA[29]	CDA[28]	CDA[27]	CDA[26]	CDA[25]	CDA[24]
		CDA[23]	CDA[22]	CDA[21]	CDA[20]	CDA[19]	CDA[18]	CDA[17]	CDA[16]
		CDA[15]	CDA[14]	CDA[13]	CDA[12]	CDA[11]	CDA[10]	CDA[9]	CDA[8]
		CDA[7]	CDA[6]	CDA[5]	CDA[4]	CDA[3]	CDA[2]	CDA[1]	CDA[0]
	CDAR15	CDA[31]	CDA[30]	CDA[29]	CDA[28]	CDA[27]	CDA[26]	CDA[25]	CDA[24]
		CDA[23]	CDA[22]	CDA[21]	CDA[20]	CDA[19]	CDA[18]	CDA[17]	CDA[16]
		CDA[15]	CDA[14]	CDA[13]	CDA[12]	CDA[11]	CDA[10]	CDA[9]	CDA[8]
		CDA[7]	CDA[6]	CDA[5]	CDA[4]	CDA[3]	CDA[2]	CDA[1]	CDA[0]
	CDAR16	CDA[31]	CDA[30]	CDA[29]	CDA[28]	CDA[27]	CDA[26]	CDA[25]	CDA[24]
		CDA[23]	CDA[22]	CDA[21]	CDA[20]	CDA[19]	CDA[18]	CDA[17]	CDA[16]
		CDA[15]	CDA[14]	CDA[13]	CDA[12]	CDA[11]	CDA[10]	CDA[9]	CDA[8]
		CDA[7]	CDA[6]	CDA[5]	CDA[4]	CDA[3]	CDA[2]	CDA[1]	CDA[0]
	CDAR17	CDA[31]	CDA[30]	CDA[29]	CDA[28]	CDA[27]	CDA[26]	CDA[25]	CDA[24]
		CDA[23]	CDA[22]	CDA[21]	CDA[20]	CDA[19]	CDA[18]	CDA[17]	CDA[16]
		CDA[15]	CDA[14]	CDA[13]	CDA[12]	CDA[11]	CDA[10]	CDA[9]	CDA[8]
		CDA[7]	CDA[6]	CDA[5]	CDA[4]	CDA[3]	CDA[2]	CDA[1]	CDA[0]
	CDAR18	CDA[31]	CDA[30]	CDA[29]	CDA[28]	CDA[27]	CDA[26]	CDA[25]	CDA[24]
		CDA[23]	CDA[22]	CDA[21]	CDA[20]	CDA[19]	CDA[18]	CDA[17]	CDA[16]
		CDA[15]	CDA[14]	CDA[13]	CDA[12]	CDA[11]	CDA[10]	CDA[9]	CDA[8]
		CDA[7]	CDA[6]	CDA[5]	CDA[4]	CDA[3]	CDA[2]	CDA[1]	CDA[0]
	CDAR19	CDA[31]	CDA[30]	CDA[29]	CDA[28]	CDA[27]	CDA[26]	CDA[25]	CDA[24]
		CDA[23]	CDA[22]	CDA[21]	CDA[20]	CDA[19]	CDA[18]	CDA[17]	CDA[16]
		CDA[15]	CDA[14]	CDA[13]	CDA[12]	CDA[11]	CDA[10]	CDA[9]	CDA[8]
		CDA[7]	CDA[6]	CDA[5]	CDA[4]	CDA[3]	CDA[2]	CDA[1]	CDA[0]
	CDAR20	CDA[31]	CDA[30]	CDA[29]	CDA[28]	CDA[27]	CDA[26]	CDA[25]	CDA[24]
		CDA[23]	CDA[22]	CDA[21]	CDA[20]	CDA[19]	CDA[18]	CDA[17]	CDA[16]
		CDA[15]	CDA[14]	CDA[13]	CDA[12]	CDA[11]	CDA[10]	CDA[9]	CDA[8]
		CDA[7]	CDA[6]	CDA[5]	CDA[4]	CDA[3]	CDA[2]	CDA[1]	CDA[0]
	CDAR21	CDA[31]	CDA[30]	CDA[29]	CDA[28]	CDA[27]	CDA[26]	CDA[25]	CDA[24]
		CDA[23]	CDA[22]	CDA[21]	CDA[20]	CDA[19]	CDA[18]	CDA[17]	CDA[16]
		CDA[15]	CDA[14]	CDA[13]	CDA[12]	CDA[11]	CDA[10]	CDA[9]	CDA[8]
		CDA[7]	CDA[6]	CDA[5]	CDA[4]	CDA[3]	CDA[2]	CDA[1]	CDA[0]
	ESR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	EIL	ET[3]	ET[2]	ET[1]	ET[0]
		-	-	-	EQN[4]	EQN[3]	EQN[2]	EQN[1]	EQN[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
EthernetAVB*2	RCR	-	-	-	RFCL[12]	RFCL[11]	RFCL[10]	RFCL[9]	RFCL[8]
		RFCL[7]	RFCL[6]	RFCL[5]	RFCL[4]	RFCL[3]	RFCL[2]	RFCL[1]	RFCL[0]
		-	-	-	-	-	-	-	-
		-	-	ETS2	ETS0	ESF[1]	ESF[0]	ENCF	EFFS
	RQC0	-	-	UFCC3[1]	UFCC3[0]	-	-	RSM3[1]	RSM3[0]
		-	-	UFCC2[1]	UFCC2[0]	-	-	RSM2[1]	RSM2[0]
		-	-	UFCC1[1]	UFCC1[0]	-	-	RSM1[1]	RSM1[0]
		-	-	UFCC0[1]	UFCC0[0]	-	-	RSM0[1]	RSM0[0]
	RQC1	-	-	UFCC3[1]	UFCC3[0]	-	-	RSM3[1]	RSM3[0]
		-	-	UFCC2[1]	UFCC2[0]	-	-	RSM2[1]	RSM2[0]
		-	-	UFCC1[1]	UFCC1[0]	-	-	RSM1[1]	RSM1[0]
		-	-	UFCC0[1]	UFCC0[0]	-	-	RSM0[1]	RSM0[0]
	RQC2	-	-	UFCC3[1]	UFCC3[0]	-	-	RSM3[1]	RSM3[0]
		-	-	UFCC2[1]	UFCC2[0]	-	-	RSM2[1]	RSM2[0]
		-	-	UFCC1[1]	UFCC1[0]	-	-	RSM1[1]	RSM1[0]
		-	-	UFCC0[1]	UFCC0[0]	-	-	RSM0[1]	RSM0[0]
	RQC3	-	-	UFCC3[1]	UFCC3[0]	-	-	RSM3[1]	RSM3[0]
		-	-	UFCC2[1]	UFCC2[0]	-	-	RSM2[1]	RSM2[0]
		-	-	UFCC1[1]	UFCC1[0]	-	-	RSM1[1]	RSM1[0]
		-	-	UFCC0[1]	UFCC0[0]	-	-	RSM0[1]	RSM0[0]
	RQC4	-	-	UFCC3[1]	UFCC3[0]	-	-	RSM3[1]	RSM3[0]
		-	-	UFCC2[1]	UFCC2[0]	-	-	RSM2[1]	RSM2[0]
		-	-	UFCC1[1]	UFCC1[0]	-	-	RSM1[1]	RSM1[0]
		-	-	UFCC0[1]	UFCC0[0]	-	-	RSM0[1]	RSM0[0]
	RPC	-	-	-	-	-	-	-	-
		DCNT[7]	DCNT[6]	DCNT[5]	DCNT[4]	DCNT[3]	DCNT[2]	DCNT[1]	DCNT[0]
		-	-	-	-	-	PCNT[2]	PCNT[1]	PCNT[0]
		-	-	-	-	-	-	-	-
	UFCS	-	-	SL3[5]	SL3[4]	SL3[3]	SL3[2]	SL3[1]	SL3[0]
		-	-	SL2[5]	SL2[4]	SL2[3]	SL2[2]	SL2[1]	SL2[0]
		-	-	SL1[5]	SL1[4]	SL1[3]	SL1[2]	SL1[1]	SL1[0]
		-	-	SL0[5]	SL0[4]	SL0[3]	SL0[2]	SL0[1]	SL0[0]
	UFCV0	-	-	CV3[5]	CV3[4]	CV3[3]	CV3[2]	CV3[1]	CV3[0]
		-	-	CV2[5]	CV2[4]	CV2[3]	CV2[2]	CV2[1]	CV2[0]
		-	-	CV1[5]	CV1[4]	CV1[3]	CV1[2]	CV1[1]	CV1[0]
		-	-	CV0[5]	CV0[4]	CV0[3]	CV0[2]	CV0[1]	CV0[0]
	UFCV1	-	-	CV3[5]	CV3[4]	CV3[3]	CV3[2]	CV3[1]	CV3[0]
		-	-	CV2[5]	CV2[4]	CV2[3]	CV2[2]	CV2[1]	CV2[0]
		-	-	CV1[5]	CV1[4]	CV1[3]	CV1[2]	CV1[1]	CV1[0]
		-	-	CV0[5]	CV0[4]	CV0[3]	CV0[2]	CV0[1]	CV0[0]
	UFCV2	-	-	CV3[5]	CV3[4]	CV3[3]	CV3[2]	CV3[1]	CV3[0]
		-	-	CV2[5]	CV2[4]	CV2[3]	CV2[2]	CV2[1]	CV2[0]
		-	-	CV1[5]	CV1[4]	CV1[3]	CV1[2]	CV1[1]	CV1[0]
		-	-	CV0[5]	CV0[4]	CV0[3]	CV0[2]	CV0[1]	CV0[0]
	UFCV3	-	-	CV3[5]	CV3[4]	CV3[3]	CV3[2]	CV3[1]	CV3[0]
		-	-	CV2[5]	CV2[4]	CV2[3]	CV2[2]	CV2[1]	CV2[0]
		-	-	CV1[5]	CV1[4]	CV1[3]	CV1[2]	CV1[1]	CV1[0]
		-	-	CV0[5]	CV0[4]	CV0[3]	CV0[2]	CV0[1]	CV0[0]
	UFCV4	-	-	CV3[5]	CV3[4]	CV3[3]	CV3[2]	CV3[1]	CV3[0]
		-	-	CV2[5]	CV2[4]	CV2[3]	CV2[2]	CV2[1]	CV2[0]
		-	-	CV1[5]	CV1[4]	CV1[3]	CV1[2]	CV1[1]	CV1[0]
		-	-	CV0[5]	CV0[4]	CV0[3]	CV0[2]	CV0[1]	CV0[0]
	UFCD0	-	-	DV3[5]	DV3[4]	DV3[3]	DV3[2]	DV3[1]	DV3[0]
		-	-	DV2[5]	DV2[4]	DV2[3]	DV2[2]	DV2[1]	DV2[0]
		-	-	DV1[5]	DV1[4]	DV1[3]	DV1[2]	DV1[1]	DV1[0]
		-	-	DV0[5]	DV0[4]	DV0[3]	DV0[2]	DV0[1]	DV0[0]
	UFCD1	-	-	DV3[5]	DV3[4]	DV3[3]	DV3[2]	DV3[1]	DV3[0]
		-	-	DV2[5]	DV2[4]	DV2[3]	DV2[2]	DV2[1]	DV2[0]
		-	-	DV1[5]	DV1[4]	DV1[3]	DV1[2]	DV1[1]	DV1[0]
		-	-	DV0[5]	DV0[4]	DV0[3]	DV0[2]	DV0[1]	DV0[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
EthernetAVB*2	UFCD2	-	-	DV3[5]	DV3[4]	DV3[3]	DV3[2]	DV3[1]	DV3[0]
		-	-	DV2[5]	DV2[4]	DV2[3]	DV2[2]	DV2[1]	DV2[0]
		-	-	DV1[5]	DV1[4]	DV1[3]	DV1[2]	DV1[1]	DV1[0]
		-	-	DV0[5]	DV0[4]	DV0[3]	DV0[2]	DV0[1]	DV0[0]
	UFCD3	-	-	DV3[5]	DV3[4]	DV3[3]	DV3[2]	DV3[1]	DV3[0]
		-	-	DV2[5]	DV2[4]	DV2[3]	DV2[2]	DV2[1]	DV2[0]
		-	-	DV1[5]	DV1[4]	DV1[3]	DV1[2]	DV1[1]	DV1[0]
		-	-	DV0[5]	DV0[4]	DV0[3]	DV0[2]	DV0[1]	DV0[0]
	UFCD4	-	-	DV3[5]	DV3[4]	DV3[3]	DV3[2]	DV3[1]	DV3[0]
		-	-	DV2[5]	DV2[4]	DV2[3]	DV2[2]	DV2[1]	DV2[0]
		-	-	DV1[5]	DV1[4]	DV1[3]	DV1[2]	DV1[1]	DV1[0]
		-	-	DV0[5]	DV0[4]	DV0[3]	DV0[2]	DV0[1]	DV0[0]
	SFO	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	FBP[5]	FBP[4]	FBP[3]	FBP[2]	FBP[1]	FBP[0]
	SFP0	FP0[31]	FP0[30]	FP0[29]	FP0[28]	FP0[27]	FP0[26]	FP0[25]	FP0[24]
		FP0[23]	FP0[22]	FP0[21]	FP0[20]	FP0[19]	FP0[18]	FP0[17]	FP0[16]
		FP0[15]	FP0[14]	FP0[13]	FP0[12]	FP0[11]	FP0[10]	FP0[9]	FP0[8]
		FP0[7]	FP0[6]	FP0[5]	FP0[4]	FP0[3]	FP0[2]	FP0[1]	FP0[0]
	SFP1	FP0[63]	FP0[62]	FP0[61]	FP0[60]	FP0[59]	FP0[58]	FP0[57]	FP0[56]
		FP0[55]	FP0[54]	FP0[53]	FP0[52]	FP0[51]	FP0[50]	FP0[49]	FP0[48]
		FP0[47]	FP0[46]	FP0[45]	FP0[44]	FP0[43]	FP0[42]	FP0[41]	FP0[40]
		FP0[39]	FP0[38]	FP0[37]	FP0[36]	FP0[35]	FP0[34]	FP0[33]	FP0[32]
	SFP2	FP1[31]	FP1[30]	FP1[29]	FP1[28]	FP1[27]	FP1[26]	FP1[25]	FP1[24]
		FP1[23]	FP1[22]	FP1[21]	FP1[20]	FP1[19]	FP1[18]	FP1[17]	FP1[16]
		FP1[15]	FP1[14]	FP1[13]	FP1[12]	FP1[11]	FP1[10]	FP1[9]	FP1[8]
		FP1[7]	FP1[6]	FP1[5]	FP1[4]	FP1[3]	FP1[2]	FP1[1]	FP1[0]
	SFP3	FP1[63]	FP1[62]	FP1[61]	FP1[60]	FP1[59]	FP1[58]	FP1[57]	FP1[56]
		FP1[55]	FP1[54]	FP1[53]	FP1[52]	FP1[51]	FP1[50]	FP1[49]	FP1[48]
		FP1[47]	FP1[46]	FP1[45]	FP1[44]	FP1[43]	FP1[42]	FP1[41]	FP1[40]
		FP1[39]	FP1[38]	FP1[37]	FP1[36]	FP1[35]	FP1[34]	FP1[33]	FP1[32]
	SFP4	FP2[31]	FP2[30]	FP2[29]	FP2[28]	FP2[27]	FP2[26]	FP2[25]	FP2[24]
		FP2[23]	FP2[22]	FP2[21]	FP2[20]	FP2[19]	FP2[18]	FP2[17]	FP2[16]
		FP2[15]	FP2[14]	FP2[13]	FP2[12]	FP2[11]	FP2[10]	FP2[9]	FP2[8]
		FP2[7]	FP2[6]	FP2[5]	FP2[4]	FP2[3]	FP2[2]	FP2[1]	FP2[0]
	SFP5	FP2[63]	FP2[62]	FP2[61]	FP2[60]	FP2[59]	FP2[58]	FP2[57]	FP2[56]
		FP2[55]	FP2[54]	FP2[53]	FP2[52]	FP2[51]	FP2[50]	FP2[49]	FP2[48]
		FP2[47]	FP2[46]	FP2[45]	FP2[44]	FP2[43]	FP2[42]	FP2[41]	FP2[40]
		FP2[39]	FP2[38]	FP2[37]	FP2[36]	FP2[35]	FP2[34]	FP2[33]	FP2[32]
	SFP6	FP3[31]	FP3[30]	FP3[29]	FP3[28]	FP3[27]	FP3[26]	FP3[25]	FP3[24]
		FP3[23]	FP3[22]	FP3[21]	FP3[20]	FP3[19]	FP3[18]	FP3[17]	FP3[16]
		FP3[15]	FP3[14]	FP3[13]	FP3[12]	FP3[11]	FP3[10]	FP3[9]	FP3[8]
		FP3[7]	FP3[6]	FP3[5]	FP3[4]	FP3[3]	FP3[2]	FP3[1]	FP3[0]
	SFP7	FP3[63]	FP3[62]	FP3[61]	FP3[60]	FP3[59]	FP3[58]	FP3[57]	FP3[56]
		FP3[55]	FP3[54]	FP3[53]	FP3[52]	FP3[51]	FP3[50]	FP3[49]	FP3[48]
		FP3[47]	FP3[46]	FP3[45]	FP3[44]	FP3[43]	FP3[42]	FP3[41]	FP3[40]
		FP3[39]	FP3[38]	FP3[37]	FP3[36]	FP3[35]	FP3[34]	FP3[33]	FP3[32]
	SFP8	FP4[31]	FP4[30]	FP4[29]	FP4[28]	FP4[27]	FP4[26]	FP4[25]	FP4[24]
		FP4[23]	FP4[22]	FP4[21]	FP4[20]	FP4[19]	FP4[18]	FP4[17]	FP4[16]
		FP4[15]	FP4[14]	FP4[13]	FP4[12]	FP4[11]	FP4[10]	FP4[9]	FP4[8]
		FP4[7]	FP4[6]	FP4[5]	FP4[4]	FP4[3]	FP4[2]	FP4[1]	FP4[0]
	SFP9	FP4[63]	FP4[62]	FP4[61]	FP4[60]	FP4[59]	FP4[58]	FP4[57]	FP4[56]
		FP4[55]	FP4[54]	FP4[53]	FP4[52]	FP4[51]	FP4[50]	FP4[49]	FP4[48]
		FP4[47]	FP4[46]	FP4[45]	FP4[44]	FP4[43]	FP4[42]	FP4[41]	FP4[40]
		FP4[39]	FP4[38]	FP4[37]	FP4[36]	FP4[35]	FP4[34]	FP4[33]	FP4[32]
	SFP10	FP5[31]	FP5[30]	FP5[29]	FP5[28]	FP5[27]	FP5[26]	FP5[25]	FP5[24]
		FP5[23]	FP5[22]	FP5[21]	FP5[20]	FP5[19]	FP5[18]	FP5[17]	FP5[16]
		FP5[15]	FP5[14]	FP5[13]	FP5[12]	FP5[11]	FP5[10]	FP5[9]	FP5[8]
		FP5[7]	FP5[6]	FP5[5]	FP5[4]	FP5[3]	FP5[2]	FP5[1]	FP5[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
EthernetAVB*2	SFP11	FP5[63]	FP5[62]	FP5[61]	FP5[60]	FP5[59]	FP5[58]	FP5[57]	FP5[56]
		FP5[55]	FP5[54]	FP5[53]	FP5[52]	FP5[51]	FP5[50]	FP5[49]	FP5[48]
		FP5[47]	FP5[46]	FP5[45]	FP5[44]	FP5[43]	FP5[42]	FP5[41]	FP5[40]
		FP5[39]	FP5[38]	FP5[37]	FP5[36]	FP5[35]	FP5[34]	FP5[33]	FP5[32]
	SFP12	FP6[31]	FP6[30]	FP6[29]	FP6[28]	FP6[27]	FP6[26]	FP6[25]	FP6[24]
		FP6[23]	FP6[22]	FP6[21]	FP6[20]	FP6[19]	FP6[18]	FP6[17]	FP6[16]
		FP6[15]	FP6[14]	FP6[13]	FP6[12]	FP6[11]	FP6[10]	FP6[9]	FP6[8]
		FP6[7]	FP6[6]	FP6[5]	FP6[4]	FP6[3]	FP6[2]	FP6[1]	FP6[0]
	SFP13	FP6[63]	FP6[62]	FP6[61]	FP6[60]	FP6[59]	FP6[58]	FP6[57]	FP6[56]
		FP6[55]	FP6[54]	FP6[53]	FP6[52]	FP6[51]	FP6[50]	FP6[49]	FP6[48]
		FP6[47]	FP6[46]	FP6[45]	FP6[44]	FP6[43]	FP6[42]	FP6[41]	FP6[40]
		FP6[39]	FP6[38]	FP6[37]	FP6[36]	FP6[35]	FP6[34]	FP6[33]	FP6[32]
	SFP14	FP7[31]	FP7[30]	FP7[29]	FP7[28]	FP7[27]	FP7[26]	FP7[25]	FP7[24]
		FP7[23]	FP7[22]	FP7[21]	FP7[20]	FP7[19]	FP7[18]	FP7[17]	FP7[16]
		FP7[15]	FP7[14]	FP7[13]	FP7[12]	FP7[11]	FP7[10]	FP7[9]	FP7[8]
		FP7[7]	FP7[6]	FP7[5]	FP7[4]	FP7[3]	FP7[2]	FP7[1]	FP7[0]
	SFP15	FP7[63]	FP7[62]	FP7[61]	FP7[60]	FP7[59]	FP7[58]	FP7[57]	FP7[56]
		FP7[55]	FP7[54]	FP7[53]	FP7[52]	FP7[51]	FP7[50]	FP7[49]	FP7[48]
		FP7[47]	FP7[46]	FP7[45]	FP7[44]	FP7[43]	FP7[42]	FP7[41]	FP7[40]
		FP7[39]	FP7[38]	FP7[37]	FP7[36]	FP7[35]	FP7[34]	FP7[33]	FP7[32]
	SFP16	FP8[31]	FP8[30]	FP8[29]	FP8[28]	FP8[27]	FP8[26]	FP8[25]	FP8[24]
		FP8[23]	FP8[22]	FP8[21]	FP8[20]	FP8[19]	FP8[18]	FP8[17]	FP8[16]
		FP8[15]	FP8[14]	FP8[13]	FP8[12]	FP8[11]	FP8[10]	FP8[9]	FP8[8]
		FP8[7]	FP8[6]	FP8[5]	FP8[4]	FP8[3]	FP8[2]	FP8[1]	FP8[0]
	SFP17	FP8[63]	FP8[62]	FP8[61]	FP8[60]	FP8[59]	FP8[58]	FP8[57]	FP8[56]
		FP8[55]	FP8[54]	FP8[53]	FP8[52]	FP8[51]	FP8[50]	FP8[49]	FP8[48]
		FP8[47]	FP8[46]	FP8[45]	FP8[44]	FP8[43]	FP8[42]	FP8[41]	FP8[40]
		FP8[39]	FP8[38]	FP8[37]	FP8[36]	FP8[35]	FP8[34]	FP8[33]	FP8[32]
	SFP18	FP9[31]	FP9[30]	FP9[29]	FP9[28]	FP9[27]	FP9[26]	FP9[25]	FP9[24]
		FP9[23]	FP9[22]	FP9[21]	FP9[20]	FP9[19]	FP9[18]	FP9[17]	FP9[16]
		FP9[15]	FP9[14]	FP9[13]	FP9[12]	FP9[11]	FP9[10]	FP9[9]	FP9[8]
		FP9[7]	FP9[6]	FP9[5]	FP9[4]	FP9[3]	FP9[2]	FP9[1]	FP9[0]
	SFP19	FP9[63]	FP9[62]	FP9[61]	FP9[60]	FP9[59]	FP9[58]	FP9[57]	FP9[56]
		FP9[55]	FP9[54]	FP9[53]	FP9[52]	FP9[51]	FP9[50]	FP9[49]	FP9[48]
		FP9[47]	FP9[46]	FP9[45]	FP9[44]	FP9[43]	FP9[42]	FP9[41]	FP9[40]
		FP9[39]	FP9[38]	FP9[37]	FP9[36]	FP9[35]	FP9[34]	FP9[33]	FP9[32]
	SFP20	FP10[31]	FP10[30]	FP10[29]	FP10[28]	FP10[27]	FP10[26]	FP10[25]	FP10[24]
		FP10[23]	FP10[22]	FP10[21]	FP10[20]	FP10[19]	FP10[18]	FP10[17]	FP10[16]
		FP10[15]	FP10[14]	FP10[13]	FP10[12]	FP10[11]	FP10[10]	FP10[9]	FP10[8]
		FP10[7]	FP10[6]	FP10[5]	FP10[4]	FP10[3]	FP10[2]	FP10[1]	FP10[0]
	SFP21	FP10[63]	FP10[62]	FP10[61]	FP10[60]	FP10[59]	FP10[58]	FP10[57]	FP10[56]
		FP10[55]	FP10[54]	FP10[53]	FP10[52]	FP10[51]	FP10[50]	FP10[49]	FP10[48]
		FP10[47]	FP10[46]	FP10[45]	FP10[44]	FP10[43]	FP10[42]	FP10[41]	FP10[40]
		FP10[39]	FP10[38]	FP10[37]	FP10[36]	FP10[35]	FP10[34]	FP10[33]	FP10[32]
	SFP22	FP11[31]	FP11[30]	FP11[29]	FP11[28]	FP11[27]	FP11[26]	FP11[25]	FP11[24]
		FP11[23]	FP11[22]	FP11[21]	FP11[20]	FP11[19]	FP11[18]	FP11[17]	FP11[16]
		FP11[15]	FP11[14]	FP11[13]	FP11[12]	FP11[11]	FP11[10]	FP11[9]	FP11[8]
		FP11[7]	FP11[6]	FP11[5]	FP11[4]	FP11[3]	FP11[2]	FP11[1]	FP11[0]
	SFP23	FP11[63]	FP11[62]	FP11[61]	FP11[60]	FP11[59]	FP11[58]	FP11[57]	FP11[56]
		FP11[55]	FP11[54]	FP11[53]	FP11[52]	FP11[51]	FP11[50]	FP11[49]	FP11[48]
		FP11[47]	FP11[46]	FP11[45]	FP11[44]	FP11[43]	FP11[42]	FP11[41]	FP11[40]
		FP11[39]	FP11[38]	FP11[37]	FP11[36]	FP11[35]	FP11[34]	FP11[33]	FP11[32]
	SFP24	FP12[31]	FP12[30]	FP12[29]	FP12[28]	FP12[27]	FP12[26]	FP12[25]	FP12[24]
		FP12[23]	FP12[22]	FP12[21]	FP12[20]	FP12[19]	FP12[18]	FP12[17]	FP12[16]
		FP12[15]	FP12[14]	FP12[13]	FP12[12]	FP12[11]	FP12[10]	FP12[9]	FP12[8]
		FP12[7]	FP12[6]	FP12[5]	FP12[4]	FP12[3]	FP12[2]	FP12[1]	FP12[0]
	SFP25	FP12[63]	FP12[62]	FP12[61]	FP12[60]	FP12[59]	FP12[58]	FP12[57]	FP12[56]
		FP12[55]	FP12[54]	FP12[53]	FP12[52]	FP12[51]	FP12[50]	FP12[49]	FP12[48]
		FP12[47]	FP12[46]	FP12[45]	FP12[44]	FP12[43]	FP12[42]	FP12[41]	FP12[40]
		FP12[39]	FP12[38]	FP12[37]	FP12[36]	FP12[35]	FP12[34]	FP12[33]	FP12[32]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
EthernetAVB*2	SFP26	FP13[31]	FP13[30]	FP13[29]	FP13[28]	FP13[27]	FP13[26]	FP13[25]	FP13[24]
		FP13[23]	FP13[22]	FP13[21]	FP13[20]	FP13[19]	FP13[18]	FP13[17]	FP13[16]
		FP13[15]	FP13[14]	FP13[13]	FP13[12]	FP13[11]	FP13[10]	FP13[9]	FP13[8]
		FP13[7]	FP13[6]	FP13[5]	FP13[4]	FP13[3]	FP13[2]	FP13[1]	FP13[0]
	SFP27	FP13[63]	FP13[62]	FP13[61]	FP13[60]	FP13[59]	FP13[58]	FP13[57]	FP13[56]
		FP13[55]	FP13[54]	FP13[53]	FP13[52]	FP13[51]	FP13[50]	FP13[49]	FP13[48]
		FP13[47]	FP13[46]	FP13[45]	FP13[44]	FP13[43]	FP13[42]	FP13[41]	FP13[40]
		FP13[39]	FP13[38]	FP13[37]	FP13[36]	FP13[35]	FP13[34]	FP13[33]	FP13[32]
	SFP28	FP14[31]	FP14[30]	FP14[29]	FP14[28]	FP14[27]	FP14[26]	FP14[25]	FP14[24]
		FP14[23]	FP14[22]	FP14[21]	FP14[20]	FP14[19]	FP14[18]	FP14[17]	FP14[16]
		FP14[15]	FP14[14]	FP14[13]	FP14[12]	FP14[11]	FP14[10]	FP14[9]	FP14[8]
		FP14[7]	FP14[6]	FP14[5]	FP14[4]	FP14[3]	FP14[2]	FP14[1]	FP14[0]
	SFP29	FP14[63]	FP14[62]	FP14[61]	FP14[60]	FP14[59]	FP14[58]	FP14[57]	FP14[56]
		FP14[55]	FP14[54]	FP14[53]	FP14[52]	FP14[51]	FP14[50]	FP14[49]	FP14[48]
		FP14[47]	FP14[46]	FP14[45]	FP14[44]	FP14[43]	FP14[42]	FP14[41]	FP14[40]
		FP14[39]	FP14[38]	FP14[37]	FP14[36]	FP14[35]	FP14[34]	FP14[33]	FP14[32]
	SFP30	FP15[31]	FP15[30]	FP15[29]	FP15[28]	FP15[27]	FP15[26]	FP15[25]	FP15[24]
		FP15[23]	FP15[22]	FP15[21]	FP15[20]	FP15[19]	FP15[18]	FP15[17]	FP15[16]
		FP15[15]	FP15[14]	FP15[13]	FP15[12]	FP15[11]	FP15[10]	FP15[9]	FP15[8]
		FP15[7]	FP15[6]	FP15[5]	FP15[4]	FP15[3]	FP15[2]	FP15[1]	FP15[0]
	SFP31	FP15[63]	FP15[62]	FP15[61]	FP15[60]	FP15[59]	FP15[58]	FP15[57]	FP15[56]
		FP15[55]	FP15[54]	FP15[53]	FP15[52]	FP15[51]	FP15[50]	FP15[49]	FP15[48]
		FP15[47]	FP15[46]	FP15[45]	FP15[44]	FP15[43]	FP15[42]	FP15[41]	FP15[40]
		FP15[39]	FP15[38]	FP15[37]	FP15[36]	FP15[35]	FP15[34]	FP15[33]	FP15[32]
	SFM0	CFM[31]	CFM[30]	CFM[29]	CFM[28]	CFM[27]	CFM[26]	CFM[25]	CFM[24]
		CFM[23]	CFM[22]	CFM[21]	CFM[20]	CFM[19]	CFM[18]	CFM[17]	CFM[16]
		CFM[15]	CFM[14]	CFM[13]	CFM[12]	CFM[11]	CFM[10]	CFM[9]	CFM[8]
		CFM[7]	CFM[6]	CFM[5]	CFM[4]	CFM[3]	CFM[2]	CFM[1]	CFM[0]
	SFM1	CFM[31]	CFM[30]	CFM[29]	CFM[28]	CFM[27]	CFM[26]	CFM[25]	CFM[24]
		CFM[23]	CFM[22]	CFM[21]	CFM[20]	CFM[19]	CFM[18]	CFM[17]	CFM[16]
		CFM[15]	CFM[14]	CFM[13]	CFM[12]	CFM[11]	CFM[10]	CFM[9]	CFM[8]
		CFM[7]	CFM[6]	CFM[5]	CFM[4]	CFM[3]	CFM[2]	CFM[1]	CFM[0]
	TGC	-	-	-	-	-	-	-	-
		-	-	TBD3[1]	TBD3[0]	-	-	TBD2[1]	TBD2[0]
		-	-	TBD1[1]	TBD1[0]	-	-	TBD0[1]	TBD0[0]
		-	-	TQP[1]	TQP[0]	TSM3	TSM2	TSM1	TSM0
	TCCR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	TFR	TFEN
		-	-	-	-	TSRQ3	TSRQ2	TSRQ1	TSRQ0
	TSR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	TFFL[2]	TFFL[1]	TFFL[0]
		-	-	-	-	CCS1[1]	CCS1[0]	CCS0[1]	CCS0[0]
	TFA0	TSV[31]	TSV[30]	TSV[29]	TSV[28]	TSV[27]	TSV[26]	TSV[25]	TSV[24]
		TSV[23]	TSV[22]	TSV[21]	TSV[20]	TSV[19]	TSV[18]	TSV[17]	TSV[16]
		TSV[15]	TSV[14]	TSV[13]	TSV[12]	TSV[11]	TSV[10]	TSV[9]	TSV[8]
		TSV[7]	TSV[6]	TSV[5]	TSV[4]	TSV[3]	TSV[2]	TSV[1]	TSV[0]
	TFA1	TSV[63]	TSV[62]	TSV[61]	TSV[60]	TSV[59]	TSV[58]	TSV[57]	TSV[56]
		TSV[55]	TSV[54]	TSV[53]	TSV[52]	TSV[51]	TSV[50]	TSV[49]	TSV[48]
		TSV[47]	TSV[46]	TSV[45]	TSV[44]	TSV[43]	TSV[42]	TSV[41]	TSV[40]
		TSV[39]	TSV[38]	TSV[37]	TSV[36]	TSV[35]	TSV[34]	TSV[33]	TSV[32]
	TFA2	-	-	-	-	-	-	TST[9]	TST[8]
		TST[7]	TST[6]	TST[5]	TST[4]	TST[3]	TST[2]	TST[1]	TST[0]
		TSV[79]	TSV[78]	TSV[77]	TSV[76]	TSV[75]	TSV[74]	TSV[73]	TSV[72]
		TSV[71]	TSV[70]	TSV[69]	TSV[68]	TSV[67]	TSV[66]	TSV[65]	TSV[64]
	CIVR0	CIV[31]	CIV[30]	CIV[29]	CIV[28]	CIV[27]	CIV[26]	CIV[25]	CIV[24]
		CIV[23]	CIV[22]	CIV[21]	CIV[20]	CIV[19]	CIV[18]	CIV[17]	CIV[16]
		CIV[15]	CIV[14]	CIV[13]	CIV[12]	CIV[11]	CIV[10]	CIV[9]	CIV[8]
		CIV[7]	CIV[6]	CIV[5]	CIV[4]	CIV[3]	CIV[2]	CIV[1]	CIV[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
EthernetAVB*2	CIVR1	CIV[31]	CIV[30]	CIV[29]	CIV[28]	CIV[27]	CIV[26]	CIV[25]	CIV[24]
		CIV[23]	CIV[22]	CIV[21]	CIV[20]	CIV[19]	CIV[18]	CIV[17]	CIV[16]
		CIV[15]	CIV[14]	CIV[13]	CIV[12]	CIV[11]	CIV[10]	CIV[9]	CIV[8]
		CIV[7]	CIV[6]	CIV[5]	CIV[4]	CIV[3]	CIV[2]	CIV[1]	CIV[0]
	CDVR0	CDV[31]	CDV[30]	CDV[29]	CDV[28]	CDV[27]	CDV[26]	CDV[25]	CDV[24]
		CDV[23]	CDV[22]	CDV[21]	CDV[20]	CDV[19]	CDV[18]	CDV[17]	CDV[16]
		CDV[15]	CDV[14]	CDV[13]	CDV[12]	CDV[11]	CDV[10]	CDV[9]	CDV[8]
		CDV[7]	CDV[6]	CDV[5]	CDV[4]	CDV[3]	CDV[2]	CDV[1]	CDV[0]
	CDVR1	CDV[31]	CDV[30]	CDV[29]	CDV[28]	CDV[27]	CDV[26]	CDV[25]	CDV[24]
		CDV[23]	CDV[22]	CDV[21]	CDV[20]	CDV[19]	CDV[18]	CDV[17]	CDV[16]
		CDV[15]	CDV[14]	CDV[13]	CDV[12]	CDV[11]	CDV[10]	CDV[9]	CDV[8]
		CDV[7]	CDV[6]	CDV[5]	CDV[4]	CDV[3]	CDV[2]	CDV[1]	CDV[0]
	CUL0	ULV[31]	ULV[30]	ULV[29]	ULV[28]	ULV[27]	ULV[26]	ULV[25]	ULV[24]
		ULV[23]	ULV[22]	ULV[21]	ULV[20]	ULV[19]	ULV[18]	ULV[17]	ULV[16]
		ULV[15]	ULV[14]	ULV[13]	ULV[12]	ULV[11]	ULV[10]	ULV[9]	ULV[8]
		ULV[7]	ULV[6]	ULV[5]	ULV[4]	ULV[3]	ULV[2]	ULV[1]	ULV[0]
	CUL1	ULV[31]	ULV[30]	ULV[29]	ULV[28]	ULV[27]	ULV[26]	ULV[25]	ULV[24]
		ULV[23]	ULV[22]	ULV[21]	ULV[20]	ULV[19]	ULV[18]	ULV[17]	ULV[16]
		ULV[15]	ULV[14]	ULV[13]	ULV[12]	ULV[11]	ULV[10]	ULV[9]	ULV[8]
		ULV[7]	ULV[6]	ULV[5]	ULV[4]	ULV[3]	ULV[2]	ULV[1]	ULV[0]
	CLL0	LLV[31]	LLV[30]	LLV[29]	LLV[28]	LLV[27]	LLV[26]	LLV[25]	LLV[24]
		LLV[23]	LLV[22]	LLV[21]	LLV[20]	LLV[19]	LLV[18]	LLV[17]	LLV[16]
		LLV[15]	LLV[14]	LLV[13]	LLV[12]	LLV[11]	LLV[10]	LLV[9]	LLV[8]
		LLV[7]	LLV[6]	LLV[5]	LLV[4]	LLV[3]	LLV[2]	LLV[1]	LLV[0]
	CLL1	LLV[31]	LLV[30]	LLV[29]	LLV[28]	LLV[27]	LLV[26]	LLV[25]	LLV[24]
		LLV[23]	LLV[22]	LLV[21]	LLV[20]	LLV[19]	LLV[18]	LLV[17]	LLV[16]
		LLV[15]	LLV[14]	LLV[13]	LLV[12]	LLV[11]	LLV[10]	LLV[9]	LLV[8]
		LLV[7]	LLV[6]	LLV[5]	LLV[4]	LLV[3]	LLV[2]	LLV[1]	LLV[0]
	DIC	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DPE15	DPE14	DPE13	DPE12	DPE11	DPE10	DPE9	DPE8
		DPE7	DPE6	DPE5	DPE4	DPE3	DPE2	DPE1	-
	DIS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		DPF15	DPF14	DPF13	DPF12	DPF11	DPF10	DPF9	DPF8
		DPF7	DPF6	DPF5	DPF4	DPF3	DPF2	DPF1	-
	EIC	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	TFFE
		CULE1	CULE0	CLLE1	CLLE0	SEE	QEE	-	-
	EIS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	QFS
		-	-	-	-	-	-	-	TFFF
		CULF1	CULF0	CLLF1	CLLF0	SEF	QEF	-	-
	RIC0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	FRE17	FRE16
		FRE15	FRE14	FRE13	FRE12	FRE11	FRE10	FRE9	FRE8
		FRE7	FRE6	FRE5	FRE4	FRE3	FRE2	FRE1	FRE0
	RIS0	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	FRF17	FRF16
		FRF15	FRF14	FRF13	FRF12	FRF11	FRF10	FRF9	FRF8
		FRF7	FRF6	FRF5	FRF4	FRF3	FRF2	FRF1	FRF0
	RIC1	RFWE	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
	RIS1	RFWF	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
EthernetAVB*2	RIC2	RFFE	-	-	-	-	-	-	-
		-	-	-	-	-	-	QFE17	QFE16
		QFE15	QFE14	QFE13	QFE12	QFE11	QFE10	QFE9	QFE8
		QFE7	QFE6	QFE5	QFE4	QFE3	QFE2	QFE1	QFE0
	RIS2	RFFF	-	-	-	-	-	-	-
		-	-	-	-	-	-	QFF17	QFF16
		QFF15	QFF14	QFF13	QFF12	QFF11	QFF10	QFF9	QFF8
		QFF7	QFF6	QFF5	QFF4	QFF3	QFF2	QFF1	QFF0
	TIC	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	TFWE	TFUE
		-	-	-	-	-	-	-	-
	TIS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	TFWF	TFUF
		-	-	-	-	-	-	-	-
	ISS	DPS15	DPS14	DPS13	DPS12	DPS11	DPS10	DPS9	DPS8
		DPS7	DPS6	DPS5	DPS4	DPS3	DPS2	DPS1	-
		-	-	CGIS	RFWS	-	-	TFWS	TFUS
		MS	ES	-	-	-	-	-	-
	GCCR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	TCSS[1]	TCSS[0]
		-	-	LMTT	LPTC	LTI	LTO	TCR[1]	TCR[0]
	GMTT	MTTV[31]	MTTV[30]	MTTV[29]	MTTV[28]	MTTV[27]	MTTV[26]	MTTV[25]	MTTV[24]
		MTTV[23]	MTTV[22]	MTTV[21]	MTTV[20]	MTTV[19]	MTTV[18]	MTTV[17]	MTTV[16]
		MTTV[15]	MTTV[14]	MTTV[13]	MTTV[12]	MTTV[11]	MTTV[10]	MTTV[9]	MTTV[8]
		MTTV[7]	MTTV[6]	MTTV[5]	MTTV[4]	MTTV[3]	MTTV[2]	MTTV[1]	MTTV[0]
	GPTC	PTCV[31]	PTCV[30]	PTCV[29]	PTCV[28]	PTCV[27]	PTCV[26]	PTCV[25]	PTCV[24]
		PTCV[23]	PTCV[22]	PTCV[21]	PTCV[20]	PTCV[19]	PTCV[18]	PTCV[17]	PTCV[16]
		PTCV[15]	PTCV[14]	PTCV[13]	PTCV[12]	PTCV[11]	PTCV[10]	PTCV[9]	PTCV[8]
		PTCV[7]	PTCV[6]	PTCV[5]	PTCV[4]	PTCV[3]	PTCV[2]	PTCV[1]	PTCV[0]
	GTI	-	-	-	-	TIV[27]	TIV[26]	TIV[25]	TIV[24]
		TIV[23]	TIV[22]	TIV[21]	TIV[20]	TIV[19]	TIV[18]	TIV[17]	TIV[16]
		TIV[15]	TIV[14]	TIV[13]	TIV[12]	TIV[11]	TIV[10]	TIV[9]	TIV[8]
		TIV[7]	TIV[6]	TIV[5]	TIV[4]	TIV[3]	TIV[2]	TIV[1]	TIV[0]
	GTO0	TOV[31]	TOV[30]	TOV[29]	TOV[28]	TOV[27]	TOV[26]	TOV[25]	TOV[24]
		TOV[23]	TOV[22]	TOV[21]	TOV[20]	TOV[19]	TOV[18]	TOV[17]	TOV[16]
		TOV[15]	TOV[14]	TOV[13]	TOV[12]	TOV[11]	TOV[10]	TOV[9]	TOV[8]
		TOV[7]	TOV[6]	TOV[5]	TOV[4]	TOV[3]	TOV[2]	TOV[1]	TOV[0]
	GTO1	TOV[63]	TOV[62]	TOV[61]	TOV[60]	TOV[59]	TOV[58]	TOV[57]	TOV[56]
		TOV[55]	TOV[54]	TOV[53]	TOV[52]	TOV[51]	TOV[50]	TOV[49]	TOV[48]
		TOV[47]	TOV[46]	TOV[45]	TOV[44]	TOV[43]	TOV[42]	TOV[41]	TOV[40]
		TOV[39]	TOV[38]	TOV[37]	TOV[36]	TOV[35]	TOV[34]	TOV[33]	TOV[32]
	GTO2	TOV[95]	TOV[94]	TOV[93]	TOV[92]	TOV[91]	TOV[90]	TOV[89]	TOV[88]
		TOV[87]	TOV[86]	TOV[85]	TOV[84]	TOV[83]	TOV[82]	TOV[81]	TOV[80]
		TOV[79]	TOV[78]	TOV[77]	TOV[76]	TOV[75]	TOV[74]	TOV[73]	TOV[72]
		TOV[71]	TOV[70]	TOV[69]	TOV[68]	TOV[67]	TOV[66]	TOV[65]	TOV[64]
	GIC	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	PTME	PTOE	PTCE
	GIS	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	PTMF	PTOF	PTCF
	GCPT	CPTV[31]	CPTV[30]	CPTV[29]	CPTV[28]	CPTV[27]	CPTV[26]	CPTV[25]	CPTV[24]
		CPTV[23]	CPTV[22]	CPTV[21]	CPTV[20]	CPTV[19]	CPTV[18]	CPTV[17]	CPTV[16]
		CPTV[15]	CPTV[14]	CPTV[13]	CPTV[12]	CPTV[11]	CPTV[10]	CPTV[9]	CPTV[8]
		CPTV[7]	CPTV[6]	CPTV[5]	CPTV[4]	CPTV[3]	CPTV[2]	CPTV[1]	CPTV[0]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
EthernetAVB*2	GCT0	CTV[31]	CTV[30]	CTV[29]	CTV[28]	CTV[27]	CTV[26]	CTV[25]	CTV[24]
		CTV[23]	CTV[22]	CTV[21]	CTV[20]	CTV[19]	CTV[18]	CTV[17]	CTV[16]
		CTV[15]	CTV[14]	CTV[13]	CTV[12]	CTV[11]	CTV[10]	CTV[9]	CTV[8]
		CTV[7]	CTV[6]	CTV[5]	CTV[4]	CTV[3]	CTV[2]	CTV[1]	CTV[0]
	GCT1	CTV[63]	CTV[62]	CTV[61]	CTV[60]	CTV[59]	CTV[58]	CTV[57]	CTV[56]
		CTV[55]	CTV[54]	CTV[53]	CTV[52]	CTV[51]	CTV[50]	CTV[49]	CTV[48]
		CTV[47]	CTV[46]	CTV[45]	CTV[44]	CTV[43]	CTV[42]	CTV[41]	CTV[40]
		CTV[39]	CTV[38]	CTV[37]	CTV[36]	CTV[35]	CTV[34]	CTV[33]	CTV[32]
	GCT2	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		CTV[79]	CTV[78]	CTV[77]	CTV[76]	CTV[75]	CTV[74]	CTV[73]	CTV[72]
		CTV[71]	CTV[70]	CTV[69]	CTV[68]	CTV[67]	CTV[66]	CTV[65]	CTV[64]
	GCEC	-	-	-	-	-	DRC[10]	DRC[9]	DRC[8]
		DRC[7]	DRC[6]	DRC[5]	DRC[4]	DRC[3]	DRC[2]	DRC[1]	DRC[0]
		-	-	-	-	-	-	SCS[1]	SCS[0]
		-	-	-	DEN	-	-	-	CES
	ECMR	-	-	-	-	-	TRCCM	-	-
		RCSC	-	DPAD	RZPF	ZPF	PFR	RXF	TXF
		-	-	-	-	-	-	-	-
		-	RE	TE	-	-	-	DM	PRM
	RFLR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	RFL[17]	RFL[16]
		RFL[15]	RFL[14]	RFL[13]	RFL[12]	RFL[11]	RFL[10]	RFL[9]	RFL[8]
		RFL[7]	RFL[6]	RFL[5]	RFL[4]	RFL[3]	RFL[2]	RFL[1]	RFL[0]
	ECSR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	PFROI	-	-	-	ICD
	ECSIPR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	PFROIP	-	-	-	ICDIP
	PIR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		-	-	-	-	MDI	MDO	MMID	MDC
	APR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		AP[15]	AP[14]	AP[13]	AP[12]	AP[11]	AP[10]	AP[9]	AP[8]
		AP[7]	AP[6]	AP[5]	AP[4]	AP[3]	AP[2]	AP[1]	AP[0]
	MPR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		MP[15]	MP[14]	MP[13]	MP[12]	MP[11]	MP[10]	MP[9]	MP[8]
		MP[7]	MP[6]	MP[5]	MP[4]	MP[3]	MP[2]	MP[1]	MP[0]
	PFTCR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		PFTXC[15]	PFTXC[14]	PFTXC[13]	PFTXC[12]	PFTXC[11]	PFTXC[10]	PFTXC[9]	PFTXC[8]
		PFTXC[7]	PFTXC[6]	PFTXC[5]	PFTXC[4]	PFTXC[3]	PFTXC[2]	PFTXC[1]	PFTXC[0]
	PFRCR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		PFRXC[15]	PFRXC[14]	PFRXC[13]	PFRXC[12]	PFRXC[11]	PFRXC[10]	PFRXC[9]	PFRXC[8]
		PFRXC[7]	PFRXC[6]	PFRXC[5]	PFRXC[4]	PFRXC[3]	PFRXC[2]	PFRXC[1]	PFRXC[0]
	TPAUSER	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		TPAUSE[15]	TPAUSE[14]	TPAUSE[13]	TPAUSE[12]	TPAUSE[11]	TPAUSE[10]	TPAUSE[9]	TPAUSE[8]
		TPAUSE[7]	TPAUSE[6]	TPAUSE[5]	TPAUSE[4]	TPAUSE[3]	TPAUSE[2]	TPAUSE[1]	TPAUSE[0]
	MAHR	MA[47]	MA[46]	MA[45]	MA[44]	MA[43]	MA[42]	MA[41]	MA[40]
		MA[39]	MA[38]	MA[37]	MA[36]	MA[35]	MA[34]	MA[33]	MA[32]
		MA[31]	MA[30]	MA[29]	MA[28]	MA[27]	MA[26]	MA[25]	MA[24]
		MA[23]	MA[22]	MA[21]	MA[20]	MA[19]	MA[18]	MA[17]	MA[16]

chanhvinh.com

Table 46.2 Register Bits

Module	Register Abbreviation	Bits 31/23/15/7	Bits 30/22/14/6	Bits 29/21/13/5	Bits 28/20/12/4	Bits 27/19/11/3	Bits 26/18/10/2	Bits 25/17/9/1	Bits 24/16/8/0
EthernetAVB*2	MALR	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
		MA[15]	MA[14]	MA[13]	MA[12]	MA[11]	MA[10]	MA[9]	MA[8]
		MA[7]	MA[6]	MA[5]	MA[4]	MA[3]	MA[2]	MA[1]	MA[0]
	CEFCR	CEFC[31]	CEFC[30]	CEFC[29]	CEFC[28]	CEFC[27]	CEFC[26]	CEFC[25]	CEFC[24]
		CEFC[23]	CEFC[22]	CEFC[21]	CEFC[20]	CEFC[19]	CEFC[18]	CEFC[17]	CEFC[16]
		CEFC[15]	CEFC[14]	CEFC[13]	CEFC[12]	CEFC[11]	CEFC[10]	CEFC[9]	CEFC[8]
		CEFC[7]	CEFC[6]	CEFC[5]	CEFC[4]	CEFC[3]	CEFC[2]	CEFC[1]	CEFC[0]
	FRECR	FREC[31]	FREC[30]	FREC[29]	FREC[28]	FREC[27]	FREC[26]	FREC[25]	FREC[24]
		FREC[23]	FREC[22]	FREC[21]	FREC[20]	FREC[19]	FREC[18]	FREC[17]	FREC[16]
		FREC[15]	FREC[14]	FREC[13]	FREC[12]	FREC[11]	FREC[10]	FREC[9]	FREC[8]
		FREC[7]	FREC[6]	FREC[5]	FREC[4]	FREC[3]	FREC[2]	FREC[1]	FREC[0]
	TSFRCR	TSFRC[31]	TSFRC[30]	TSFRC[29]	TSFRC[28]	TSFRC[27]	TSFRC[26]	TSFRC[25]	TSFRC[24]
		TSFRC[23]	TSFRC[22]	TSFRC[21]	TSFRC[20]	TSFRC[19]	TSFRC[18]	TSFRC[17]	TSFRC[16]
		TSFRC[15]	TSFRC[14]	TSFRC[13]	TSFRC[12]	TSFRC[11]	TSFRC[10]	TSFRC[9]	TSFRC[8]
		TSFRC[7]	TSFRC[6]	TSFRC[5]	TSFRC[4]	TSFRC[3]	TSFRC[2]	TSFRC[1]	TSFRC[0]
	TLFRCR	TLFC[31]	TLFC[30]	TLFC[29]	TLFC[28]	TLFC[27]	TLFC[26]	TLFC[25]	TLFC[24]
		TLFC[23]	TLFC[22]	TLFC[21]	TLFC[20]	TLFC[19]	TLFC[18]	TLFC[17]	TLFC[16]
		TLFC[15]	TLFC[14]	TLFC[13]	TLFC[12]	TLFC[11]	TLFC[10]	TLFC[9]	TLFC[8]
		TLFC[7]	TLFC[6]	TLFC[5]	TLFC[4]	TLFC[3]	TLFC[2]	TLFC[1]	TLFC[0]
	RFCR	RFC[31]	RFC[30]	RFC[29]	RFC[28]	RFC[27]	RFC[26]	RFC[25]	RFC[24]
		RFC[23]	RFC[22]	RFC[21]	RFC[20]	RFC[19]	RFC[18]	RFC[17]	RFC[16]
		RFC[15]	RFC[14]	RFC[13]	RFC[12]	RFC[11]	RFC[10]	RFC[9]	RFC[8]
		RFC[7]	RFC[6]	RFC[5]	RFC[4]	RFC[3]	RFC[2]	RFC[1]	RFC[0]
	MAFCR	MAFC[31]	MAFC[30]	MAFC[29]	MAFC[28]	MAFC[27]	MAFC[26]	MAFC[25]	MAFC[24]
		MAFC[23]	MAFC[22]	MAFC[21]	MAFC[20]	MAFC[19]	MAFC[18]	MAFC[17]	MAFC[16]
		MAFC[15]	MAFC[14]	MAFC[13]	MAFC[12]	MAFC[11]	MAFC[10]	MAFC[9]	MAFC[8]
		MAFC[7]	MAFC[6]	MAFC[5]	MAFC[4]	MAFC[3]	MAFC[2]	MAFC[1]	MAFC[0]

Note 1. RZ/A1L only
Note 2. RZ/A1LU only
Note 3. RZ/A1L only. This bit is reserved in the RZ/A1LU and RZ/A1LC.
Note 4. RZ/A1LU only. This bit is reserved in the RZ/A1L and RZ/A1LC.
Note 5. RZ/A1L and RZ/A1LU only. This bit is reserved in the RZ/A1LC.
Note 6. RZ/A1LU and RZ/A1LC only. This bit is reserved in the RZ/A1L.

chanhvinh.com

46.3 Register States

Table 46.3 Register States

Module	Register Name	Power-On Reset	Deep Standby	Software Standby	Module Standby	Sleep
Secondary cache	All registers	Initialized	Initialized	Retained	-	Retained
LSI internal bus	All registers	Initialized	Initialized	Retained	-	Retained
Clock pulse generator	FRQCR	Initialized*1	Initialized	Retained	-	Retained
Interrupt controller	All registers	Initialized	Initialized	Retained	-	Retained
Bus state controller	RTCSR	Initialized	Initialized	Retained	-	Retained*2
	RTCNT	Initialized	Initialized	Retained	-	Retained*3
	All registers other than above	Initialized	Initialized	Retained	-	Retained
Direct memory access controller	All registers	Initialized	Initialized	Retained	-	Retained*4
Multi-function timer pulse unit 2	All registers	Initialized	Initialized	Initialized	Retained	Retained
OS timer	All registers	Initialized	Initialized	Retained	Retained	Retained
Watchdog timer	WRCSR	Initialized*1	Initialized	Retained	-	Retained
	All registers other than above	Initialized	Initialized	Retained	-	Retained
Realtime clock	R64CNT	Retained*3	Retained*3	Retained*3	Retained	Retained*3
	RSECCNT					
	RMINCNT					
	RHRCNT					
	RWKCNT					
	RDAYCNT					
	RMONCNT					
	RYRCNT					
	RSECAR	Retained	Retained	Retained	Retained	Retained
	RMINAR					
	RHRAR					
	RWKAR					
	RDAYAR					
	RMONAR					
	RYRAR					
	RCR1	Initialized	Initialized	Retained	Retained	Retained
	RCR2	Initialized	Initialized*5	Retained	Retained	Retained
	RCR3	Retained	Retained	Retained	Retained	Retained
	RCR5	Retained	Retained	Retained	Retained	Retained
	RFRH	Retained	Retained	Retained	Retained	Retained
	RFRL	Retained	Retained	Retained	Retained	Retained
Serial communication interface with FIFO	All registers	Initialized	Initialized	Retained	Retained	Retained
Serial communications interface	All registers	Initialized	Initialized	Retained	Retained	Retained
Renesas serial peripheral interface	All registers	Initialized	Initialized	Retained	Retained	Retained
SPI multi I/O bus controller	All registers	Initialized	Initialized	Retained	Retained	Retained
I^2C bus interface	All registers	Initialized	Initialized	Retained	Retained	Retained
Serial sound interface	All registers	Initialized	Initialized	Retained	Retained	Retained
Media local bus*11	All registers	Initialized	Initialized	Retained	Retained	Retained
CAN interface	All registers	Initialized	Initialized	Retained	Retained	Retained
IEBus controller*11	All registers	Initialized	Initialized	Retained	Retained	Retained
Renesas SPDIF interface	All registers	Initialized	Initialized	Retained	Retained	Retained
CD-ROM decoder*11	All registers	Initialized	Initialized	Retained	Retained	Retained
LIN interface*11	All registers	Initialized	Initialized	Retained	Retained	Retained

chanhvinh.com

Table 46.3 Register States

Module	Register Name	Power-On Reset	Deep Standby	Software Standby	Module Standby	Sleep
Ethernet controller	All registers	Initialized	Initialized	Retained	Retained	Retained
EthernetAVB*12	All registers	Initialized	Initialized	Retained	Retained	Retained
A/D converter	All registers	Initialized	Initialized	Initialized	Initialized	Retained
USB2.0 host/function module	All registers	Initialized	Initialized	Retained	Retained	Retained
Video display controller 5	All registers	Initialized	Initialized	Retained	Retained	Retained
JPEG codec unit*12	All registers	Initialized	Initialized	Retained	Retained	Retained
Capture engine unit	All registers	Initialized	Initialized	Retained	Retained	Retained
SCUX	All registers	Initialized	Initialized	Retained	Retained	Retained
SD host interface	All registers	Initialized	Initialized	Retained	Retained	Retained
MMC host interface	All registers	Initialized	Initialized	Retained	Retained	Retained
Ports	All registers*10	Initialized	Initialized	Retained	-	Retained
Power-down modes	DSFR	Initialized	Retained	Retained	-	Retained
	XTALCTR	Initialized*1	Retained*6	Retained*6	-	Retained
	All registers other than above	Initialized	Initialized	Retained	-	Retained
Debugger interface	Registers in CA9 PMU*7	Initialized	Initialized	Retained	-	Retained
	All registers other than above*8	Retained	Initialized*9	Retained	Retained	Retained

Note 1. Retained on internal power-on reset by the watchdog timer.
Note 2. Flag processing continues.
Note 3. Counting up continues.
Note 4. Transfer can proceed.
Note 5. The value of RTCEN bit is retained.
Note 6. The GAIN0 bit is initialized when the realtime clock is not using the EXTAL pin.
Note 7. Access to these registers from the I/O area (SLV6) is not possible while $\overline{\text{TRST}}$ is asserted.
Note 8. Initialized on assertion of $\overline{\text{TRST}}$.
Note 9. Retained in FAKE debug mode.
Note 10. An internal power-on reset by the watchdog timer does not initialize PM6[0], PMC6[0], PFC6[0], PFCE6[0], and PFCAE6[0].
Note 11. RZ/A1L only
Note 12. RZ/A1LU only

chanhvinh.com

47. Electrical Characteristics

47.1 Absolute Maximum Ratings

Table 47.1 Absolute Maximum Ratings

Item		Symbol	Value	Unit
Power supply voltage (I/O)		PV_{CC}	–0.3 to 4.2	V
Power supply voltage (Internal)		V_{CC}	–0.3 to 1.6	V
PLL power supply voltage		$PLLV_{CC}$	–0.3 to 1.6	V
Analog power supply voltage		AV_{CC}	–0.3 to 4.2	V
Analog reference voltage		AVref	–0.3 to AV_{CC} +0.3	V
USB transceiver analog power supply voltage (I/O)		$USBAPV_{CC}$	–0.3 to 4.2	V
USB transceiver digital power supply voltage (I/O) Note: Products in BGA packages do not have this pin.		$USBDPV_{CC}$	–0.3 to 4.2	V
USB transceiver analog power supply voltage (internal)		$USBAV_{CC}$	–0.3 to 1.6	V
USB transceiver digital power supply voltage (internal) Note: Products in BGA packages do not have this pin.		$USBDV_{CC}$	–0.3 to 1.6	V
Power supply for USB 480 MHz (internal) Note: Products in BGA packages do not have this pin.		$USBUV_{CC}$	–0.3 to 1.6	V
Input voltage	VBUS	V_{in}	–0.3 to 5.5	V
	Other input pins	V_{in}	–0.3- to 3.3-V power supply (PV_{CC}, AV_{CC}, $USBAPV_{CC}$, $USBDPV_{CC}$) +0.3	V
Operating temperature		T_{opr}	–40 to +85	°C
Storage temperature		T_{stg}	–55 to +125	°C

Caution: Permanent damage to the LSI may result if absolute maximum ratings are exceeded.

47.2 Power-On/Power-Off Sequence

The 1.2-V power supply (V_{CC}, PLLVcc, $USBAV_{CC}$, $USBDV_{CC}$, and $USBUV_{CC}$) and 3.3-V power supply (PV_{CC}, AV_{CC}, $USBAPV_{CC}$, and $USBDPV_{CC}$) can be turned on and off in any order.

When turning on the power, be sure to drive both the $\overline{TRST}$ and $\overline{RES}$ pins low; otherwise, the output pins and input/output pins output undefined levels, resulting in system malfunction.

When turning off the power, drive the $\overline{TRST}$ and $\overline{RES}$ pins low if the undefined output may cause a problem.

chanhvinh.com

47.3 DC Characteristics

- Conditions used to obtain DC characteristics (2) in Table 47.2 other than current consumption
  V_{CC} = USBDV_{CC}= USBUV_{CC} = 1.10 to 1.26 V, PV_{CC} = USBDPV_{CC} = 3.0 to 3.6 V, PLLV_{CC} = 1.10 to 1.26 V,
  AV_{CC} = 3.0 to 3.6 V, USBAPV_{CC} = 3.0 to 3.6 V, USBAV_{CC} = 1.10 to 1.26 V,
  V_{SS} = AV_{SS} = USBDV_{SS} = USBAV_{SS} = USBDPV_{SS} = USBAPV_{SS} = USBUV_{SS} = 0 V,
  Ta = –40 to 85 °C

- Conditions used to obtain DC characteristics (2) in Table 47.2 for current consumption
  V_{CC} = USBDV_{CC} = USBUV_{CC} = 1.18 V, PV_{CC} = USBDPV_{CC} = 3.3 V, PLLV_{CC} = 1.18 V, AV_{CC} = 3.3 V,
  USBAPV_{CC} = 3.3 V, USBAV_{CC} = 1.18 V,
  V_{SS} = AV_{SS} = USBDV_{SS} = USBAV_{SS} = USBDPV_{SS} = USBAPV_{SS} = USBUV_{SS} = 0 V, AVref = 3.3 V, VBUS = 5.0 V,
  Ta = –40 to 85 °C
  Iϕ = 400.00 MHz, Bϕ = 133.33 MHz, P1ϕ = 66.67 MHz, P0ϕ = 33.33 MHz

Note: Products in BGA packages do not have USBDV_{CC}, USBUV_{CC}, USBDPV_{CC}, USBDV_{SS}, USBAV_{SS}, USBDPV_{SS}, USBAPV_{SS}, and USBUV_{SS} pins.

Table 47.2 DC Characteristics (1) [Common Items]

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power supply voltage		PV_{CC}	3.0	3.3	3.6	V	
		V_{CC}	1.10	1.18	1.26	V	
PLL power supply voltage		PLLV_{CC}	1.10	1.18	1.26	V	
Analog power supply voltage		AV_{CC}	3.0	3.3	3.6	V	
USB power supply voltage Note: Products in BGA packages do not have USBDPVcc, USBDVcc, and USBUVcc pins.		USBAPV_{CC} USBDPV_{CC}	3.0	3.3	3.6	V	
		USBAV_{CC} USBDV_{CC} USBUV_{CC}	1.10	1.18	1.26	V	
Input leakage current	All input pins	$\|I_{in}\|$	—	—	1.0	µA	V_{in} = 0.5 to PV_{CC} – 0.5 V
Three-state leakage current	All input/output pins, all output pins (except P1_0 to P1_7) (off state)	$\|I_{STI}\|$	—	—	1.0	µA	V_{in} = 0.5 to PV_{CC} – 0.5 V
	P1_0 to P1_7		—	—	10	µA	
Input capacitance	all input/output pins, all input pins	Cin	—	—	10	pF	

chanhvinh.com

Table 47.2 DC Characteristics (2) [Current Consumption]

Item		Power Supply	Symbol	Typ.	Max.	Unit	Test Conditions
Current consumption in normal operation		V_{CC}	I_{CC}	—	492	mA	
		$PLLV_{CC}$	$PLLI_{CC}$	—	8	mA	
		PV_{CC}	PI_{CC}	70*1*2	—	mA	
		AV_{CC}	AI_{CC}	—	4	mA	During A/D conversion
		AVref	AIref	—	1	mA	During A/D conversion
		$USBAV_{CC}$	UAI_{CC}	—	6	mA	When the USB host/function is in use.
		$USBDV_{CC}$ + $USBUV_{CC}$	UDI_{CC}*3	—	20	mA	In USB high-speed operation (2ch)
		$USBAPV_{CC}$	$UAPI_{CC}$	—	4	mA	When the USB host/function is in use.
		$USBDPV_{CC}$	$UDPI_{CC}$*4	70*1*2	—	mA	In USB high-speed operation (2ch)
		VBUS	VI_{CC}	—	10	µA	
Current consumption in sleep mode		V_{CC}	Isleep	—	300	mA	
		For the other power supply, the current consumption is the same as in normal operation.					
Current consumption in software standby mode	Ta > 50 °C	V_{CC} + $PLLV_{CC}$ + $USBAV_{CC}$ + $USBDV_{CC}$ + $USBUV_{CC}$	Isstby	32	120	mA	
		PV_{CC} + AV_{CC} + AVref + $USBAPV_{CC}$ + $USBDPV_{CC}$	PIsstby	20	26	µA	
		VBUS	VIsstby	8	10	µA	
	Ta ≤ 50 °C	V_{CC} + $PLLV_{CC}$ + $USBAV_{CC}$ + $USBDV_{CC}$ + $USBUV_{CC}$	Isstby	13	55	mA	
		PV_{CC} + AV_{CC} + AVref + $USBAPV_{CC}$ + $USBDPV_{CC}$	PIsstby	17	20	µA	
		VBUS	VIsstby	8	10	µA	

chanhvinh.com

Table 47.2 DC Characteristics (2) [Current Consumption]

Item		Power Supply	Symbol	Typ.	Max.	Unit	Test Conditions
Current consumption in deep standby mode	Ta > 50 °C	V_{CC} + $PLLV_{CC}$ + $USBAV_{CC}$ + $USBDV_{CC}$ + $USBUV_{CC}$	Idstby	27	119	µA	RAM 0 Kbytes retained, RTC_X1 selected
				32	137	µA	RAM 16 Kbytes retained, RTC_X1 selected
				37	155	µA	RAM 32 Kbytes retained, RTC_X1 selected
				42	191	µA	RAM 64 Kbytes retained, RTC_X1 selected
				62	263	µA	RAM 128 Kbytes retained, RTC_X1 selected
				When the EXTAL 13 MHz is selected, 5 µA and 7 µA are added to the "Typ." and "Max." values above, respectively.			
		PV_{CC} + AV_{CC} + AVref + $USBAPV_{CC}$ + $USBDPV_{CC}$	PIdstby	13	15	µA	RTC is not operating
				20	25	µA	RTC_X1 selected
				1	—	mA	EXTAL 13 MHz selected, small gain*1
		VBUS	VIdstby	8	10	µA	
	Ta ≤ 50 °C	V_{CC} + $PLLV_{CC}$ + $USBAV_{CC}$ + $USBDV_{CC}$+ $USBUV_{CC}$	Idstby	17	54	µA	RAM 0 Kbytes retained, RTC_X1 selected
				21	71	µA	RAM 16 Kbytes retained, RTC_X1 selected
				25	88	µA	RAM 32 Kbytes retained, RTC_X1 selected
				33	122	µA	RAM 64 Kbytes retained, RTC_X1 selected
				49	190	µA	RAM 128 Kbytes retained, RTC_X1 selected
				When the EXTAL 13 MHz is selected, 5 µA and 7 µA are added to the "Typ." and "Max." values above, respectively.			
		PV_{CC} + AV_{CC} + AVref + $USBAPV_{CC}$ + $USBDPV_{CC}$	PIdstby	12	13	µA	RTC is not operating
				19	23	µA	RTC_X1 selected
				1	—	mA	EXTAL 13 MHz selected, small gain*1
		VBUS	VIdstby	8	10	µA	

Note 1. Reference value. The actual operating current greatly depends on the system (such as slow rising/falling edges caused by IO load and toggle frequency). Be sure to determine the value using the actual system.
Note 2. The sum of PIcc and UDPIcc must be equal to or lower than 150 mA.
Note 3. In the products in BGA packages, UDIcc is added to Icc.
Note 4. In the products in BGA packages, UDPIcc is added to PIcc.

chanhvinh.com

Table 47.2 DC Characteristics (3) [Except I²C Bus Interface, and USB 2.0 Host/Function Module-Related Pins]

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input high voltage*		V_{IH}	2.2	—	PV_{CC} + 0.3	V	
Input low voltage*		V_{IL}	–0.3	—	0.8	V	
Schmitt trigger input characteristics		VT^+	$PV_{CC} \times 0.665$	—	—	V	
		VT^-	—	—	0.8	V	
		$VT^+ - VT^-$	0.2	—	—	V	
Output high voltage		V_{OH}	PV_{CC} – 0.5	—	—	V	I_{OH} = –2.0 mA
Output low voltage		V_{OL}	—	—	0.4	V	I_{OL} = 2.0 mA
RAM standby voltage	Software standby mode (large-capacity on-chip RAM)	V_{RAMS}	0.85	—	—	V	Measured with V_{CC} as parameter
	Deep standby mode (only the on-chip RAM for data retention)	V_{RAMD}	1.10	—	—	V	

Note: * Values for the input of data for boundary scanning through pins TMS, TCK, JP0_0, JP0_1, P5_0 to P5_15, and P6_0 to P6_15.

Table 47.2 DC Characteristics (4) [I²C Bus Interface Related Pins*]

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input high voltage	V_{IH}	$PV_{CC} \times 0.7$	—	PV_{CC} + 0.3	V	
Input low voltage	V_{IL}	–0.3	—	$PV_{CC} \times 0.3$	V	
Schmitt trigger input characteristics	$V_{IH} - V_{IL}$	$PV_{CC} \times 0.05$	—	—	V	
Output low voltage	V_{OL}	—	—	0.4	V	I_{OL} = 3.0 mA

Note: * The P1_0 to P1_7 pins are open-drain pins.

Table 47.2 DC Characteristics (5) [USB 2.0 Host/Function Module-Related Pins*]

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Reference resistance	R_{REF}	5.6 kΩ ± 1%	5.6 kΩ ± 1%	5.6 kΩ ± 1%		
Input high voltage (VBUS1, VBUS0)	V_{IH}	4.02	—	5.25	V	
Input low voltage (VBUS1, VBUS0)	V_{IL}	–0.3	—	0.5	V	
Input high voltage (USB_X1)	V_{IH}	PV_{CC} – 0.5	—	PV_{CC} + 0.3	V	
Input low voltage (USB_X1)	V_{IL}	–0.3	—	0.5	V	

Note: * REFRIN, VBUS1, VBUS0, USB_X1, and USB_X2 pins

chanhvinh.com

Table 47.2 DC Characteristics (6) [USB 2.0 Host/Function Module-Related Pins* (Low-Speed, Full-Speed, and High-Speed Common Items)]

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
DP pull-up resistance (when function is selected)	R_{pu}	0.900	—	1.575	kΩ	In idle mode
		1.425	—	3.090	kΩ	In transmit/receive mode
DP and DM pull-down resistance (when host is selected)	R_{pd}	14.25	—	24.80	kΩ	

Note: * DP1, DP0, DM1, and DM pins

Table 47.2 DC Characteristics (7) [USB 2.0 Host/Function Module-Related Pins* (Low-Speed and Full-Speed)]

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input high voltage	V_{IH}	2.0	—	—	V	
Input low voltage	V_{IL}	—	—	0.8	V	
Differential input sensitivity	V_{DI}	0.2	—	—	V	| (DP) – (DM) |
Differential common mode range	V_{CM}	0.8	—	2.5	V	
Output high voltage	V_{OH}	2.8	—	3.6	V	I_{OH} = –200 µA
Output low voltage	V_{OL}	0.0	—	0.3	V	I_{OL} = 2 mA
Output signal crossover voltage	V_{CRS}	1.3	—	2.0	V	C_L = 50 pF (full-speed) C_L = 200 to 600 pF (low-speed)

Note: * DP1, DP0, DM1, and DM pins

Table 47.2 DC Characteristics (8) [USB 2.0 Host/Function Module-Related Pins* (High-Speed)]

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Squelch detection threshold voltage (differential voltage)	V_{HSSQ}	100	—	150	mV	
Common mode voltage range	V_{HSCM}	–50	—	500	mV	
Idle state	V_{HSOI}	–10.0	—	10.0	mV	
Output high voltage	V_{HSOH}	360	—	440	mV	
Output low voltage	V_{HSOL}	–10.0	—	10.0	mV	
Chirp J output voltage (difference)	V_{CHIRPJ}	700	—	1100	mV	
Chirp K output voltage (difference)	V_{CHIRPK}	–900	—	–500	mV	

Note: * DP1, DP0, DM1, and DM pins

Table 47.3 Permissible Output Currents

Item		Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	P1_0 to P1_7	I_{OL}	—	—	10	mA
	Output pins other than above		—		2	mA
Permissible output high current (per pin)		$-I_{OH}$	—	—	2	mA
Permissible output current (total)		ΣI_O	—	—	150	mA

Caution: To protect the LSI's reliability, do not exceed the output current values in Table 47.3.

chanhvinh.com

47.4 AC Characteristics

Signals input to this LSI are basically handled as signals in synchronization with a clock. The setup and hold times for input pins must be followed.

Conditions for AC characteristics: V_{CC} = $USBDV_{CC}$ = $USBUV_{CC}$ = 1.10 to 1.26 V, PV_{CC} = $USBDPV_{CC}$ = 3.0 to 3.6 V, $PLLV_{CC}$ = 1.10 to 1.26 V, AV_{CC} = 3.0 to 3.6 V, $USBAPV_{CC}$ = 3.0 to 3.6 V, $USBAV_{CC}$ = 1.10 to 1.26 V, V_{SS} = AV_{SS} = $USBDV_{SS}$ = $USBAV_{SS}$ = $USBDPV_{SS}$ = $USBAPV_{SS}$ = $USBUV_{SS}$ = 0 V, Ta = –40 to 85 °C

Note: Products in BGA packages do not have $USBDV_{CC}$, $USBUV_{CC}$, $USBDPV_{CC}$, $USBDV_{SS}$, $USBAV_{SS}$, $USBDPV_{SS}$, $USBAPV_{SS}$, and $USBUV_{SS}$ pins.

Table 47.4 Operating Frequency

Item		Symbol	Min.	Max.	Unit	Remarks
Operating frequency	CPU clock (Iϕ)	f	100.00	400.00	MHz	
	Internal bus clock (Bϕ)		100.00	133.33	MHz	
	Peripheral clock 1 (P1ϕ)		50.00	66.67	MHz	
	Peripheral clock 0 (P0ϕ)		25.00	33.33	MHz	

47.4.1 Clock Timing

Table 47.5 Clock Timing

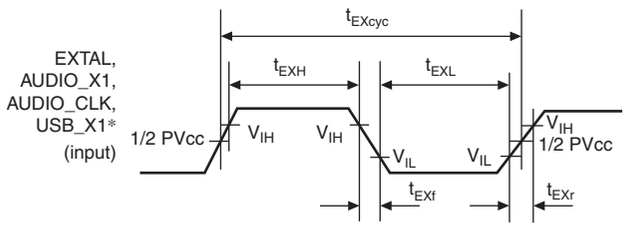
Item	Symbol	Min.	Max.	Unit	Figure
EXTAL clock input frequency (when the clock is supplied to USB 2.0 host/function module)	f_{EX}	12MHz ± 100ppm			Figure 47.1
EXTAL clock input frequency (when the clock isn't supplied to USB 2.0 host/function module)		10.00	13.33	MHz	
EXTAL clock input cycle time (when the clock isn't supplied to USB 2.0 host/function module)	t_{EXcyc}	75.00	100.00	ns	
AUDIO_X1 clock input frequency (crystal resonator connected)	f_{EX}	10.00	50.00	MHz	
AUDIO_X1 clock input cycle time (crystal resonator connected)	t_{EXcyc}	20.00	100.00	ns	
AUDIO_X1, AUDIO_CLK clock input frequency (external clock input)	f_{EX}	1.00	50.00	MHz	
AUDIO_X1, AUDIO_CLK clock input cycle time (external clock input)	t_{EXcyc}	20.00	1000.00	ns	
USB_X1 clock input frequency (when the 48-MHz clock is supplied to USB 2.0 host/function module and high-speed transfer function is used)	f_{EX}	48 MHz ±100 ppm			
USB_X1 clock input frequency (when the 48-MHz clock is supplied to USB 2.0 host/function module, high-speed transfer function is not used, and host controller function is used)		48 MHz ±500 ppm			
USB_X1 clock input frequency (when the 48-MHz clock is supplied to USB 2.0 host/function module, high-speed transfer function is not used, and host controller function is not used)		48 MHz ±2500 ppm			
EXTAL, AUDIO_X1, AUDIO_CLK, USB_X1 clock input low pulse width	t_{EXL}	0.4	0.6	t_{EXcyc}	
EXTAL, AUDIO_X1, AUDIO_CLK, USB_X1 clock input high pulse width	t_{EXH}	0.4	0.6	t_{EXcyc}	

chanhvinh.com

Table 47.5 Clock Timing

Item	Symbol	Min.	Max.	Unit	Figure
EXTAL, AUDIO_X1, AUDIO_CLK, USB_ X1 clock input rise time	t_{EXr}	—	4	ns	Figure 47.1
EXTAL, AUDIO_X1, AUDIO_CLK, USB_ X1 clock input fall time	t_{EXf}	—	4	ns	
CKIO clock output frequency	f_{OP}	50.00	66.67	MHz	
CKIO clock output cycle time	t_{cyc}	15.00	20.00	ns	Figure 47.2 (1) and Figure 47.2 (2)
CKIO clock output low pulse width 1	t_{CKOL1}	$t_{cyc}/2 - t_{CKOr1}$	—	ns	Figure 47.2 (1)
CKIO clock output high pulse width 1	t_{CKOH1}	$t_{cyc}/2 - t_{CKOr1}$	—	ns	
CKIO clock output rise time 1	t_{CKOr1}	—	3	ns	
CKIO clock output fall time 1	t_{CKOf1}	—	3	ns	
CKIO clock output low pulse width 2	t_{CKOL2}	$t_{cyc}/2 - t_{CKOr2}$	—	ns	Figure 47.2 (2)
CKIO clock output high pulse width 2	t_{CKOH2}	$t_{cyc}/2 - t_{CKOr2}$	—	ns	
CKIO clock output rise time 2	t_{CKOr2}	—	2	ns	
CKIO clock output fall time 2	t_{CKOf2}	—	2	ns	
On-chip PLL circuit oscillation settling time	t_{POSC}	1	—	ms	Figure 47.3 and Figure 47.5 (1)
On-chip oscillation circuit oscillation settling time (RTC_X1)	t_{ROSC}	—	3*	s	Figure 47.6
On-chip oscillation circuit oscillation settling time (other than above)		—	4*	ms	Figure 47.3, Figure 47.5 (1), and Figure 47.6
Mode hold time	t_{MDH}	200	—	ns	Figure 47.3 and Figure 47.5 (1)
SSCG stabilizing time	t_{SSCG}	1	—	us	Figure 47.4

Note: * Settings for values smaller than the above specifications may be possible, as long as the values are confirmed through evaluation by the manufacturer of the oscillator.



Note: * When the clock is input on the EXTAL, AUDIO_X1, AUDIO_CLK, or USB_X1

Figure 47.1 EXTAL, AUDIO_X1, AUDIO_CLK, and USB_X1 Clock Input Timing

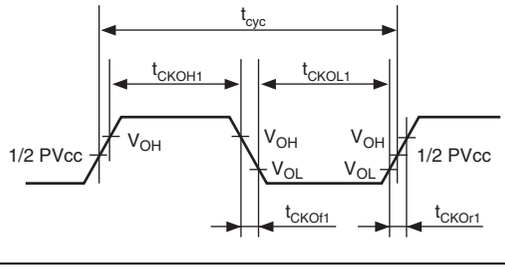


Figure 47.2 (1) CKIO Clock Output Timing 1

chanhvinh.com

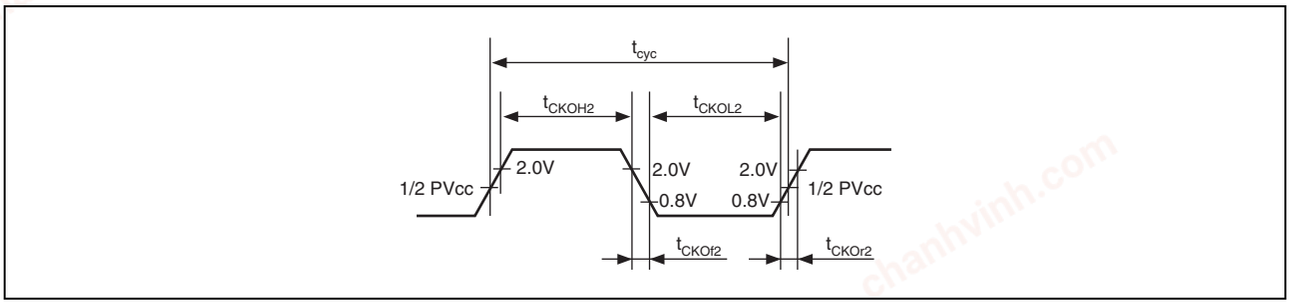


Figure 47.2 (2) CKIO Clock Output Timing 2

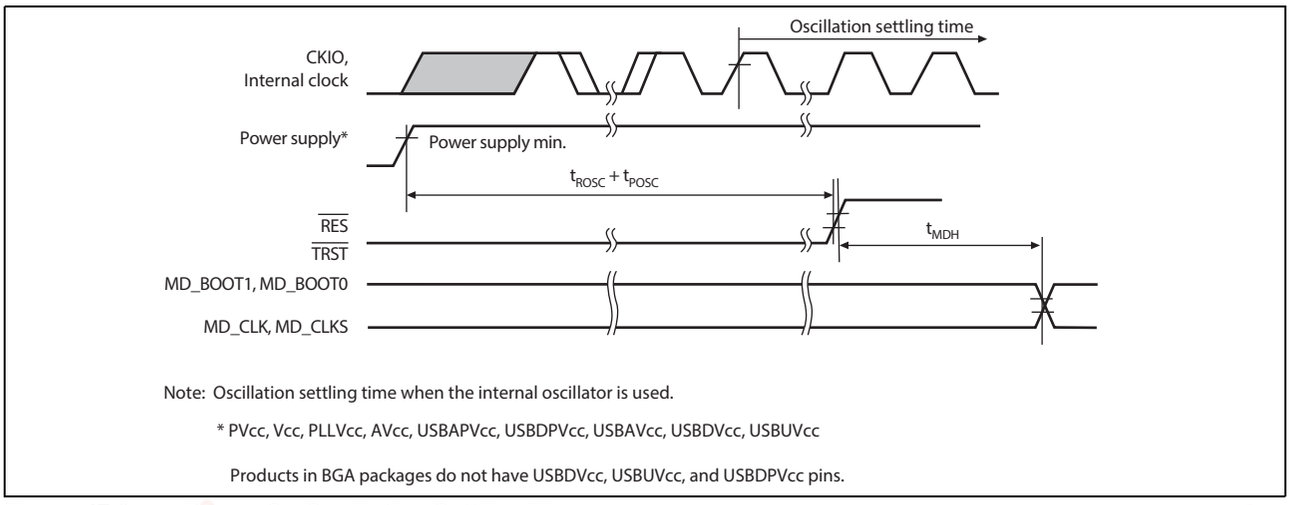


Figure 47.3 Power-On Oscillation Settling Time

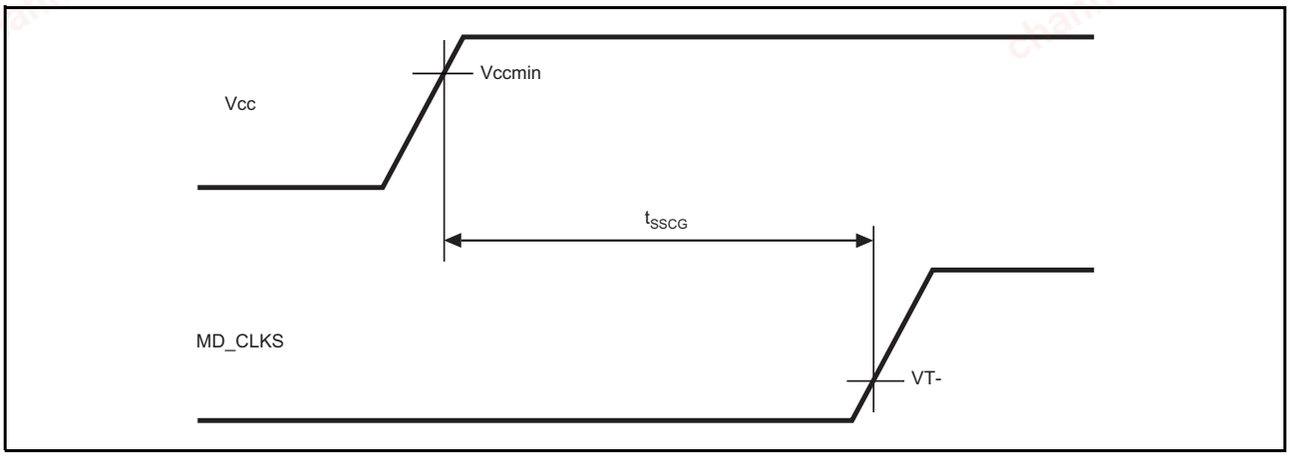


Figure 47.4 SSCG Stabilizing Time

chanhvinh.com

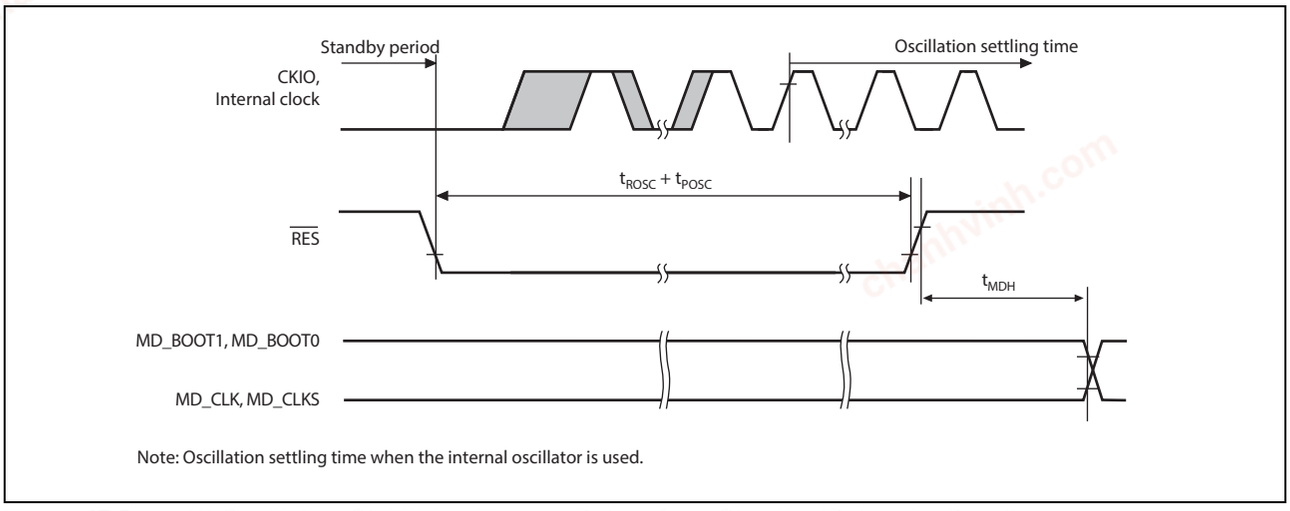


Figure 47.5 (1) Oscillation Stabilizing Time on Return from Standby (Return by Reset)

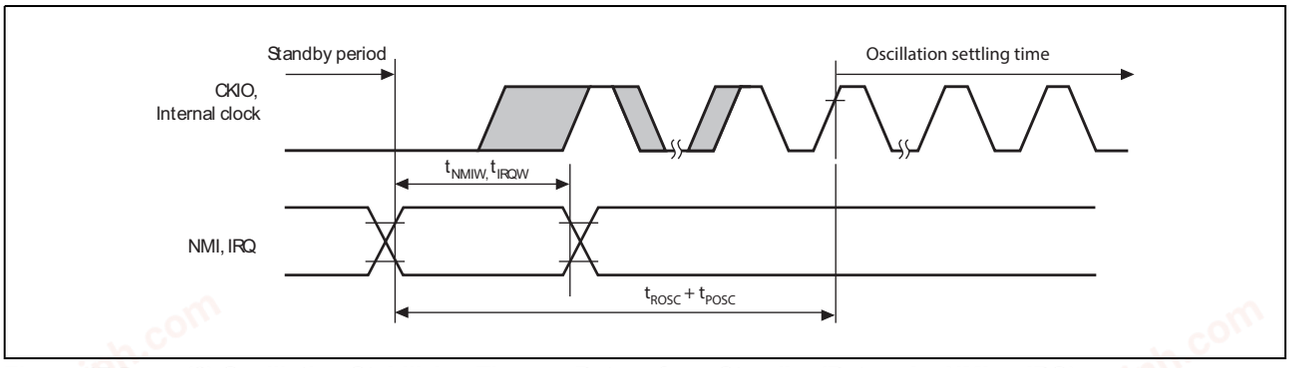


Figure 47.5 (2) Oscillation Stabilizing Time on Return from Standby (Return by NMI or IRQ)

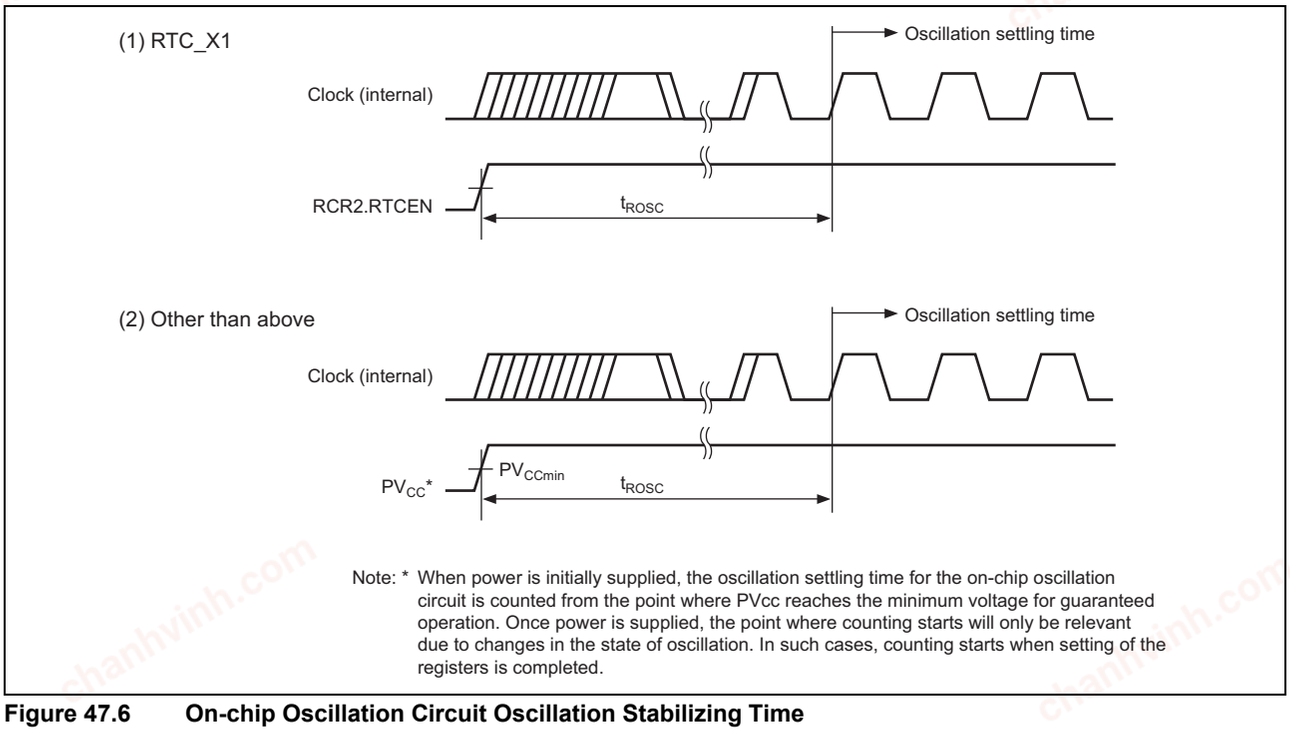


Figure 47.6 On-chip Oscillation Circuit Oscillation Stabilizing Time

chanhvinh.com

47.4.2 Control Signal Timing

Table 47.6 Control Signal Timing

Item		Symbol	Min.	Max.	Unit	Figure
$\overline{\text{RES}}$ pulse width	Exit from standby mode	t_{RESW}	t_{ROSC} + t_{POSC}		ms	Figure 47.7 (1) and Figure 47.5 (1)
	Other than above		20	—	t_{cyc}	
$\overline{\text{TRST}}$ pulse width		t_{TRSW}	20	—	t_{cyc}	
NMI pulse width		t_{NMIW}	20	—	t_{cyc}	Figure 47.7 (2) and Figure 47.5 (2)
IRQ pulse width		t_{IRQW}	20	—	t_{cyc}	
TINT pulse width		t_{TINTW}	20	—	t_{cyc}	
$\overline{\text{RES}}$ input rise time*1		t_{RSr}	—	500	µs	Figure 47.7 (3)
$\overline{\text{RES}}$ negating hold time*2		t_{RSNH}	0	—	ns	Figure 47.7 (4)

Note 1. Make sure that this specification is satisfied when the same signal is controlling the $\overline{\text{TRST}}$ and $\overline{\text{RES}}$ pins.
Note 2. Make sure that this specification is satisfied when different signals are controlling the $\overline{\text{TRST}}$ and $\overline{\text{RES}}$ pins.

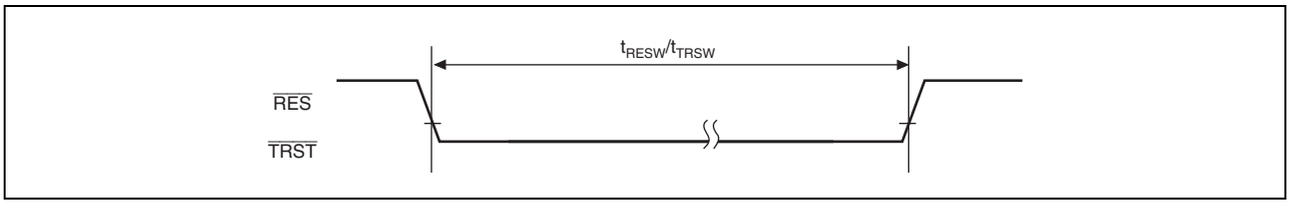


Figure 47.7 (1) Reset Input Timing 1

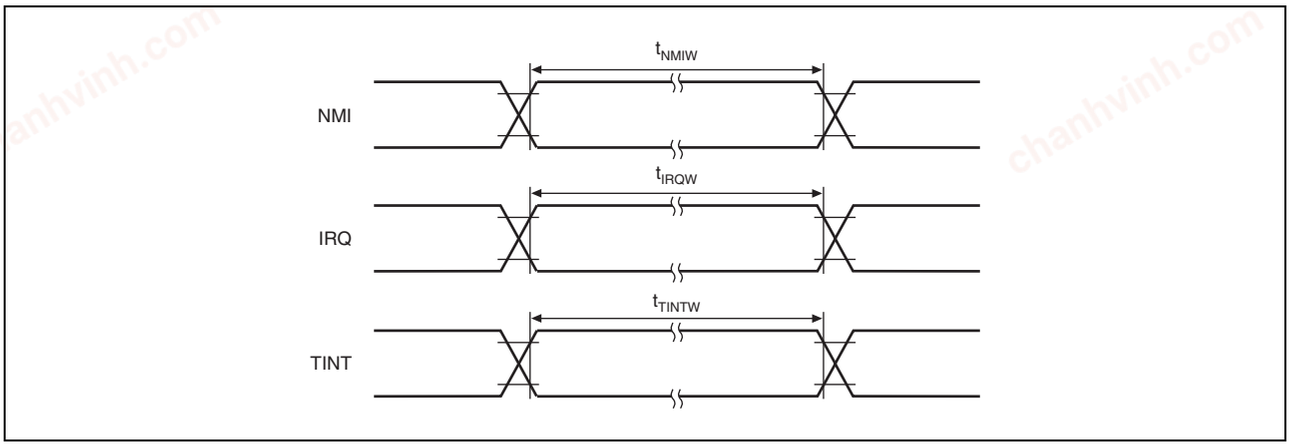


Figure 47.7 (2) Interrupt Signal Input Timing

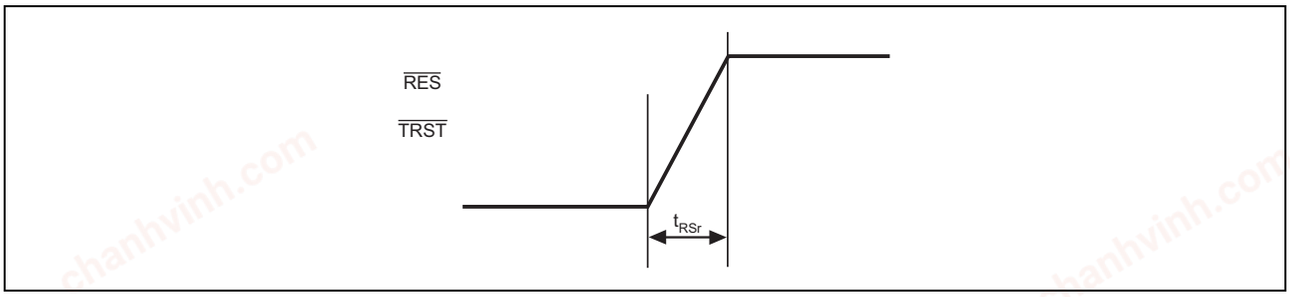


Figure 47.7 (3) Reset Input Timing 2

chanhvinh.com

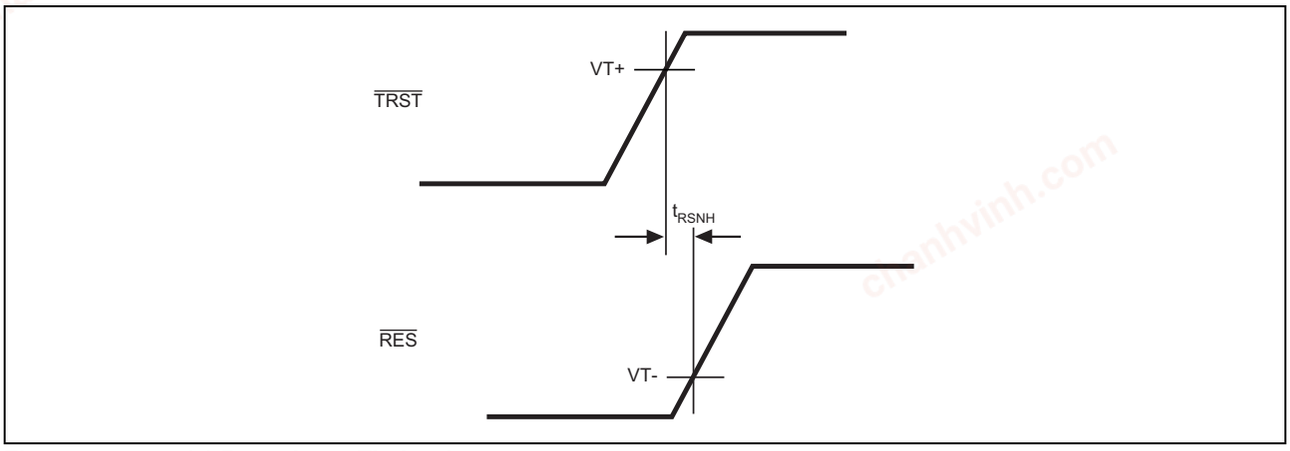


Figure 47.7 (4) Reset Input Timing 3

chanhvinh.com

47.4.3 Bus Timing

Table 47.7 Bus Timing

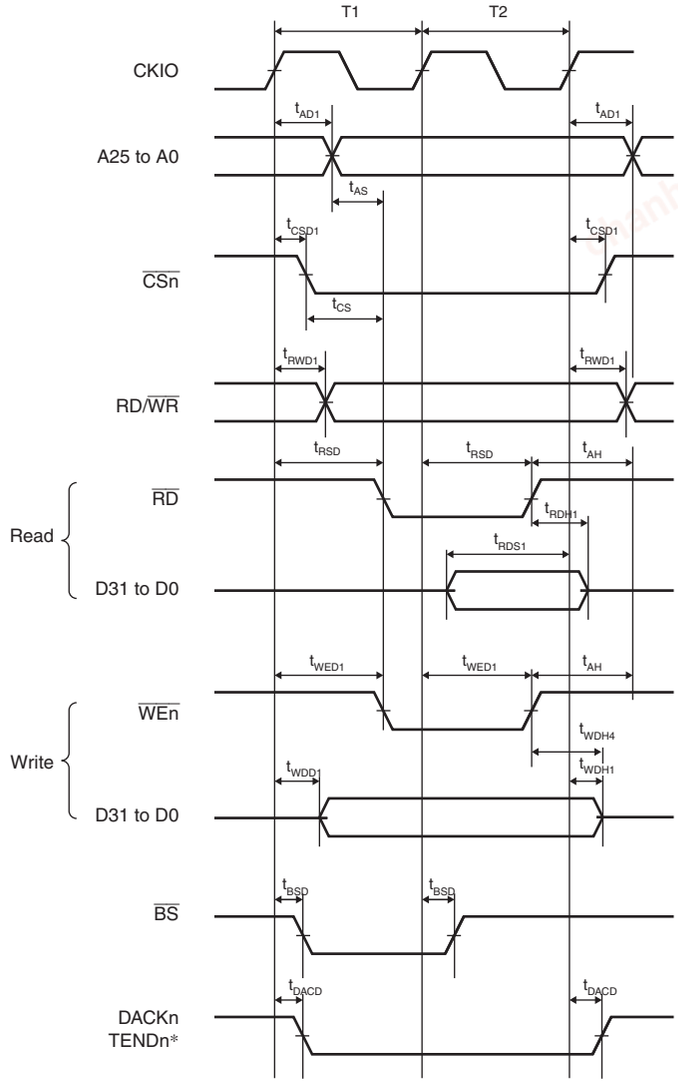
Item	Symbol	CKIO = 66.67 MHz*1 Min.	Max.	Unit	Figure
Address delay time 1	t_{AD1}	0/2*3	12	ns	Figure 47.8 to Figure 47.32
Address delay time 2	t_{AD2}	1/2t_{cyc}	1/2t_{cyc} + 12	ns	Figure 47.15
Address setup time	t_{AS}	0	—	ns	Figure 47.8 to Figure 47.11, Figure 47.15
Chip enable setup time	t_{CS}	0	—	ns	Figure 47.8 to Figure 47.11, Figure 47.15
Address hold time	t_{AH}	0	—	ns	Figure 47.8 to Figure 47.11
$\overline{BS}$ delay time	t_{BSD}	—	12	ns	Figure 47.8 to Figure 47.29
$\overline{CS}$ delay time 1	t_{CSD1}	0/2*3	12	ns	Figure 47.8 to Figure 47.32
Read write delay time 1	t_{RWD1}	0/2*3	12	ns	Figure 47.8 to Figure 47.32
Read strobe delay time	t_{RSD}	1/2t_{cyc}	1/2t_{cyc} + 12	ns	Figure 47.8 to Figure 47.15
Read data setup time 1	t_{RDS1}	1/2t_{cyc}+ 5	—	ns	Figure 47.8 to Figure 47.14
Read data setup time 2	t_{RDS2}	7	—	ns	Figure 47.16 to Figure 47.19, Figure 47.24 to Figure 47.26
Read data setup time 3	t_{RDS3}	1/2t_{cyc} + 5	—	ns	Figure 47.15
Read data hold time 1	t_{RDH1}	0	—	ns	Figure 47.8 to Figure 47.14
Read data hold time 2	t_{RDH2}	2	—	ns	Figure 47.16 to Figure 47.19, Figure 47.24 to Figure 47.26
Read data hold time 3	t_{RDH3}	0	—	ns	Figure 47.15
Write enable delay time 1	t_{WED1}	1/2t_{cyc}	1/2t_{cyc} + 12	ns	Figure 47.8 to Figure 47.13
Write enable delay time 2	t_{WED2}	—	12	ns	Figure 47.14
Write data delay time 1	t_{WDD1}	—	12	ns	Figure 47.8 to Figure 47.14
Write data delay time 2	t_{WDD2}	—	12	ns	Figure 47.20 to Figure 47.23, Figure 47.27 to Figure 47.29
Write data hold time 1	t_{WDH1}	1	—	ns	Figure 47.8 to Figure 47.14
Write data hold time 2	t_{WDH2}	2	—	ns	Figure 47.20 to Figure 47.23, Figure 47.27 to Figure 47.29
Write data hold time 4	t_{WDH4}	0	—	ns	Figure 47.8 to Figure 47.12
$\overline{WAIT}$ setup time	t_{WTS}	1/2t_{cyc} + 4.5	—	ns	Figure 47.9 to Figure 47.15
$\overline{WAIT}$ hold time	t_{WTH}	1/2t_{cyc} + 3.5	—	ns	Figure 47.9 to Figure 47.15
$\overline{RAS}$ delay time 1	t_{RASD1}	2	12	ns	Figure 47.16 to Figure 47.32
$\overline{CAS}$ delay time 1	t_{CASD1}	2	12	ns	Figure 47.16 to Figure 47.32
DQM delay time 1	t_{DQMD1}	2	12	ns	Figure 47.16 to Figure 47.29
CKE delay time 1	t_{CKED1}	2	12	ns	Figure 47.31
$\overline{AH}$ delay time	t_{AHD}	1/2t_{cyc}	1/2t_{cyc} + 12	ns	Figure 47.12
Multiplexed address delay time	t_{MAD}	—	12	ns	Figure 47.12
Multiplexed address hold time	t_{MAH}	1	—	ns	Figure 47.12
Address setup time for $\overline{AH}$	t_{AVVH}	1/2t_{cyc} – 2	—	ns	Figure 47.12
DACK, TEND delay time	t_{DACD}	Refer to the direct memory access controller timing		ns	Figure 47.8 to Figure 47.29

Note 1. The maximum value (fmax) of CKIO (external bus clock) depends on the number of wait cycles and the system configuration of your board.

Note 2. 1/2 t_{cyc} indicated in minimum and maximum values for the item of delay, setup, and hold times represents a half cycle from the rising edge with a clock. That is, 1/2 t_{cyc} describes a reference of the falling edge with a clock.

Note 3. Values when SDRAM is used.

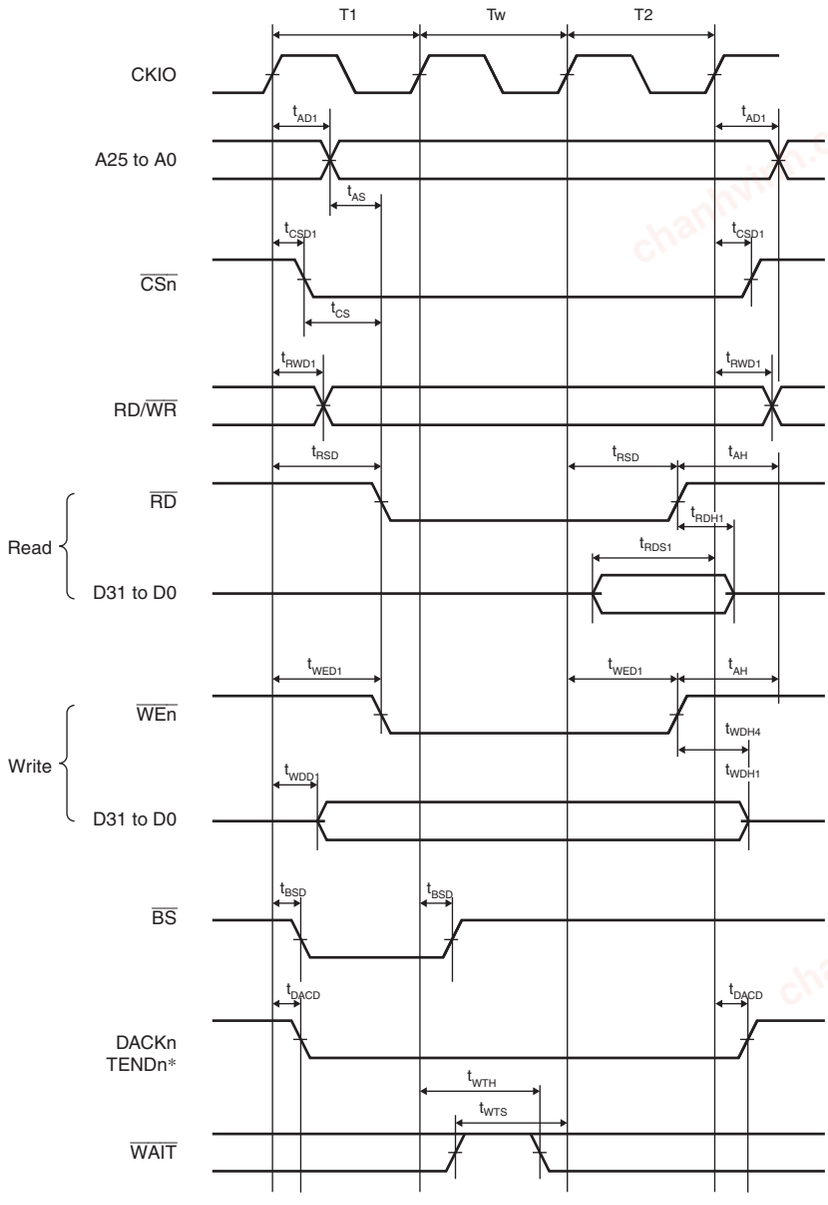
chanhvinh.com



Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 47.8 Basic Bus Timing for Normal Space (No Wait)

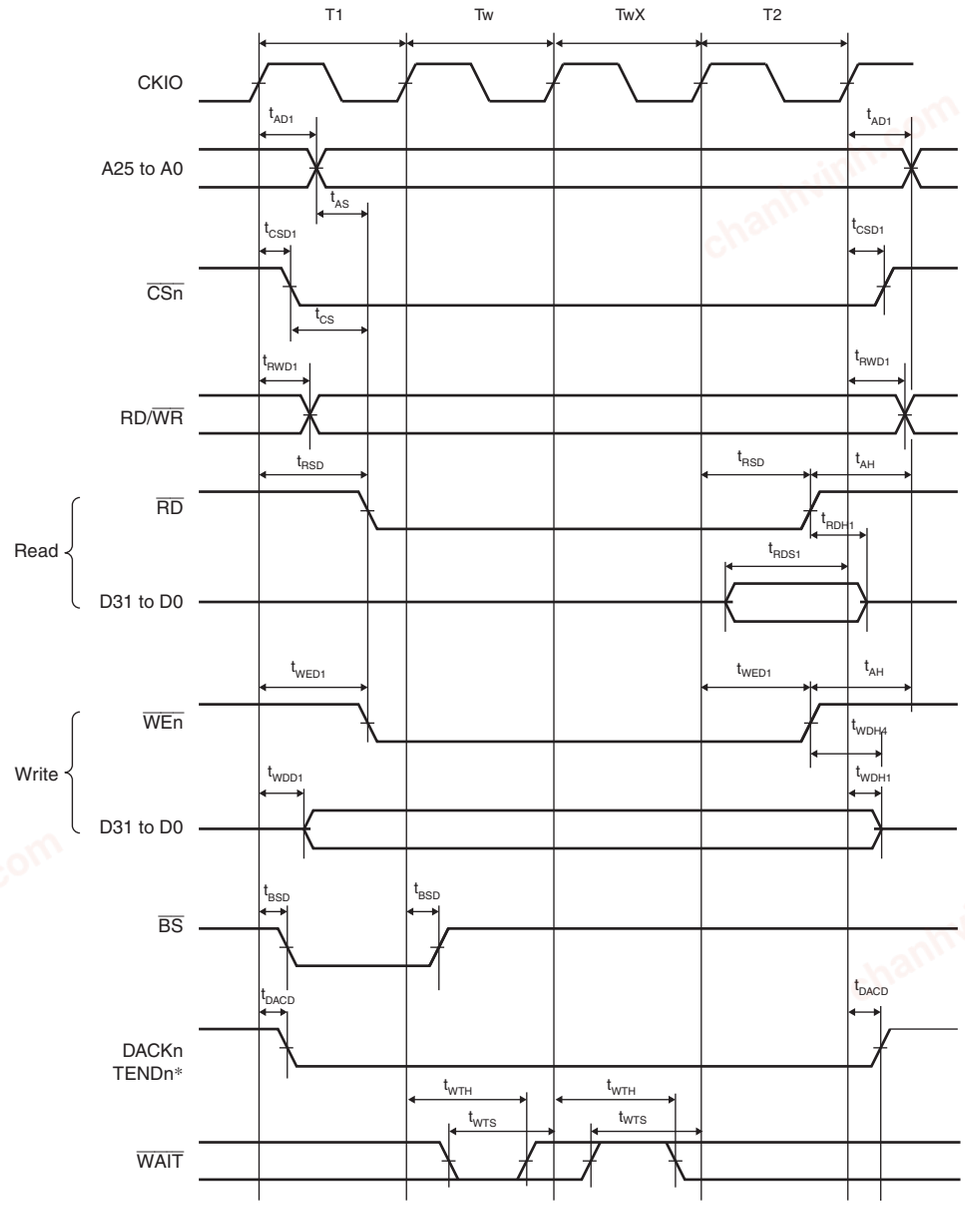
chanhvinh.com



Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 47.9 Basic Bus Timing for Normal Space (One Software Wait Cycle)

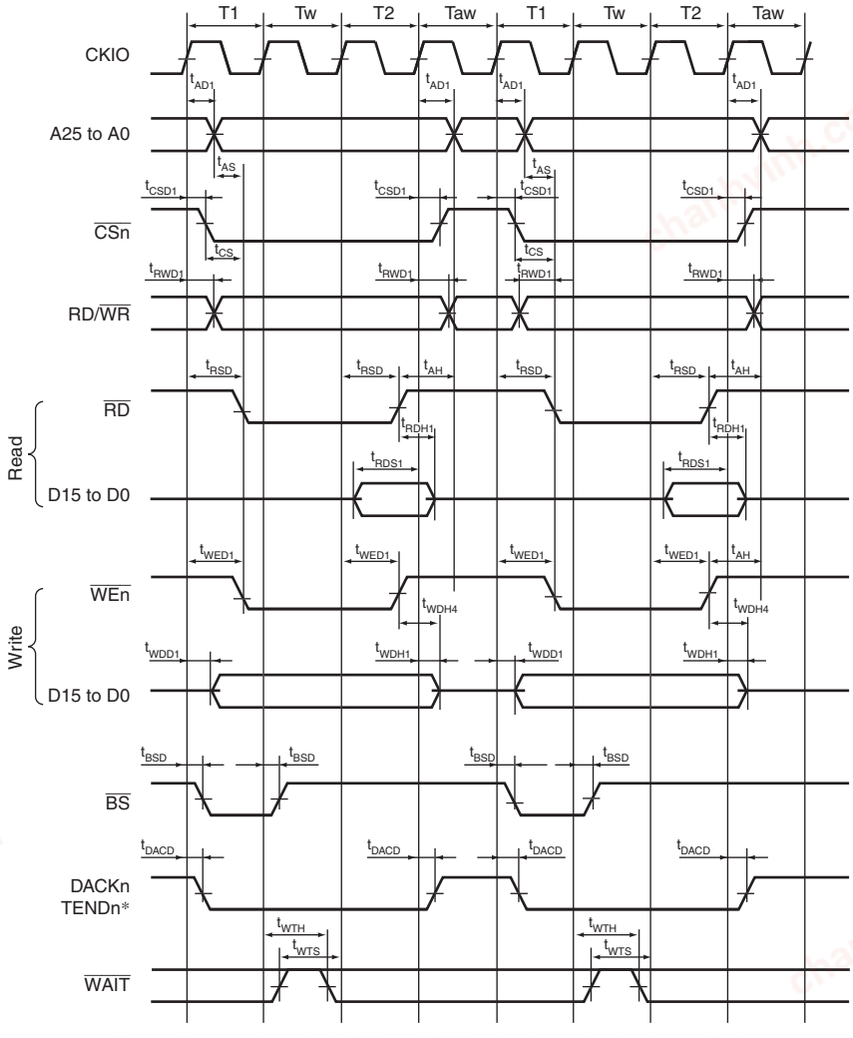
chanhvinh.com



Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 47.10 Basic Bus Timing for Normal Space (One Software Wait Cycle, One External Wait Cycle)

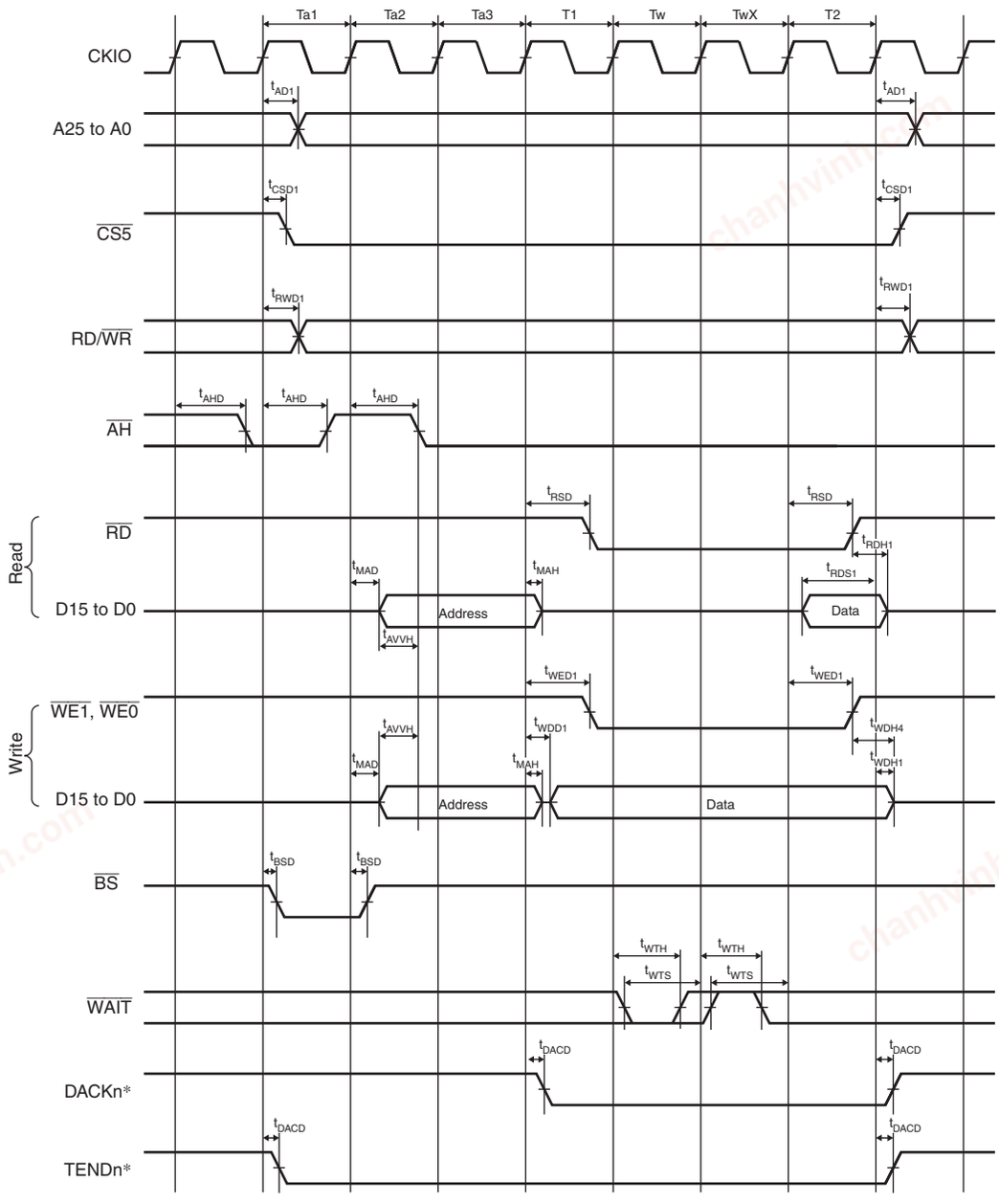
chanhvinh.com



Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 47.11 Basic Bus Timing for Normal Space (One Software Wait Cycle, External Wait Cycle Valid (WM Bit = 0), No Idle Cycle)

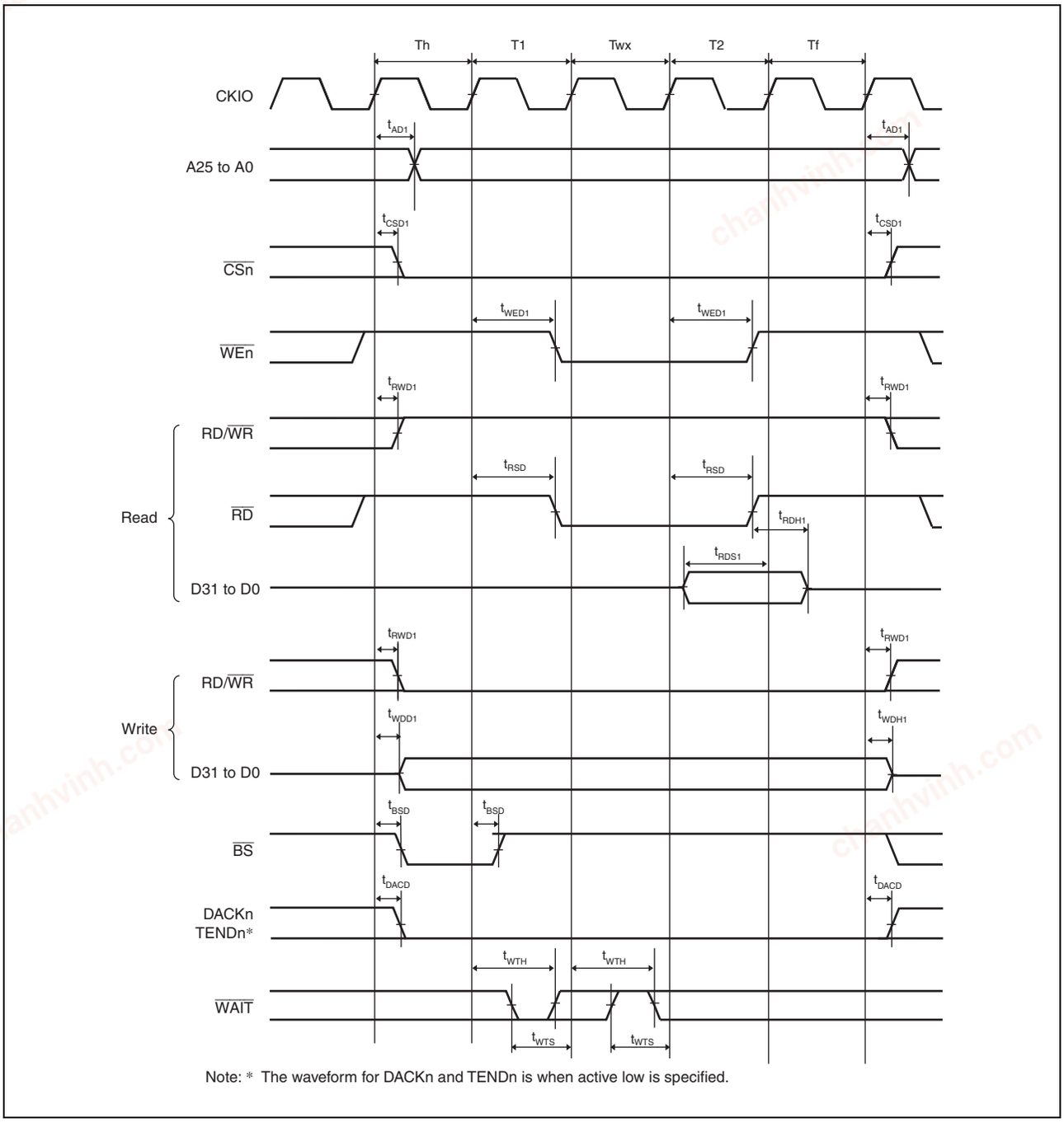
chanhvinh.com



Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 47.12 MPX-I/O Interface Bus Cycle (Three Address Cycles, One Software Wait Cycle, One External Wait Cycle)

chanhvinh.com



Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 47.13 Bus Cycle of SRAM with Byte Selection (SW = 1 Cycle, HW = 1 Cycle, One Asynchronous External Wait Cycle, BAS = 0 (Write Cycle $\overline{UB}$/$\overline{LB}$ Control))

chanhvinh.com

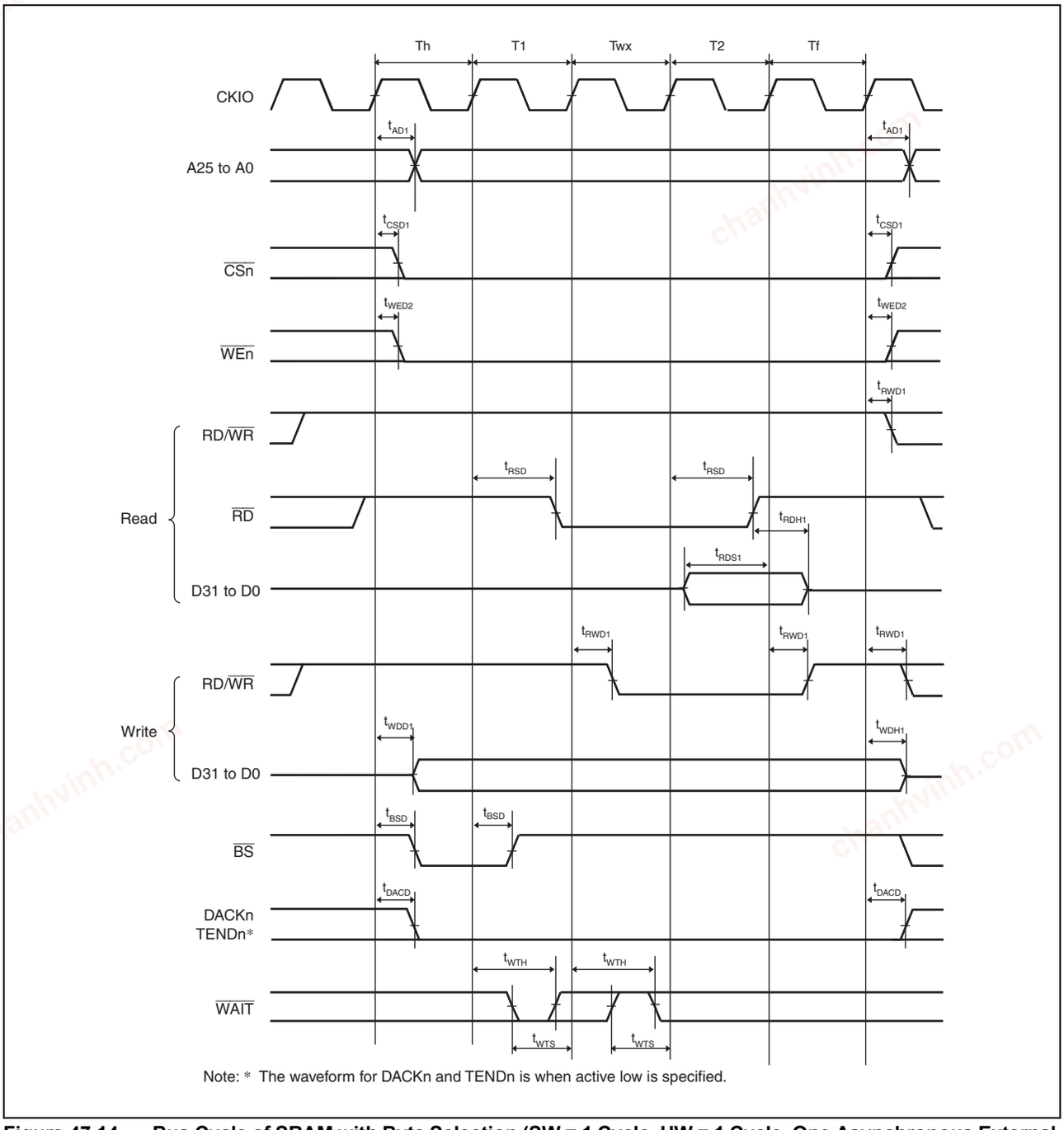
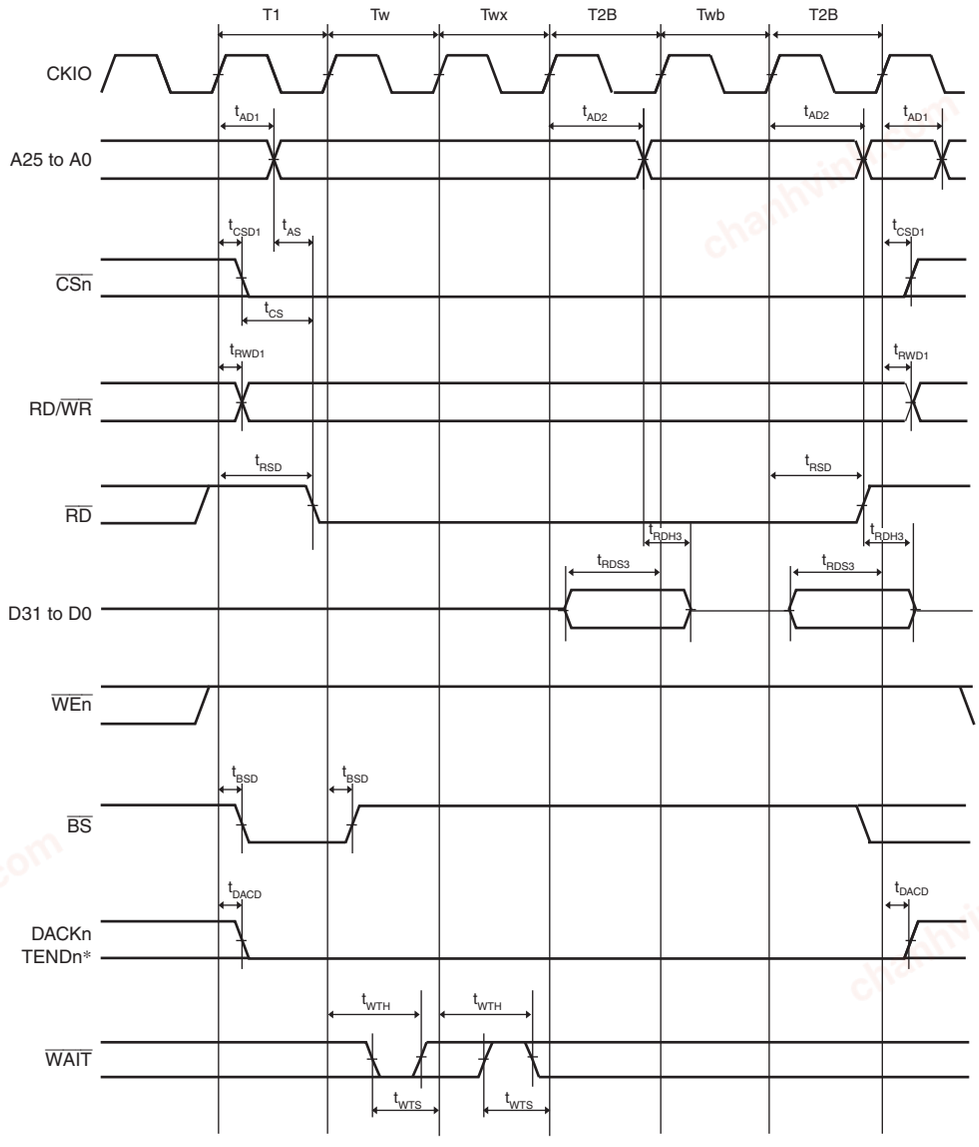


Figure 47.14 Bus Cycle of SRAM with Byte Selection (SW = 1 Cycle, HW = 1 Cycle, One Asynchronous External Wait Cycle, BAS = 1 (Write Cycle $\overline{WE}$ Control))

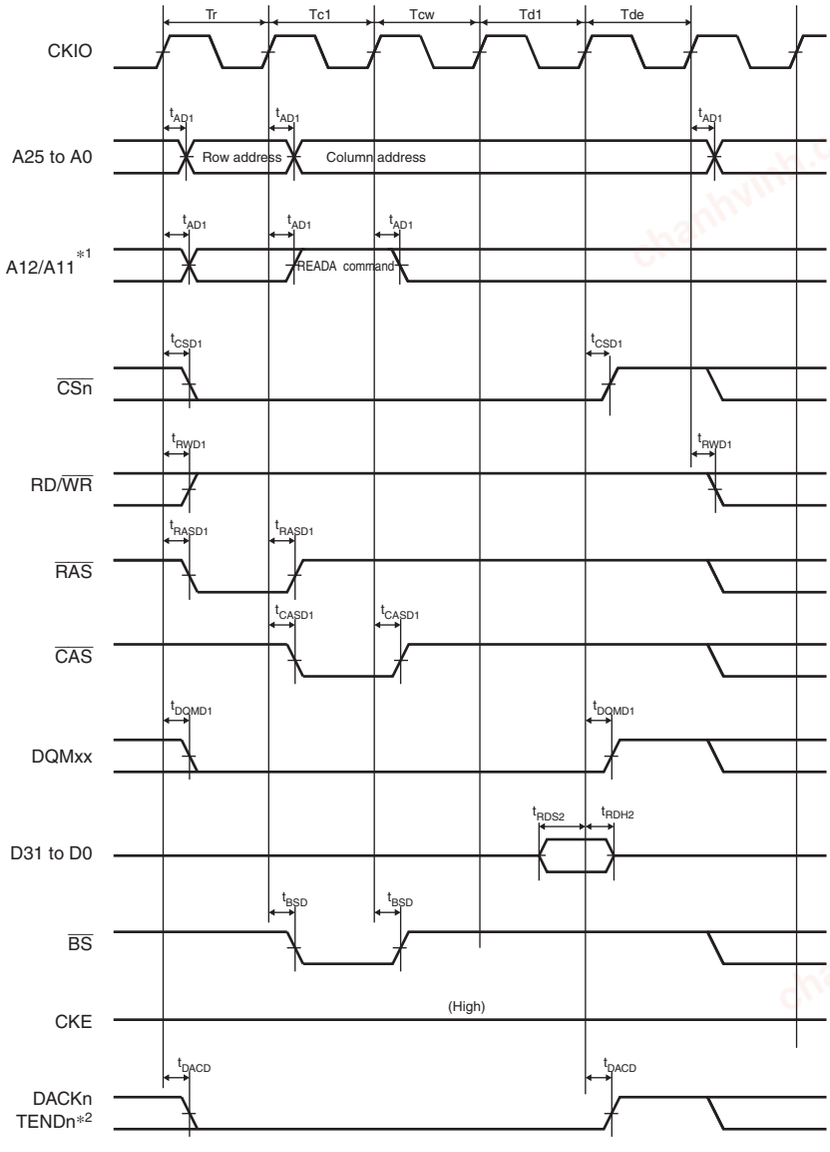
chanhvinh.com



Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 47.15 Burst ROM Read Cycle (One Software Wait Cycle, One Asynchronous External Burst Wait Cycle, Two Burst)

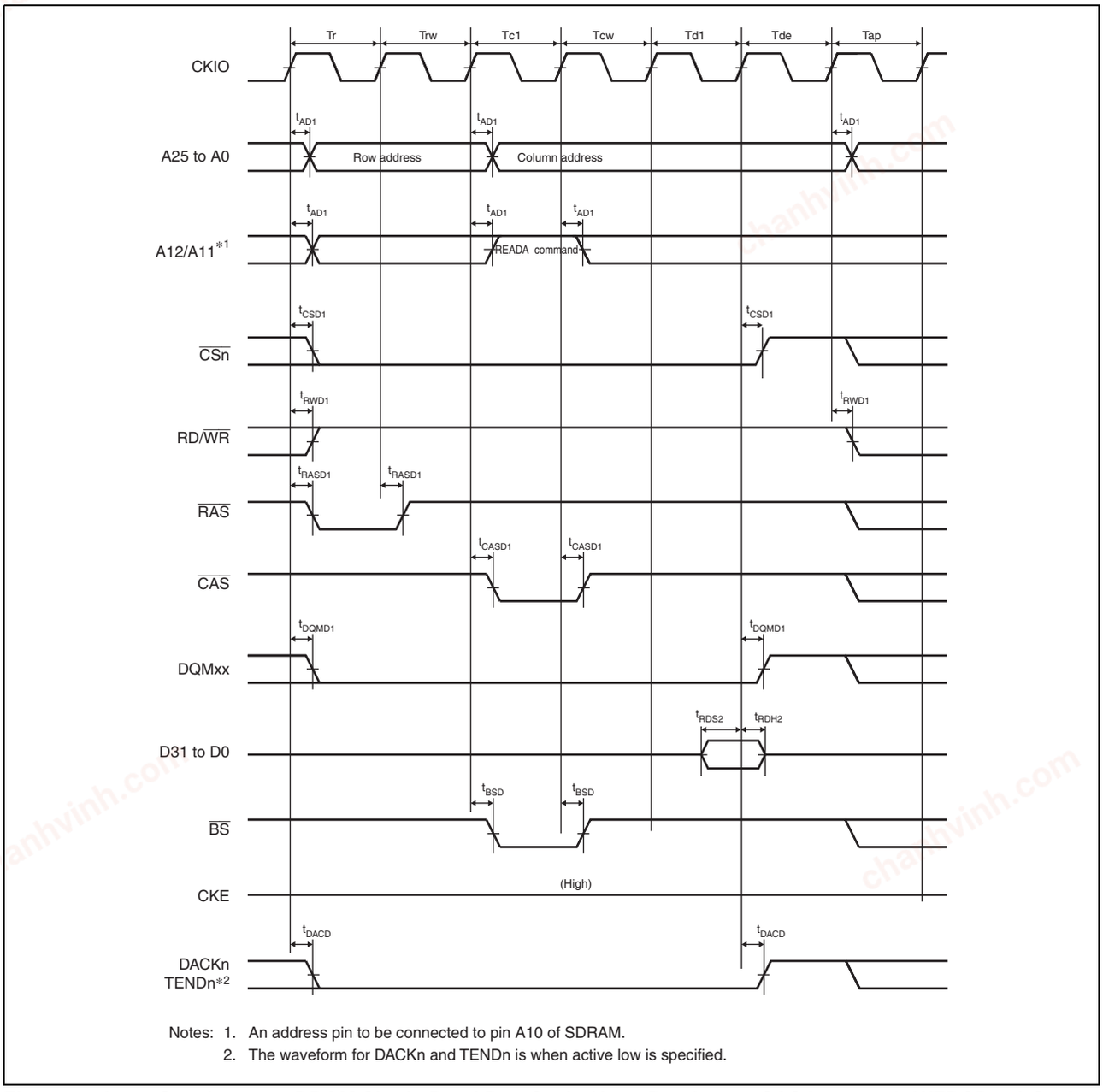
chanhvinh.com



Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 47.16 Synchronous DRAM Single Read Bus Cycle (Auto Precharge, CAS Latency 2, WTRCD = 0 Cycle, WTRP = 0 Cycle)

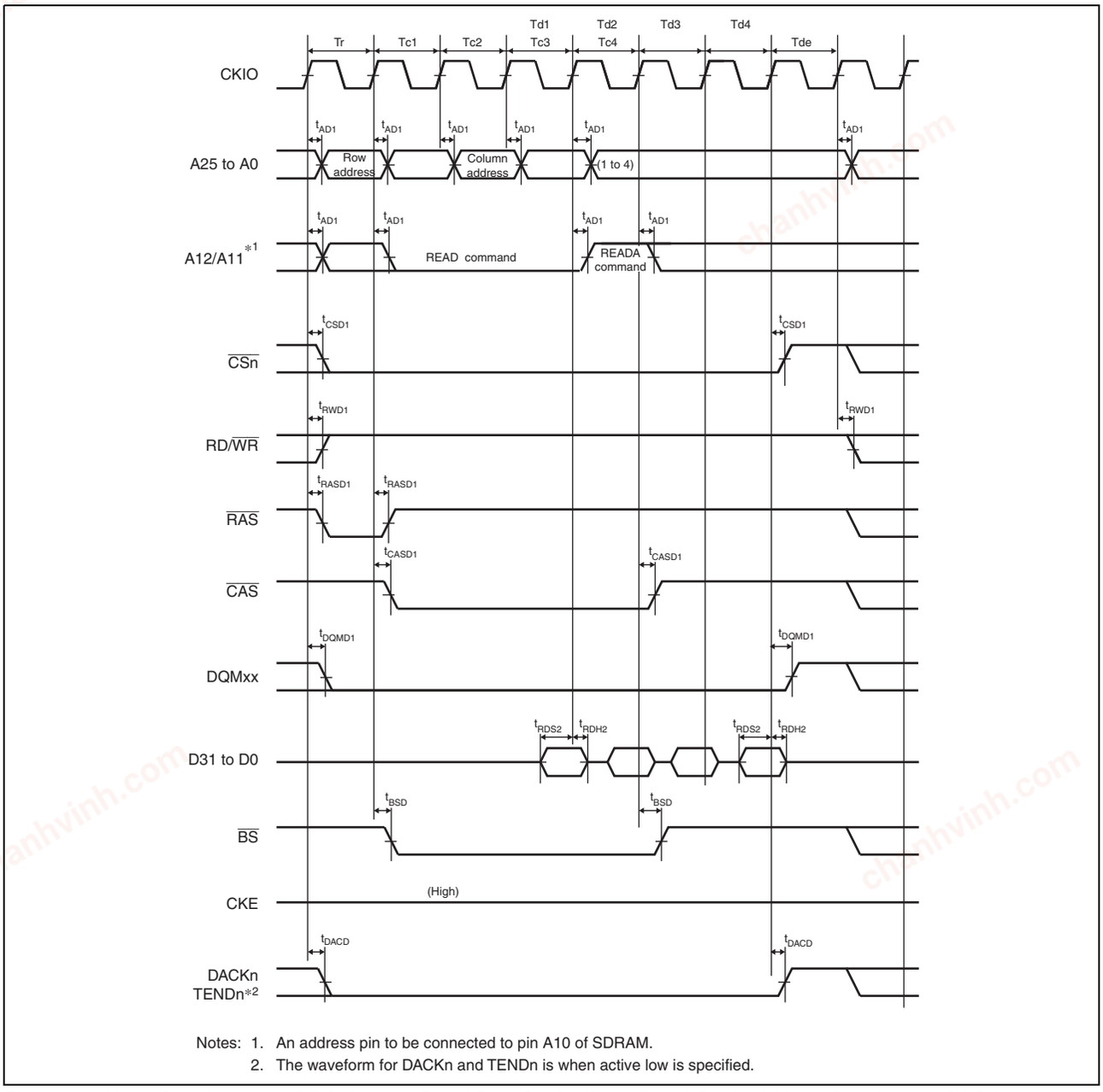
chanhvinh.com



Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 47.17 Synchronous DRAM Single Read Bus Cycle (Auto Precharge, CAS Latency 2, WTRCD = 1 Cycle, WTRP = 1 Cycle)

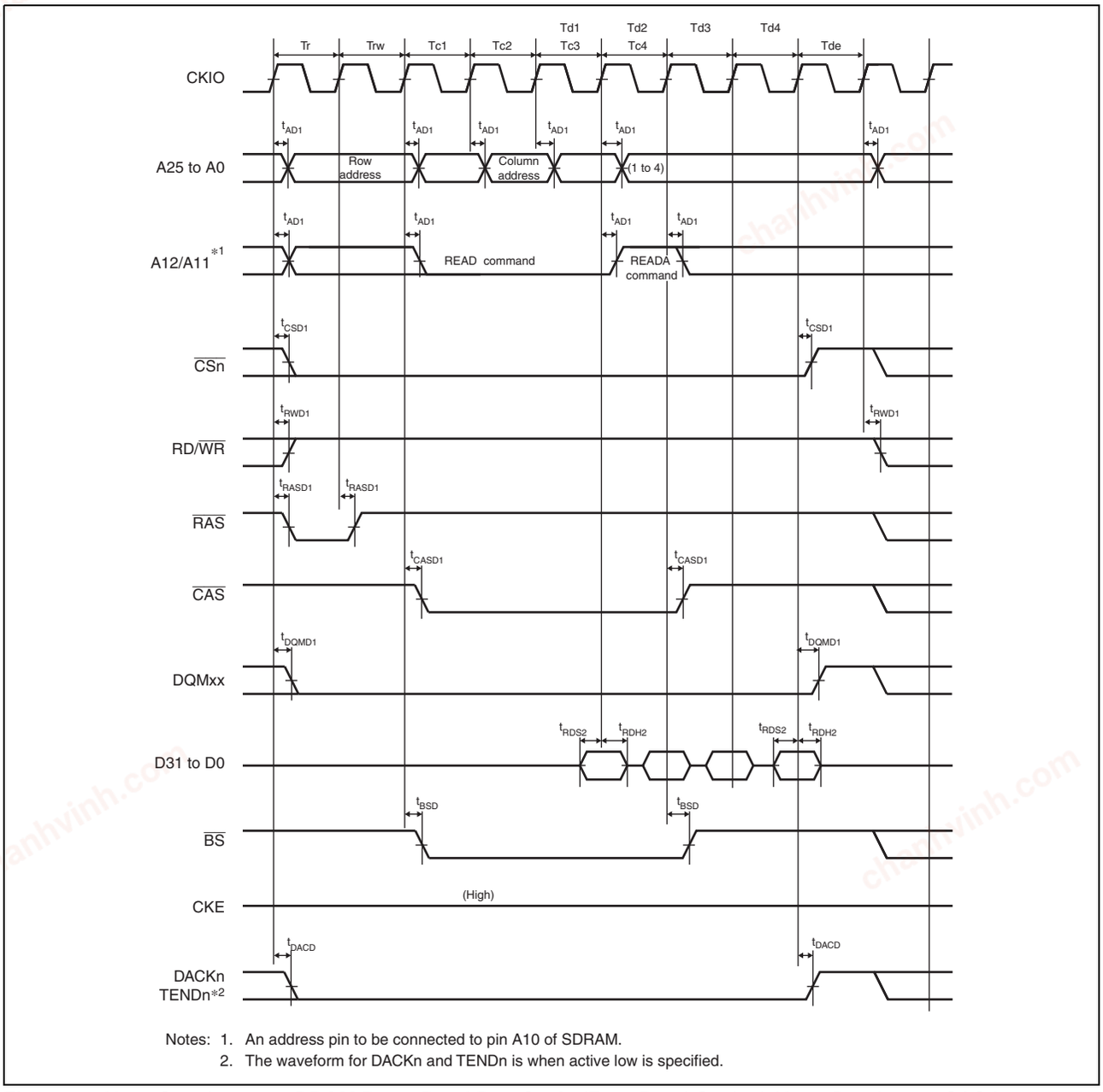
chanhvinh.com



Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 47.18 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles) (Auto Precharge, CAS Latency 2, WTRCD = 0 Cycle, WTRP = 1 Cycle)

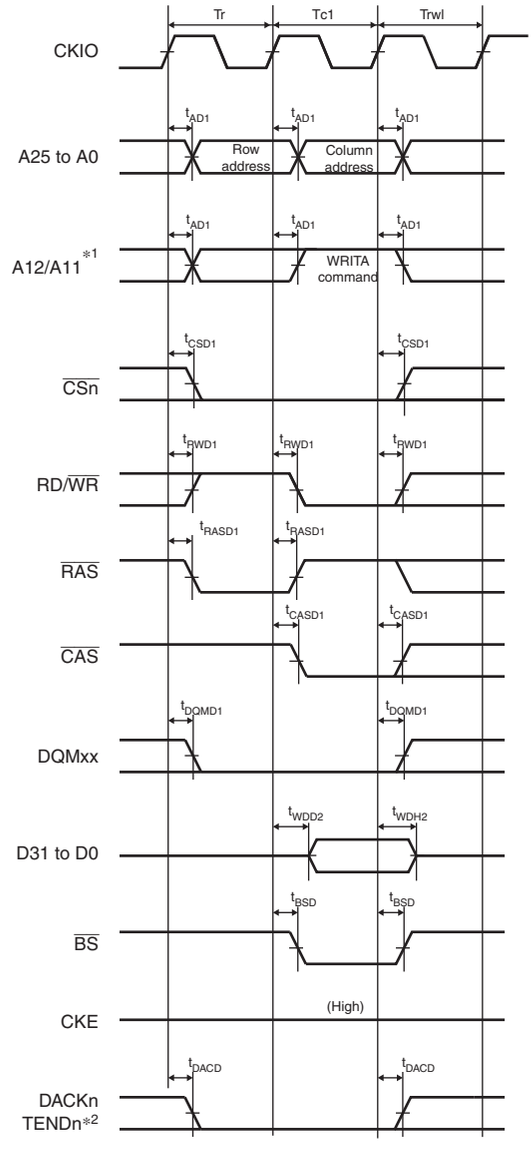
chanhvinh.com



Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 47.19 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles) (Auto Precharge, CAS Latency 2, WTRCD = 1 Cycle, WTRP = 0 Cycle)

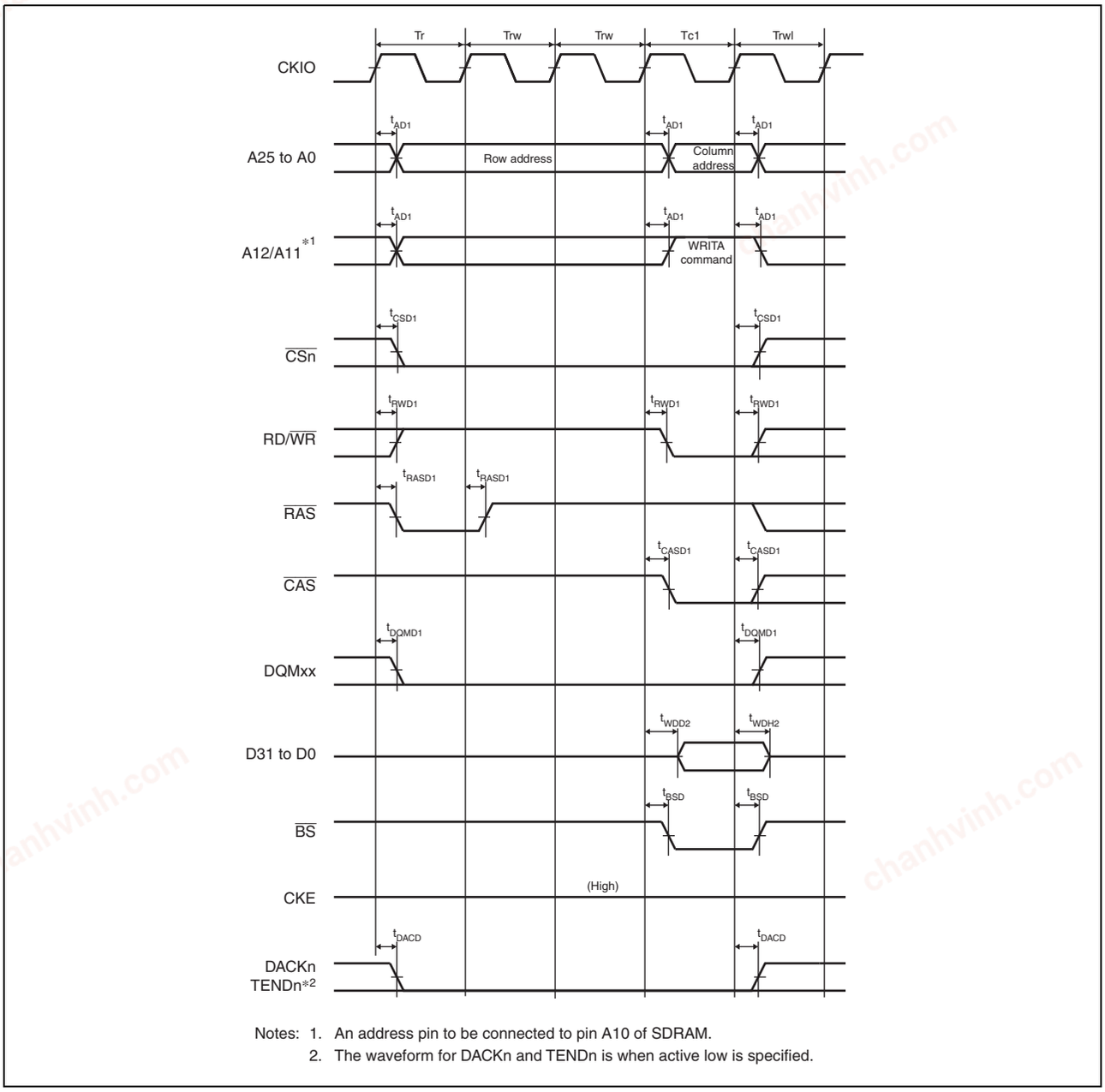
chanhvinh.com



Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 47.20 Synchronous DRAM Single Write Bus Cycle (Auto Precharge, TRWL = 1 Cycle)

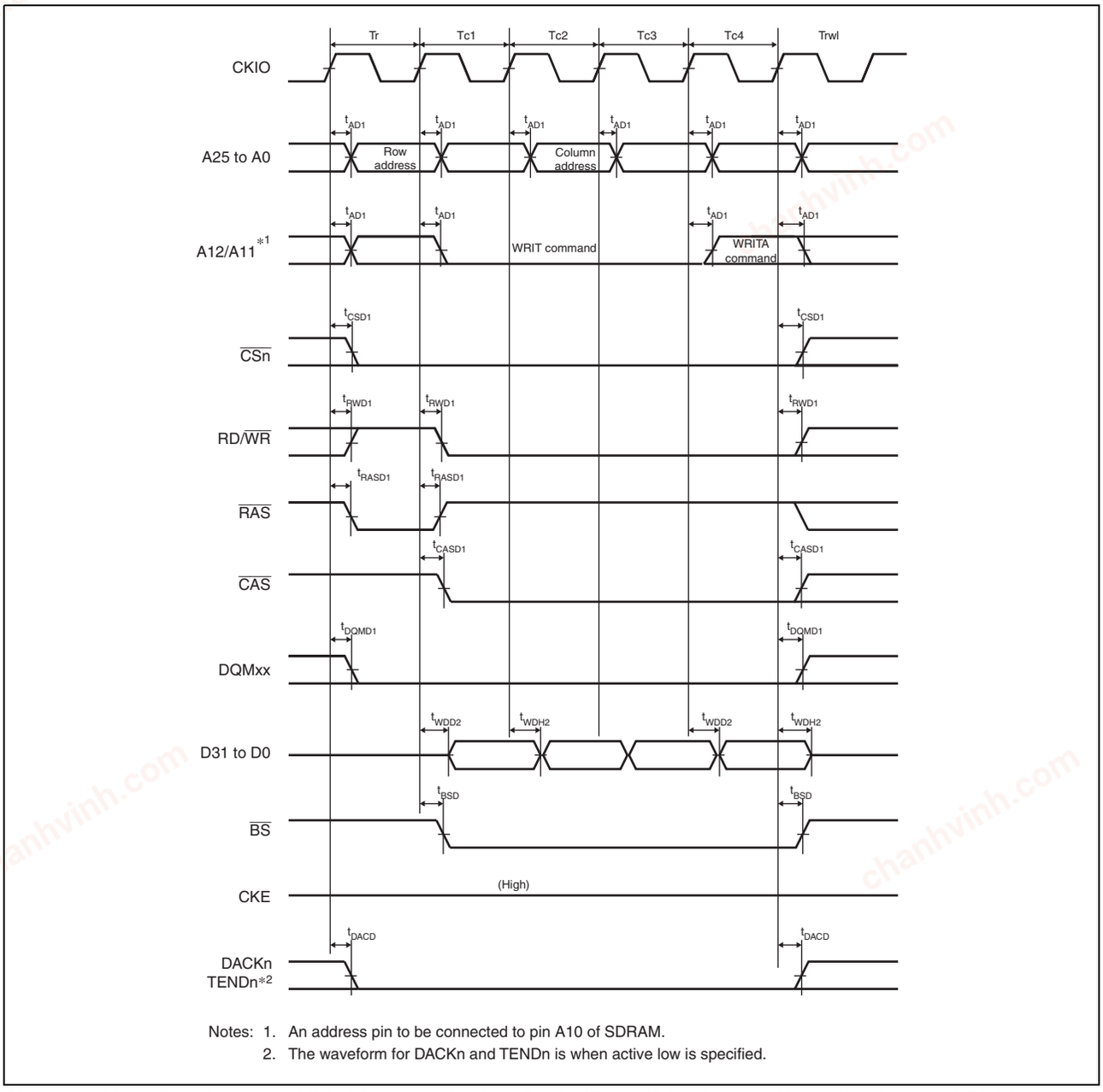
chanhvinh.com



Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 47.21 Synchronous DRAM Single Write Bus Cycle (Auto Precharge, WTRCD = 2 Cycles, TRWL = 1 Cycle)

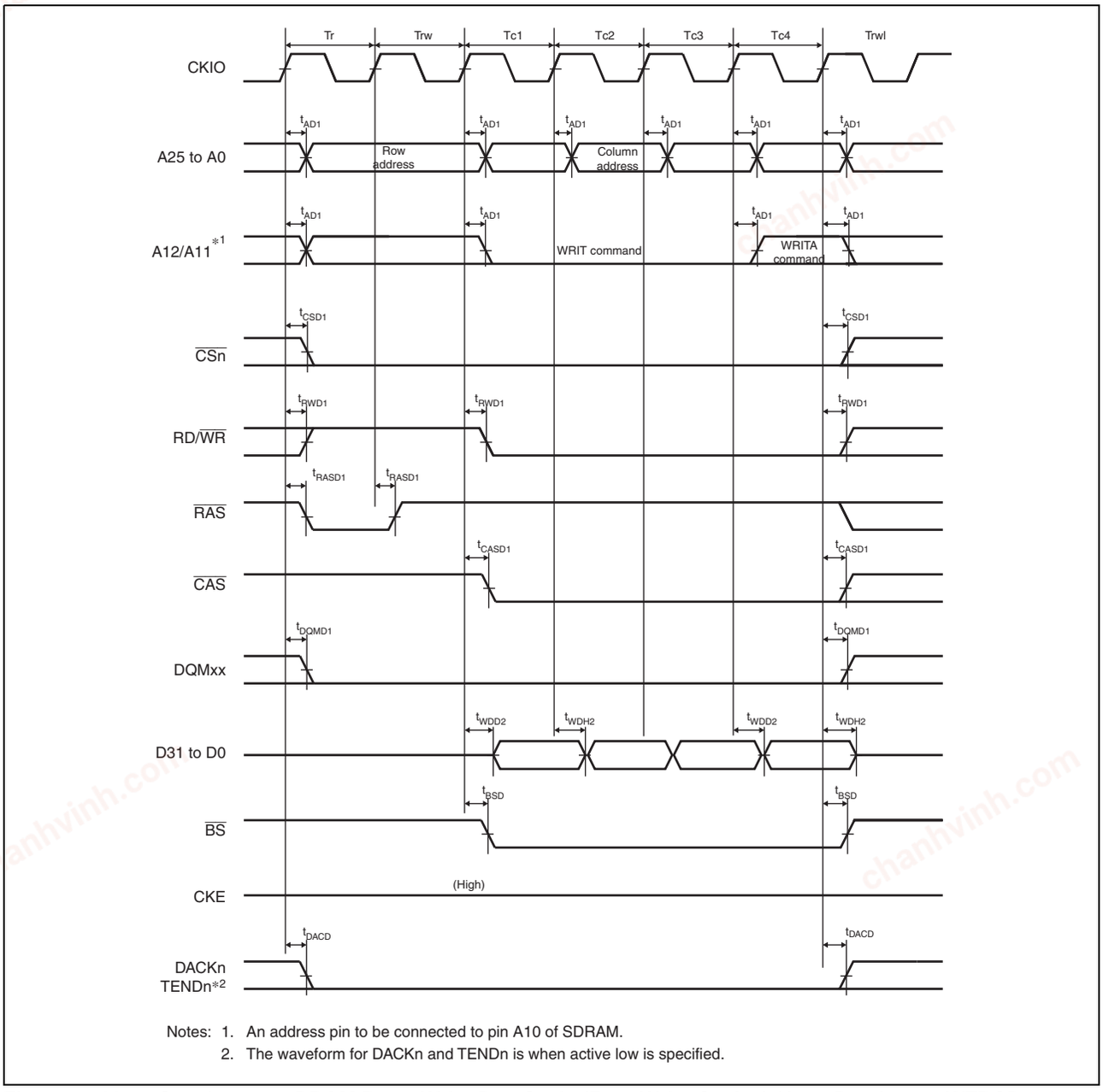
chanhvinh.com



Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 47.22 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Auto Precharge, WTRCD = 0 Cycle, TRWL = 1 Cycle)

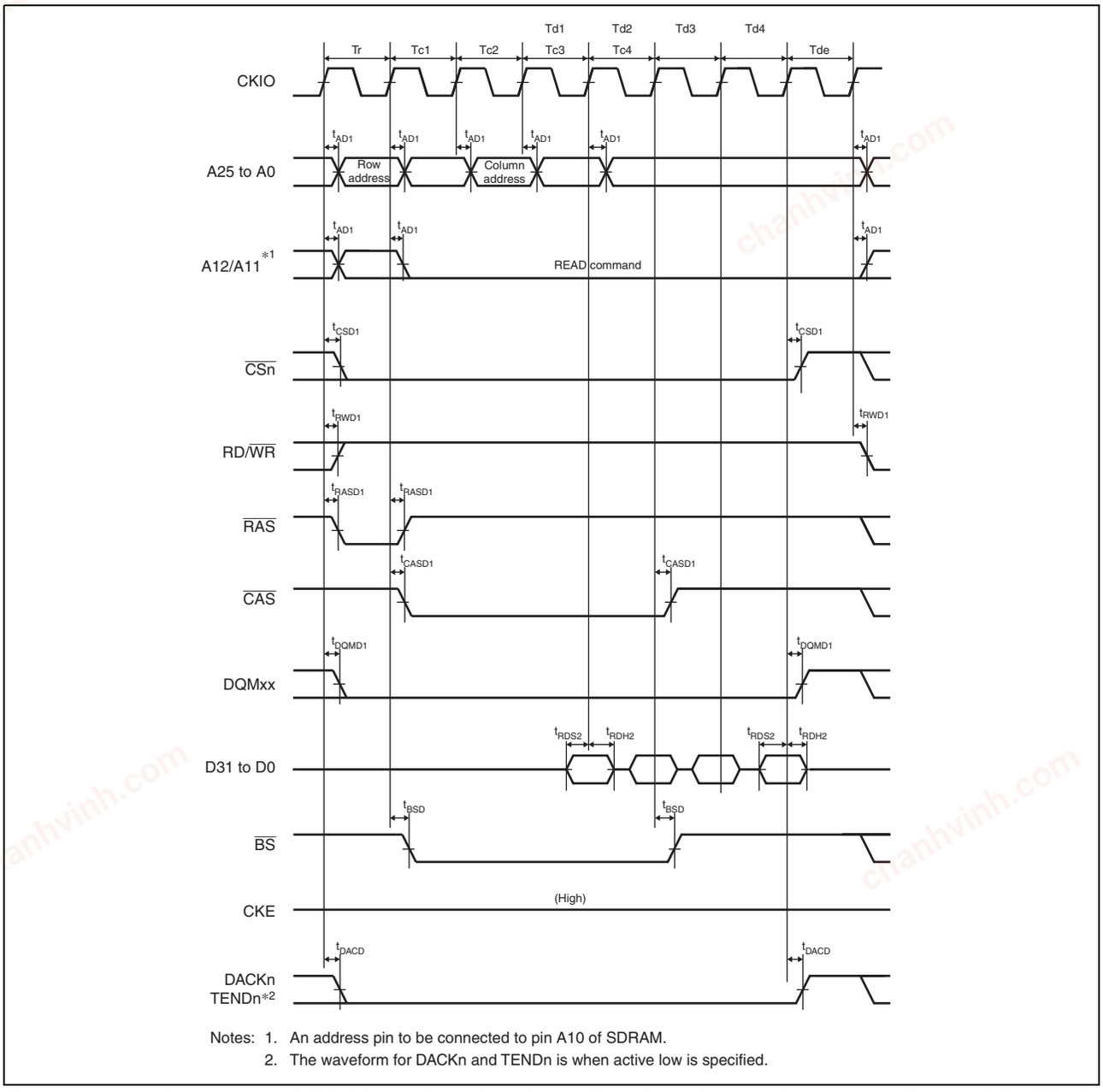
chanhvinh.com



Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 47.23 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Auto Precharge, WTRCD = 1 Cycle, TRWL = 1 Cycle)

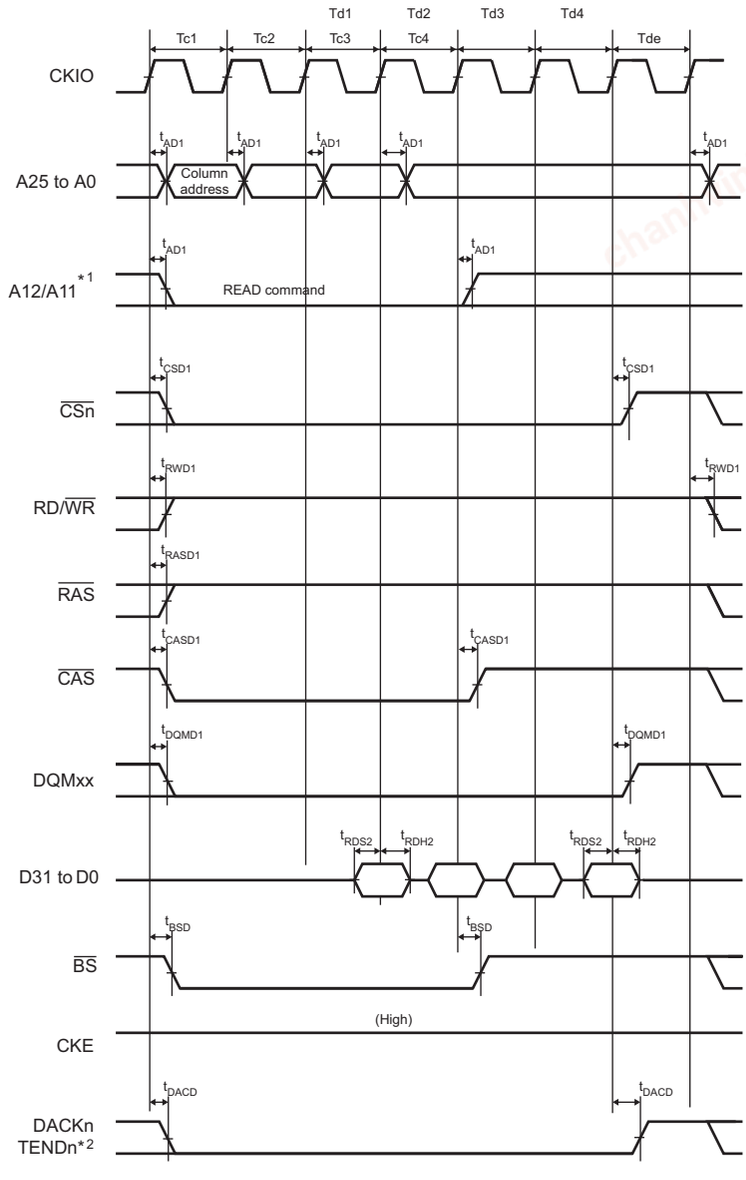
chanhvinh.com



Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 47.24 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles) (Bank Active Mode: ACT + READ Commands, CAS Latency 2, WTRCD = 0 Cycle)

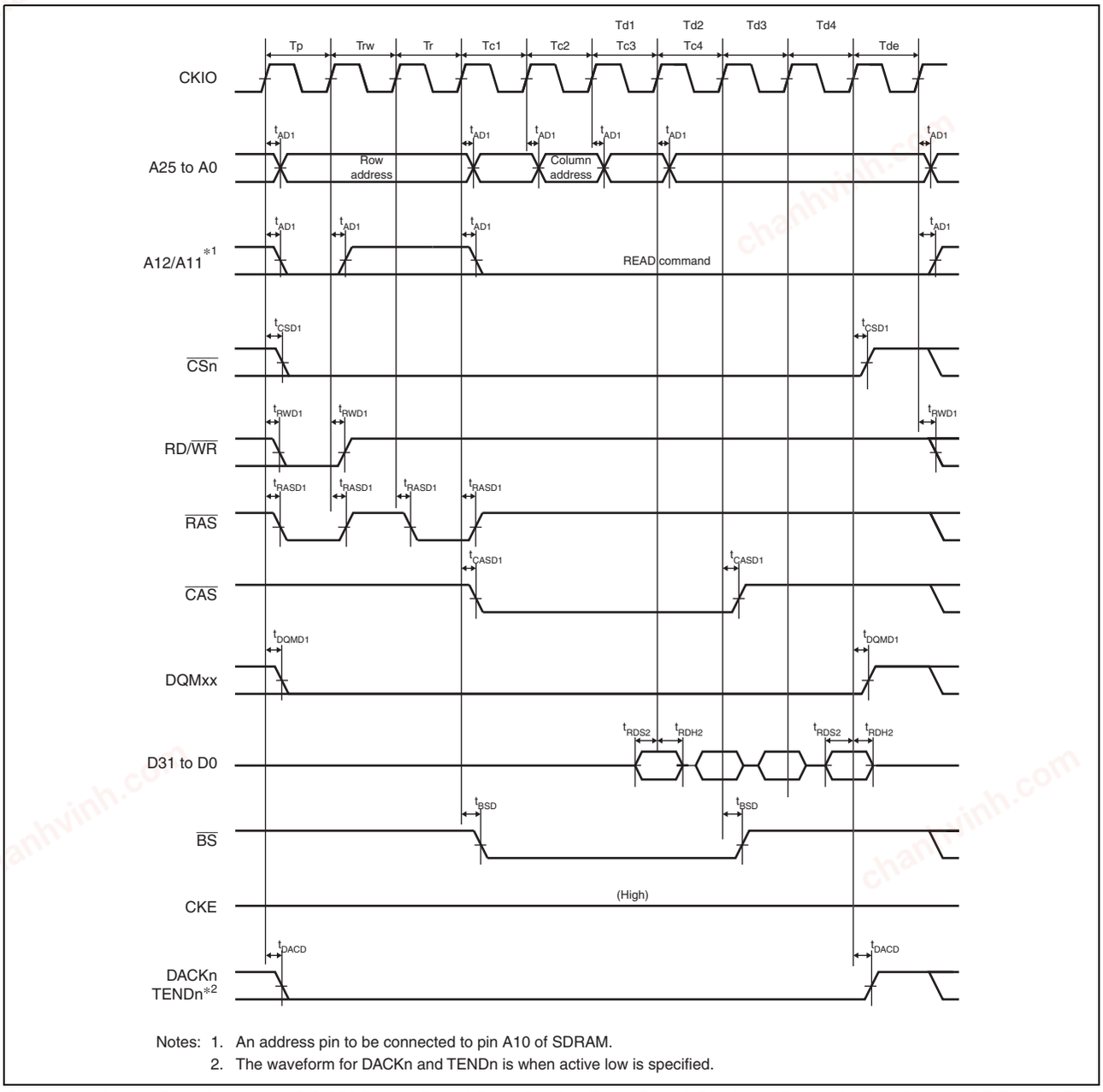
chanhvinh.com



Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 47.25 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles) (Bank Active Mode: READ Command, Same Row Address, CAS Latency 2, WTRCD = 0 Cycle)

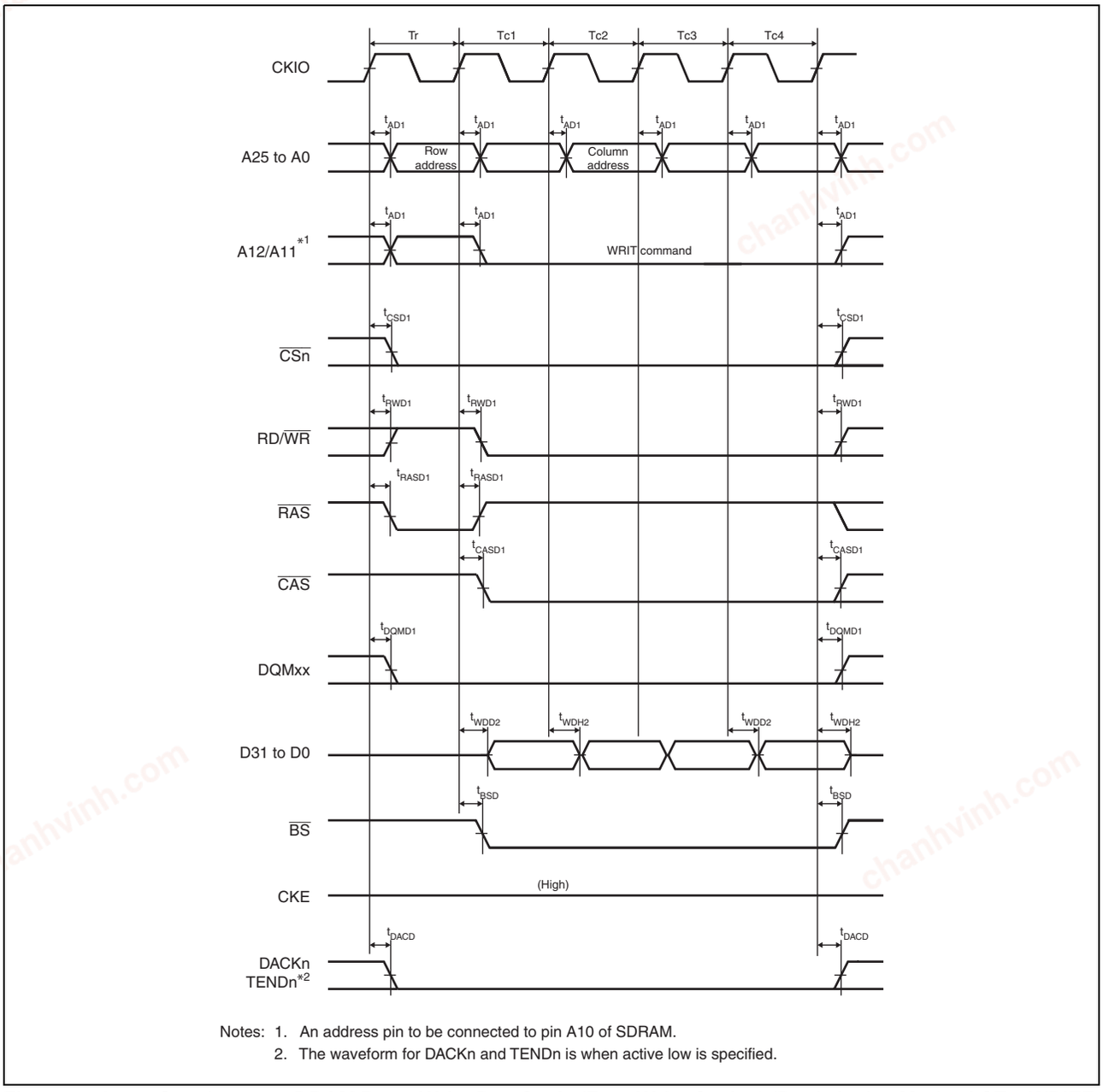
chanhvinh.com



Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 47.26 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles) (Bank Active Mode: PRE + ACT + READ Commands, Different Row Addresses, CAS Latency 2, WTRCD = 0 Cycle)

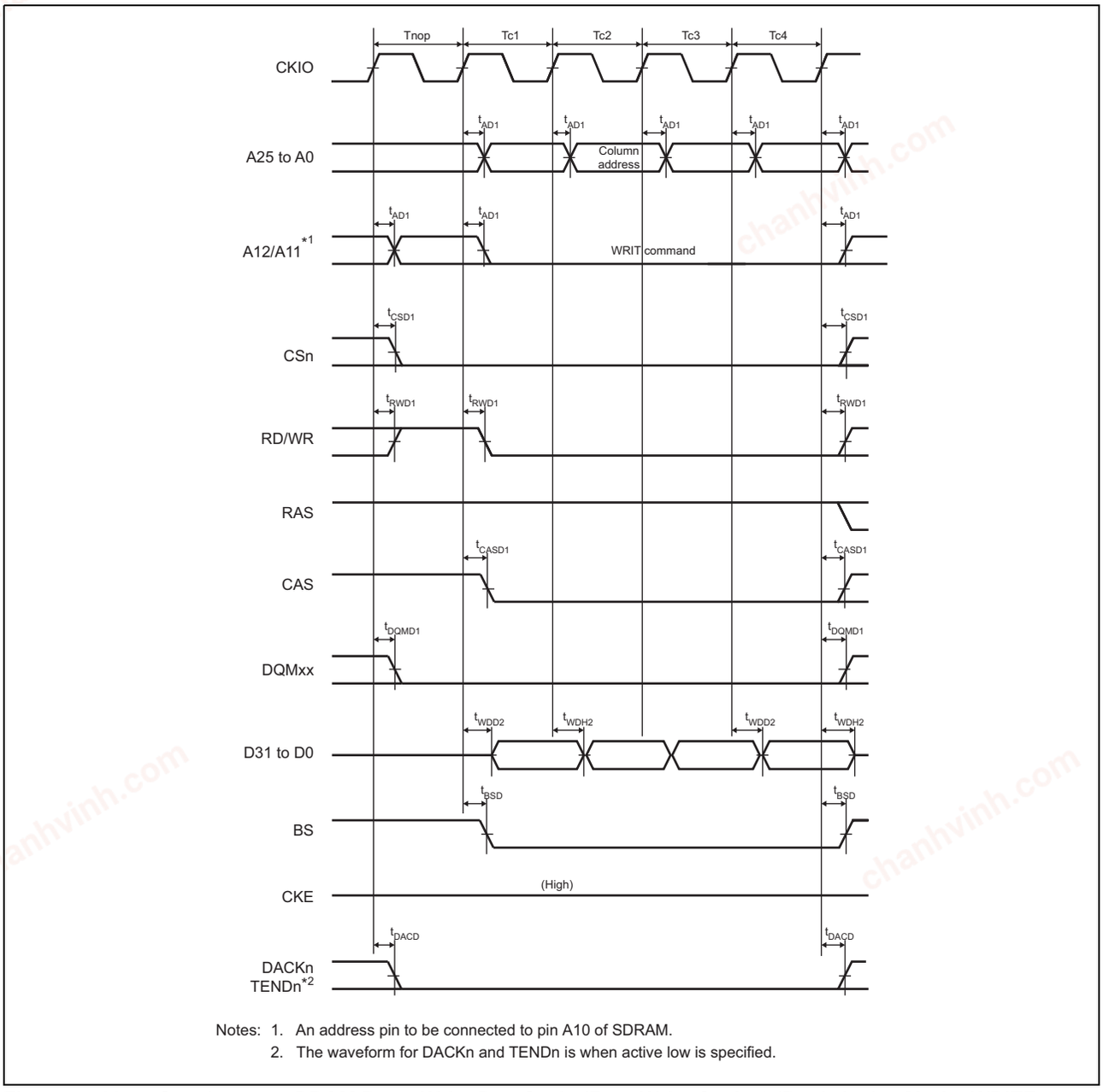
chanhvinh.com



Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 47.27 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Bank Active Mode: ACT + WRITE Commands, WTRCD = 0 Cycle, TRWL = 0 Cycle)

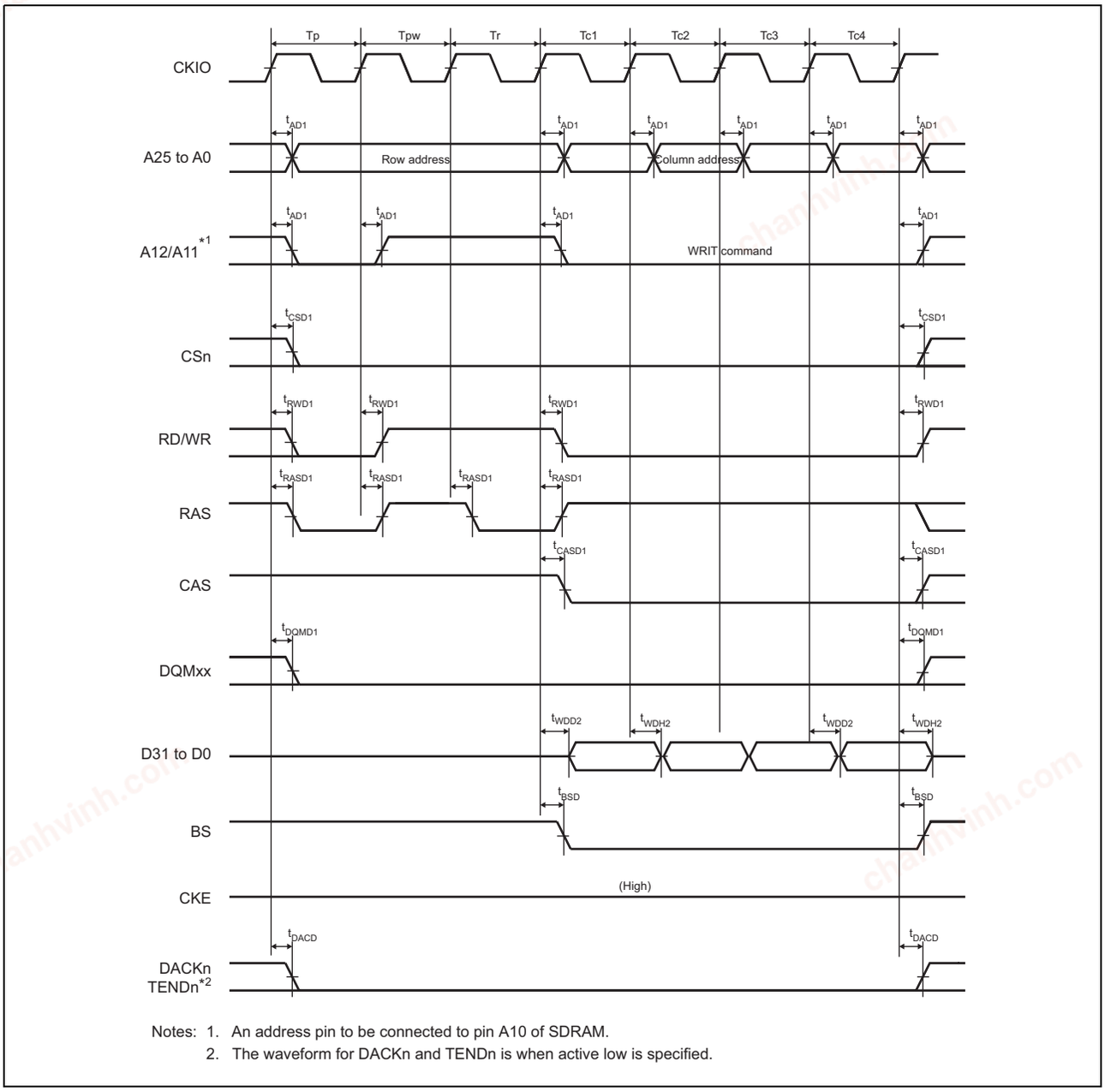
chanhvinh.com



Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 47.28 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Bank Active Mode: WRITE Command, Same Row Address, WTRCD = 0 Cycle, TRWL = 0 Cycle)

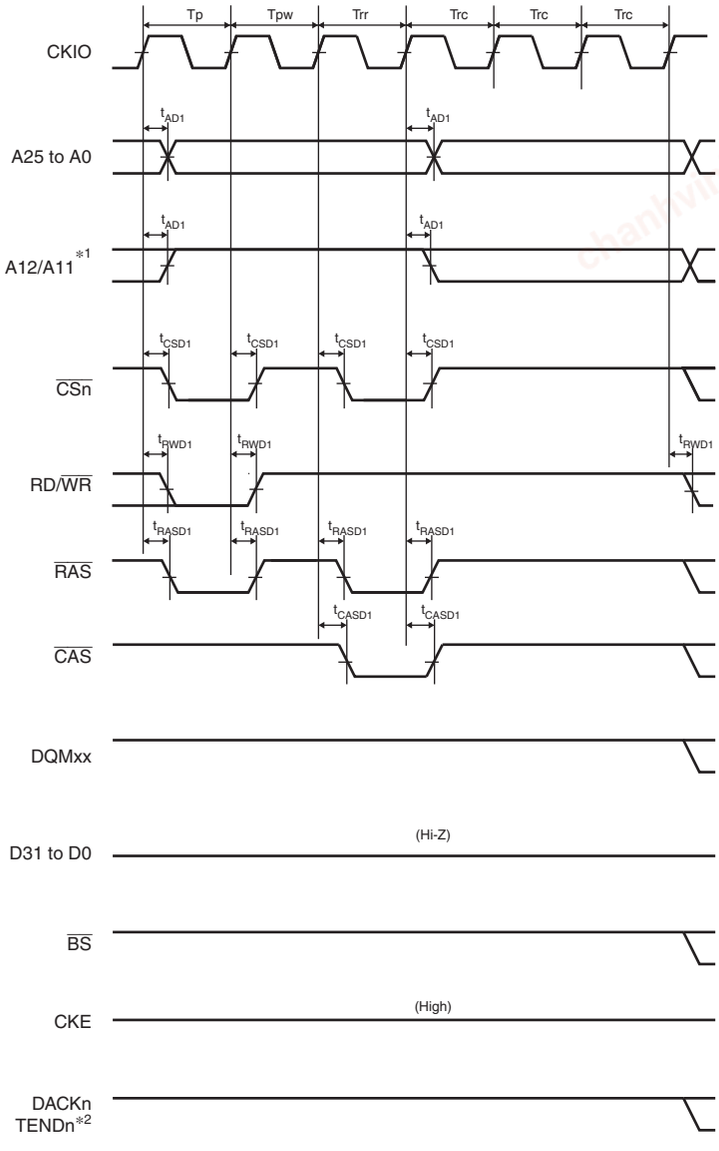
chanhvinh.com



Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 47.29 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Bank Active Mode: PRE + ACT + WRITE Commands, Different Row Addresses, WTRCD = 0 Cycle, TRWL = 0 Cycle)

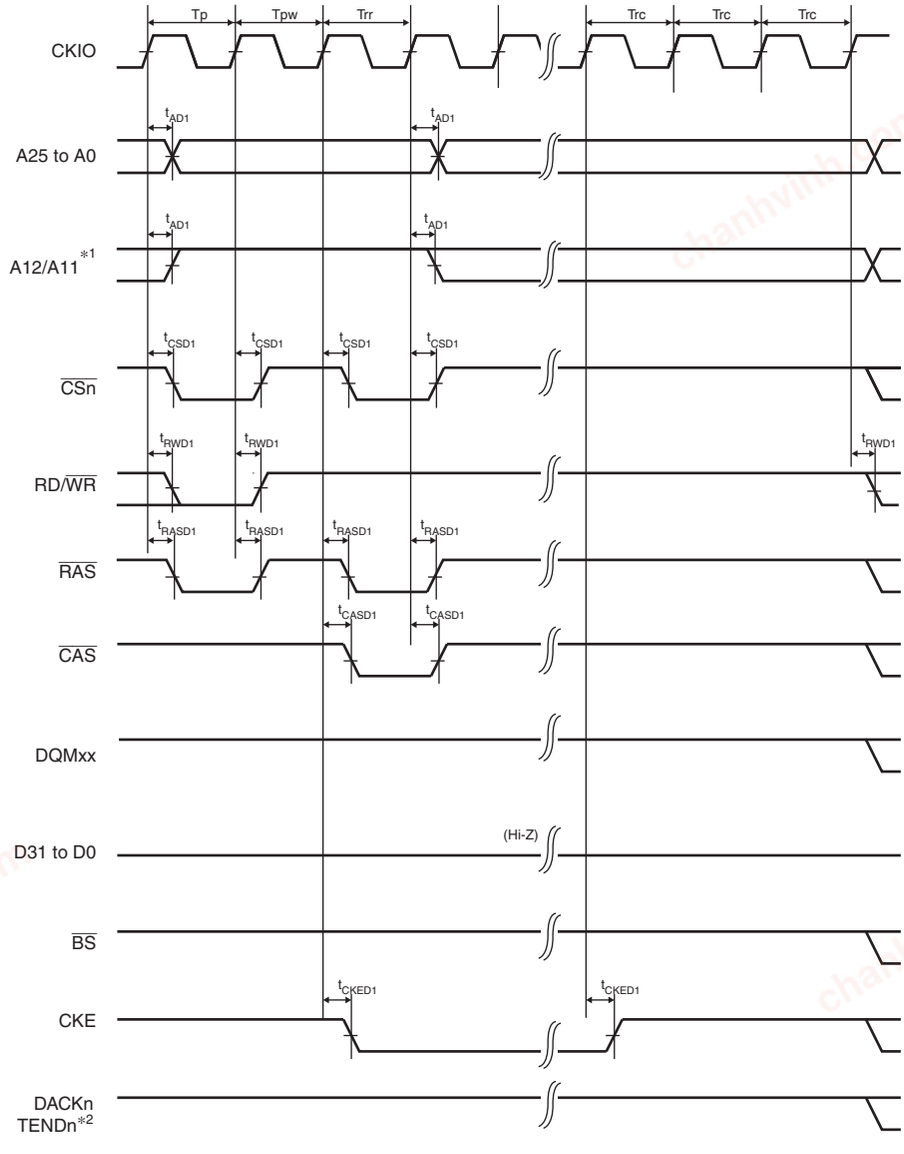
chanhvinh.com



Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 47.30 Synchronous DRAM Auto-Refreshing Timing (WTRP = 1 Cycle, WTRC = 3 Cycles)

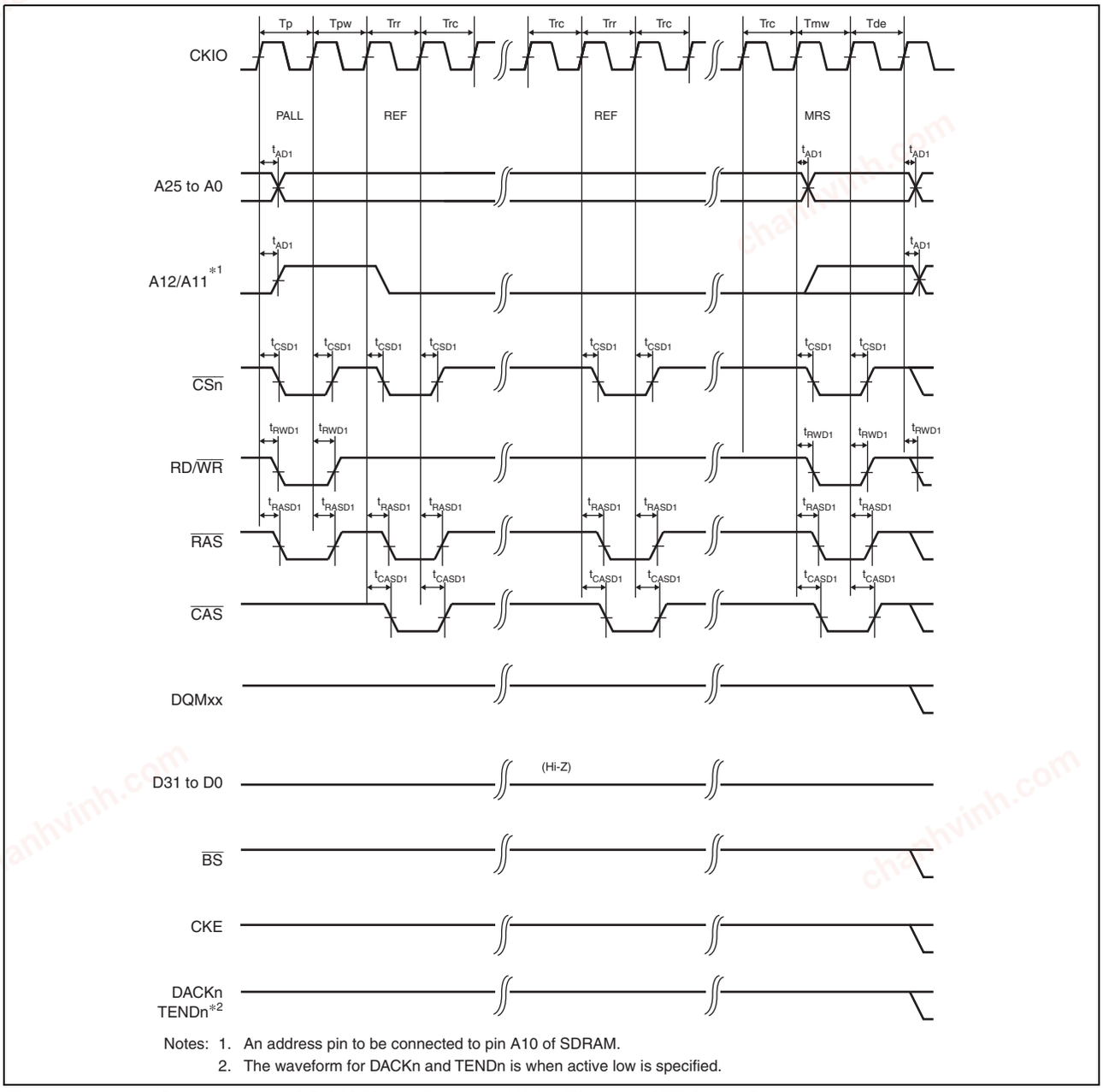
chanhvinh.com



Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 47.31 Synchronous DRAM Self-Refreshing Timing (WTRP = 1 Cycle)

chanhvinh.com



Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 47.32 Synchronous DRAM Mode Register Write Timing (WTRP = 1 Cycle)

chanhvinh.com

47.4.4 Direct Memory Access Controller Timing

Table 47.8 Direct Memory Access Controller Timing

Item	Symbol	Min.	Max.	Unit	Figure
DREQ setup time	t_{DRQS}	5.5	—	ns	Figure 47.33
DREQ hold time	t_{DRQH}	2.5	—		
DACK, TEND delay time	t_{DACD}	0	12		Figure 47.34

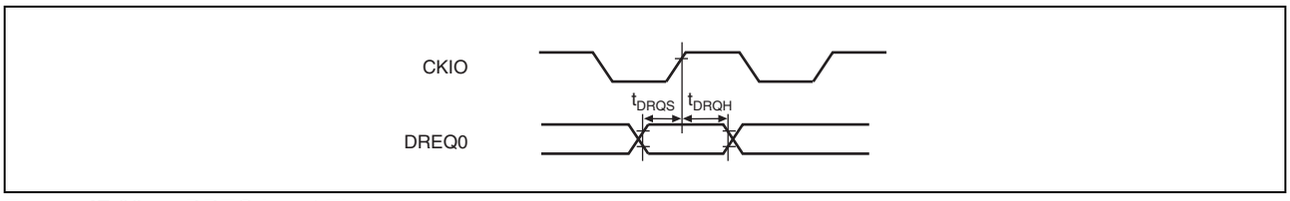


Figure 47.33 DREQ Input Timing

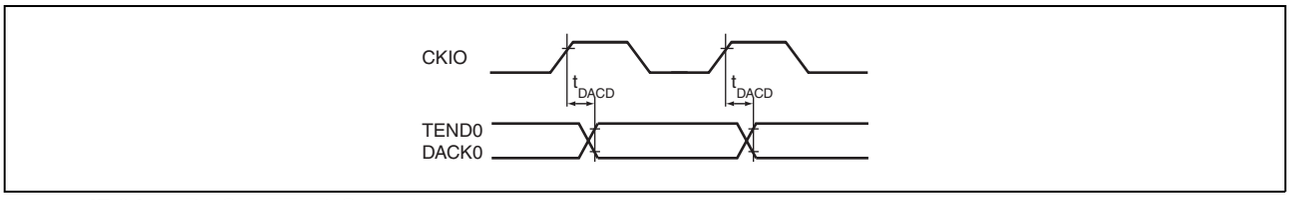


Figure 47.34 DACK, TEND Output Timing

chanhvinh.com

47.4.5 Multi-Function Timer Pulse Unit 2 Timing

Table 47.9 Multi-Function Timer Pulse Unit 2 Timing

Item	Symbol	Min.	Max.	Unit	Figure
Timer clock pulse width (single edge)	$t_{TCKWH/L}$	1.5	—	t_{p0cyc}	Figure 47.35
Timer clock pulse width (both edges)	$t_{TCKWH/L}$	2.5	—	t_{p0cyc}	
Timer clock pulse width (phase counting mode)	$t_{TCKWH/L}$	2.5	—	t_{p0cyc}	

Note: t_{p0cyc} indicates peripheral clock (P0ϕ) cycle.

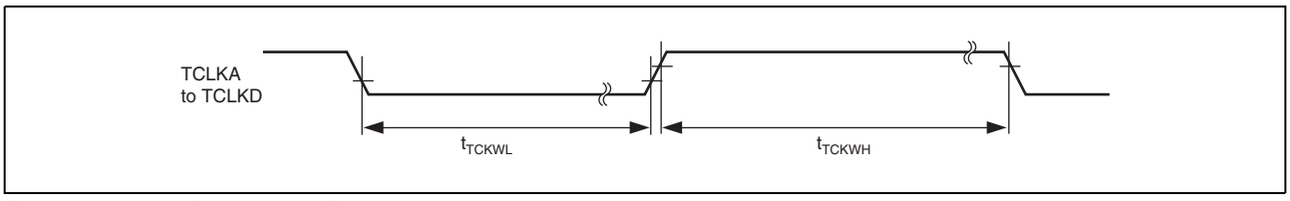


Figure 47.35 Clock Input Timing

chanhvinh.com

47.4.6 Watchdog Timer Timing

Table 47.10 Watchdog Timer Timing

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{WDTOVF}}$ delay time	t_{WOVD}	—	100	ns	Figure 47.36

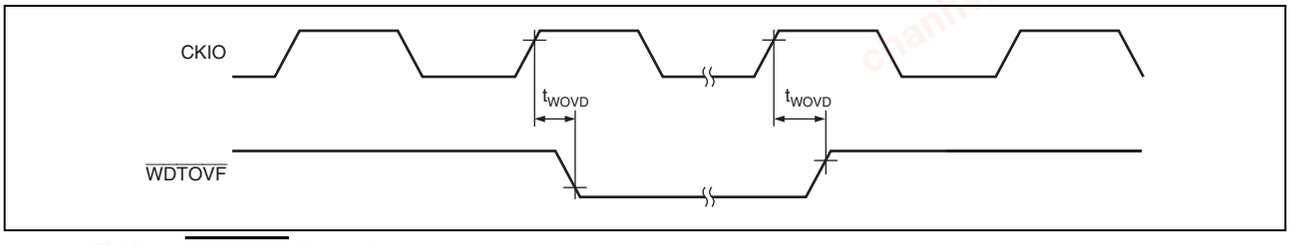


Figure 47.36 $\overline{\text{WDTOVF}}$ Output Timing

chanhvinh.com

47.4.7 Serial Communication Interface with FIFO Timing

Table 47.11 Serial Communication Interface with FIFO Timing

Item		Symbol	Min.	Max.	Unit	Figure
Input clock cycle	(clocked synchronous)	t_{Scyc}	12	—	t_{p1cyc}	Figure 47.37
	(asynchronous)		4	—	t_{p1cyc}	
Input clock rise time		t_{SCKr}	—	1.5	t_{p1cyc}	
Input clock fall time		t_{SCKf}	—	1.5	t_{p1cyc}	
Input clock width		t_{SCKW}	0.4	0.6	t_{Scyc}	
Transmit data delay time (clocked synchronous)		t_{TXD}	—	3 t_{p1cyc} + 15	ns	Figure 47.38
Receive data setup time (clocked synchronous)		t_{RXS}	4 t_{p1cyc} + 15	—	ns	
Receive data hold time (clocked synchronous)		t_{RXH}	1 t_{p1cyc} + 15	—	ns	

Note: t_{p1cyc} indicates the peripheral clock 1 (P1ϕ) cycle.

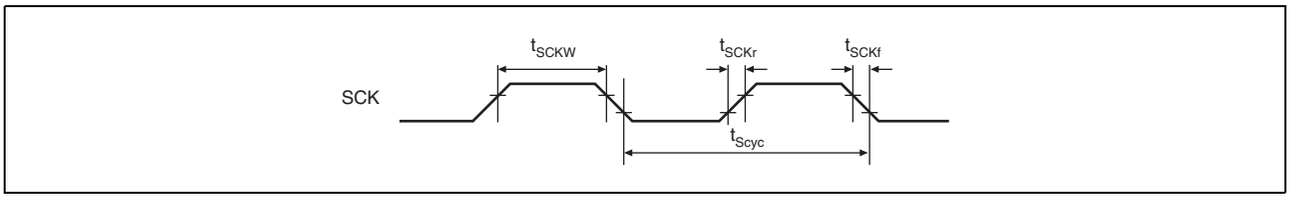


Figure 47.37 SCK Input Clock Timing

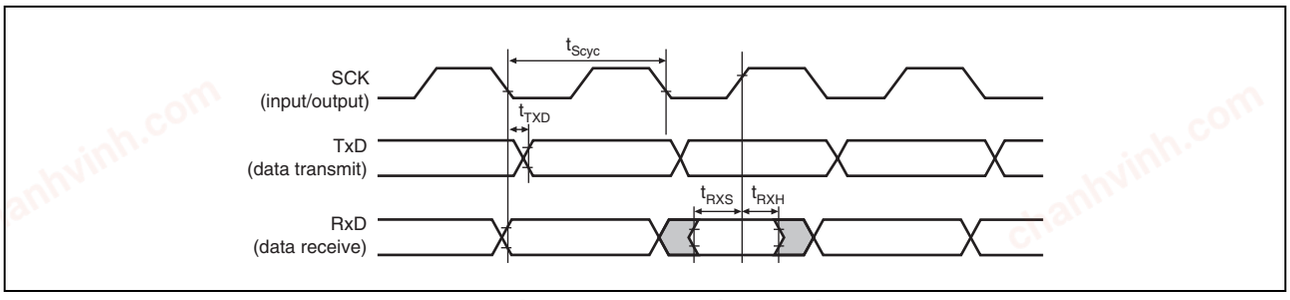


Figure 47.38 Transmit/Receive Data Input/Output Timing in Clocked Synchronous Mode

chanhvinh.com

47.4.8 Serial Communication Interface Timing

Table 47.12 Serial Communication Interface Timing

Item		Symbol	Min.	Max.	Unit	Figure
Input clock cycle	(asynchronous)	t_{Scyc}	4	—	t_{p1cyc}	Figure 47.39
	(clocked synchronous)		6	—		
Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
Input clock rise time		t_{SCKr}	—	20	ns	
Input clock fall time		t_{SCKf}	—	20	ns	
Output clock cycle	(asynchronous)	t_{Scyc}	16	—	t_{p1cyc}	
	(clocked synchronous)		4	—		
Output clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
Output clock rise time		t_{SCKr}	—	20	ns	
Output clock fall time		t_{SCKf}	—	20	ns	
Transmit data delay time	(clocked synchronous)	t_{TXD}	—	40	ns	Figure 47.40
Receive data setup time	(clocked synchronous)	t_{RXS}	40	—	ns	
Receive data hold time	(clocked synchronous)	t_{RXH}	40	—	ns	

Note: t_{p1cyc} indicates the peripheral clock 1 (P1ϕ) cycle.

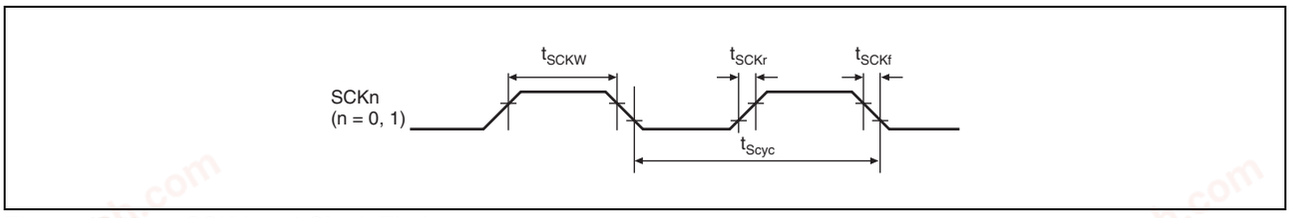


Figure 47.39 SCK Input Clock Timing

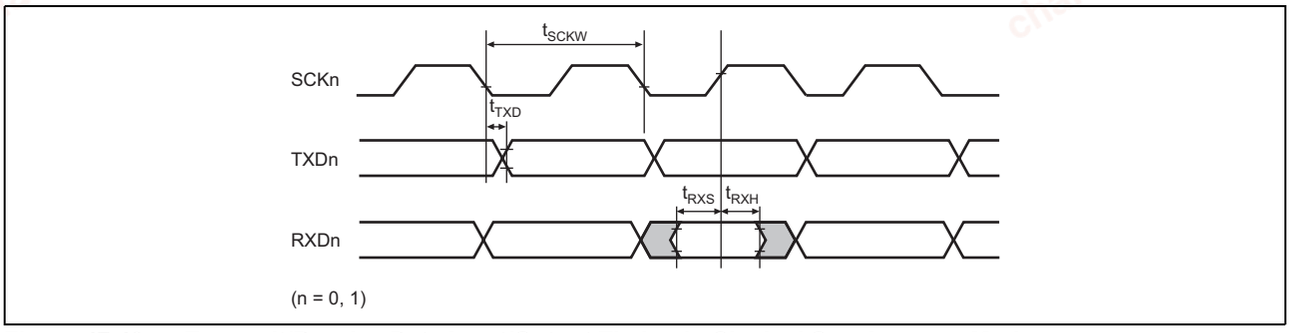


Figure 47.40 Transmit/Receive Data Input/Output Timing in Clocked Synchronous Mode

chanhvinh.com

47.4.9 Renesas Serial Peripheral Interface Timing

Table 47.13 Renesas Serial Peripheral Interface Timing

Item		Symbol	Min.	Max.	Unit	Figure
RSPCK clock cycle	Master	t_{SPcyc}	2	4096	t_{cyc}	Figure 47.41
	Slave		8	4096		
RSPCK clock high pulse width	Master	t_{SPCKWH}	0.4	—	t_{SPcyc}	
	Slave		0.4	—		
RSPCK clock low pulse width	Master	t_{SPCKWL}	0.4	—	t_{SPcyc}	
	Slave		0.4	—		
Data input setup time	Master	t_{SU}	15	—	ns	Figure 47.42 to Figure 47.45
	Slave		0	—	t_{cyc}	
Data input hold time	Master	t_H	0	—	ns	
	Slave		4	—	t_{cyc}	
SSL setup time	Master	t_{LEAD}	$1 \times t_{SPcyc} - 20$	$8 \times t_{SPcyc}$	ns	
	Slave		4	—	t_{cyc}	
SSL hold time	Master	t_{LAG}	$1 \times t_{SPcyc}$	$8 \times t_{SPcyc} + 20$	ns	
	Slave		4	—	t_{cyc}	
Data output delay time	Master	t_{OD}	—	21	ns	
	Slave		—	4	t_{cyc}	
Data output hold time	Master	t_{OH}	5	—	ns	
	Slave		3	—	t_{cyc}	
Continuous transmission delay time	Master	t_{TD}	$1 \times t_{SPcyc} + 2\ t_{cyc}$	$8 \times t_{SPcyc} + 2 \times t_{cyc}$	ns	
	Slave		$4 \times t_{cyc}$	—		
Slave access time		t_{SA}	—	4	t_{cyc}	Figure 47.44 and Figure 47.45
Slave out release time		t_{REL}	—	3	t_{cyc}	

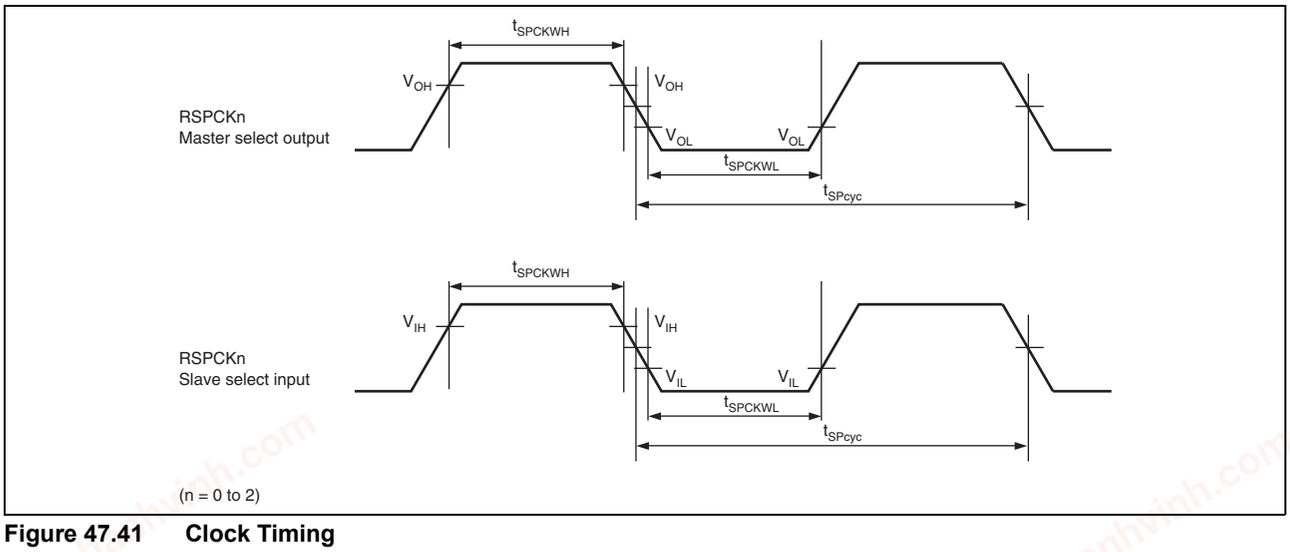


Figure 47.41 Clock Timing

chanhvinh.com

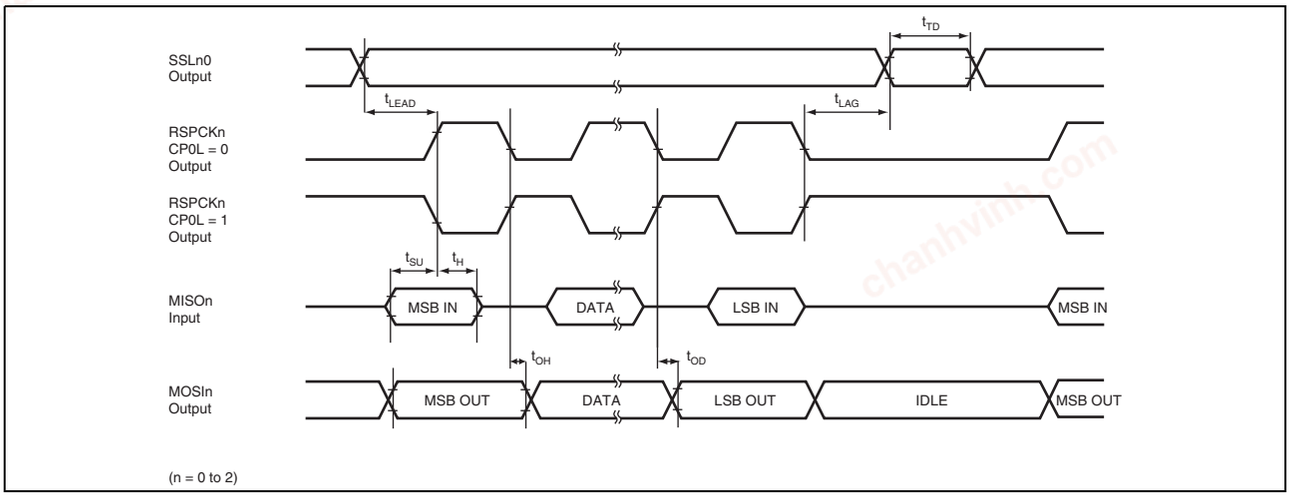


Figure 47.42 Transmission and Reception Timing (Master, CPHA = 0)

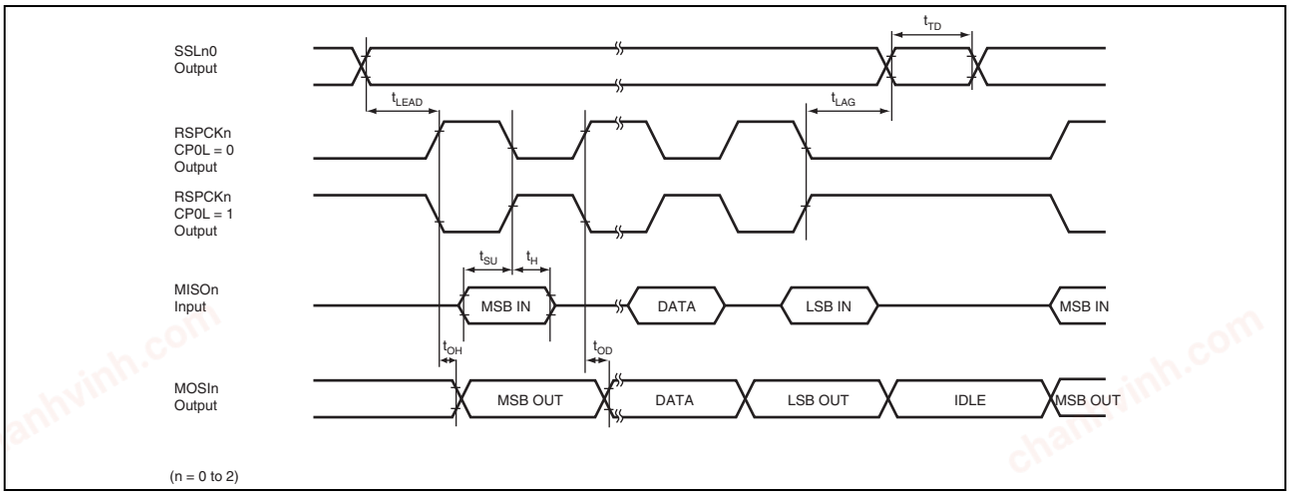


Figure 47.43 Transmission and Reception Timing (Master, CPHA = 1)

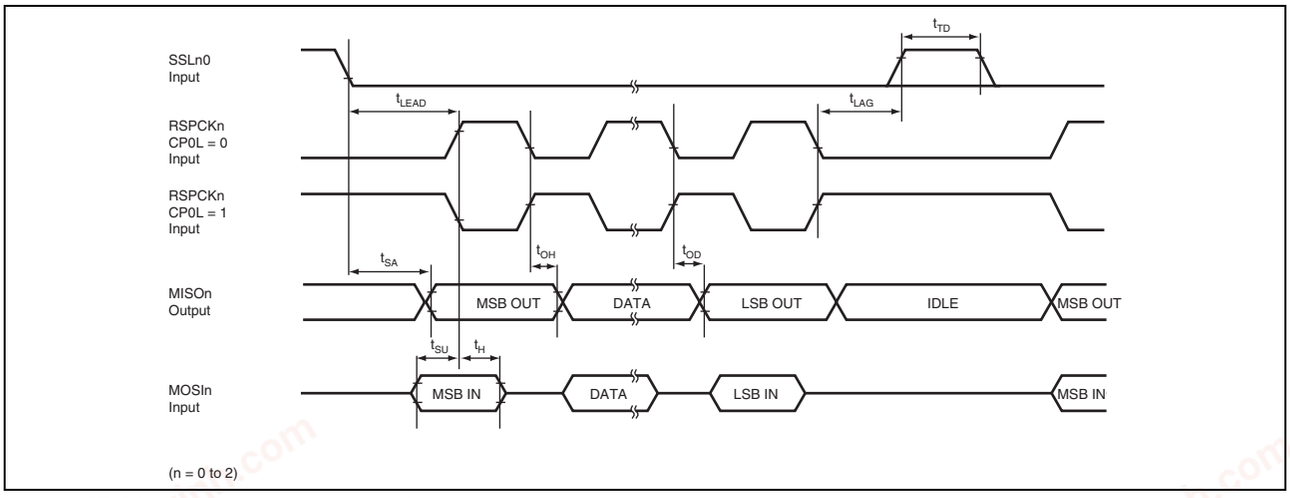


Figure 47.44 Transmission and Reception Timing (Slave, CPHA = 0)

chanhvinh.com

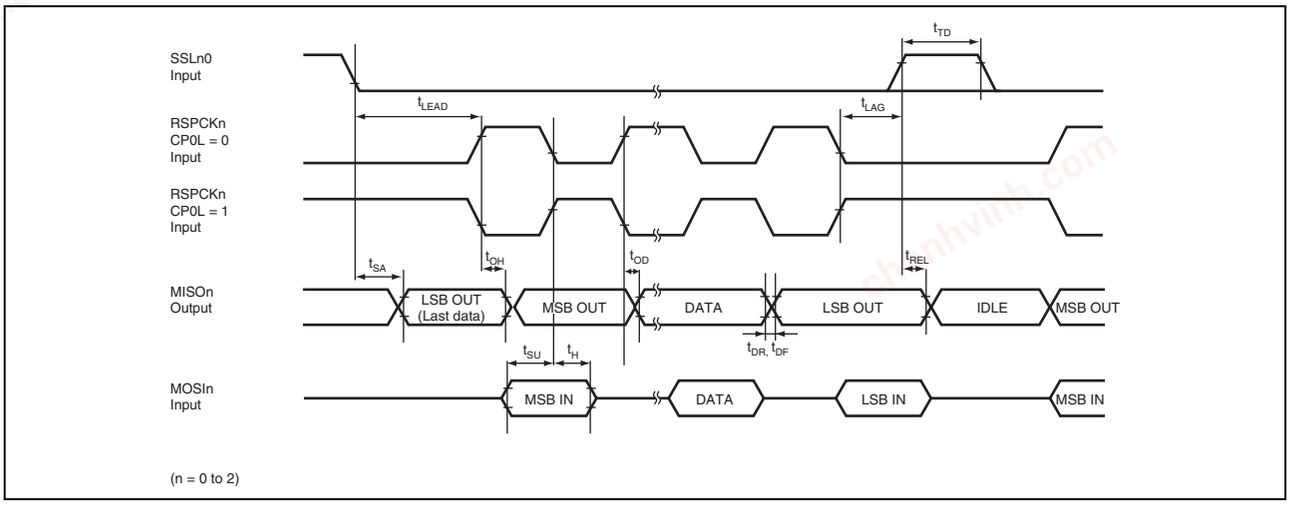


Figure 47.45 Transmission and Reception Timing (Slave, CPHA = 1)

chanhvinh.com

47.4.10 SPI Multi I/O Bus Controller Timing

Table 47.14 SPI Multi I/O Bus Controller Timing

Item		Symbol	Min.	Max.	Unit	Figure
SPBCLK clock cycle		t_{SPBcyc}	2	4080	t_{bcyc}	Figure 47.46
SPBCLK high pulse width		t_{SPBWH}	0.475	0.525	t_{SPBcyc}	
SPBCLK low pulse width		t_{SPBWL}	0.475	0.525	t_{SPBcyc}	
SPBCLK rise time		t_{SPBR}	—	3	ns	
SPBCLK fall time		t_{SPBF}	—	3	ns	
Data input setup time	CKDLY = B'0100 (initial value)	t_{SU}	5.0	—	ns	Figure 47.47, Figure 47.48, Figure 47.49, and Figure 47.50
	CKDLY = B'1010 (RZ/A1LU only)		2.0*2	—		
Data input hold time	CKDLY = B'0100	t_H	0.0	—	ns	
	CKDLY = B'1010 (RZ/A1LU only)		1.0*2	—		
SSL setup time		t_{LEAD}	1 × t_{SPBcyc} - 3	8 × t_{SPBcyc} + 3	ns	
SSL hold time		t_{LAG}	1.5 × t_{SPBcyc}	8.5 × t_{SPBcyc} + 3	ns	
Continuous transfer delay time		t_{TD}	1	8	t_{SPBcyc}	
Data output delay time	SPODLY = H'0000 (initial value)	t_{OD}	—	4.0	ns	
	SPODLY = H'1111 (RZ/A1LU only)		—	5.0*2		
Data output hold time	SPODLY = H'0000 (initial value)	t_{OH}	–2.0	—	ns	
	SPODLY = H'1111 (RZ/A1LU only)		1.0*2	—		
Data output buffer on time	SPODLY = H'0000 (initial value)	t_{BON}	—	4.0	ns	Figure 47.47, Figure 47.48, Figure 47.49, Figure 47.50, Figure 47.51, and Figure 47.52
	SPODLY = H'1111 (RZ/A1LU only)		—	5.0*2		
Data output buffer off time	SPODLY = H'0000 (initial value)	t_{BOFF}	–9.0	0	ns	
	SPODLY = H'1111 (RZ/A1LU only)		1.5*2	7.5*2		

Note 1. t_{bcyc} indicates the bus clock (Bϕ) cycle.
Note 2. Value when the output load is 15 pF

chanhvinh.com

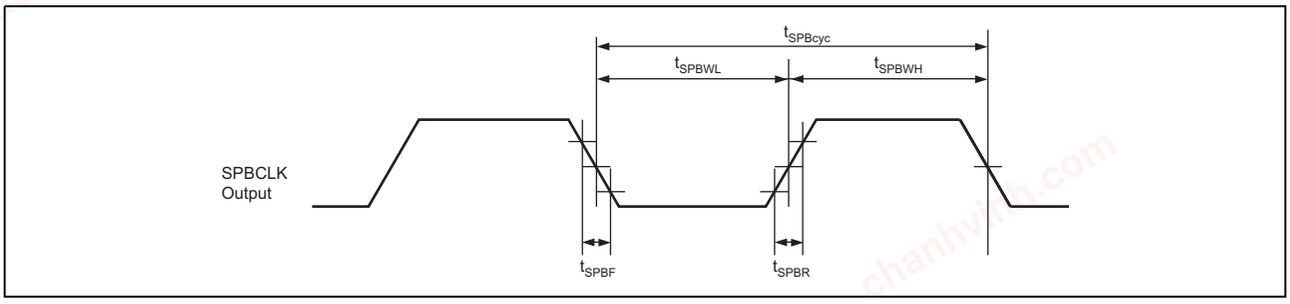


Figure 47.46 Clock Timing

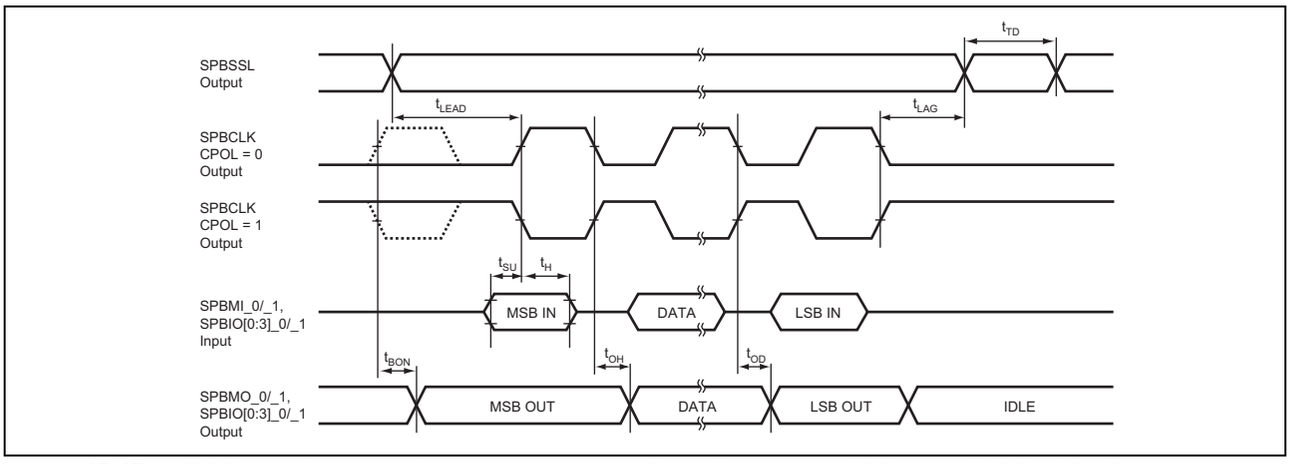


Figure 47.47 SDR Transfer Format Transmission and Reception Timing (CPHAT = 0, CPHAR = 0)

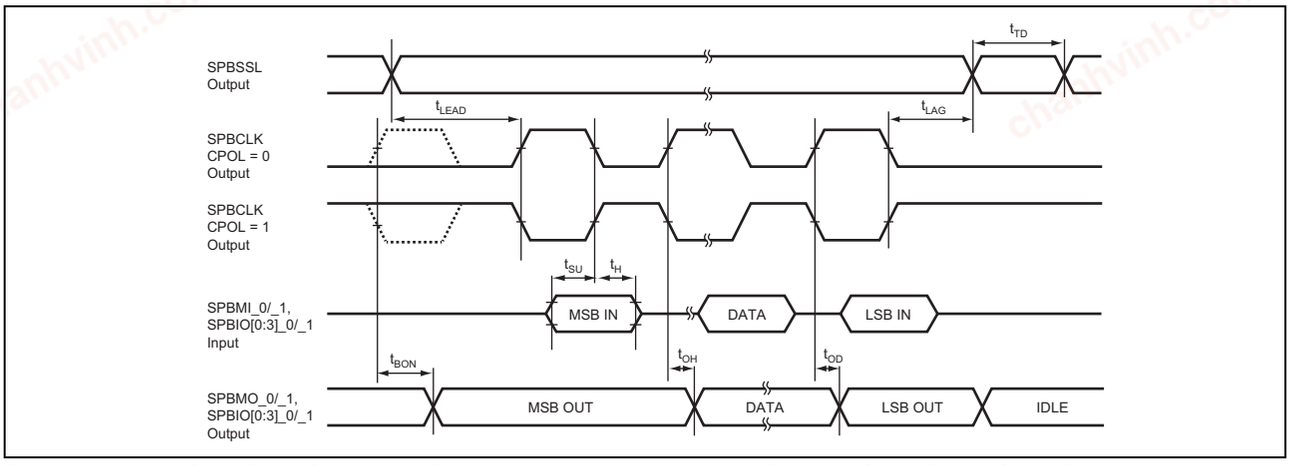


Figure 47.48 SDR Transfer Format Transmission and Reception Timing (CPHAT = 1, CPHAR = 1)

chanhvinh.com

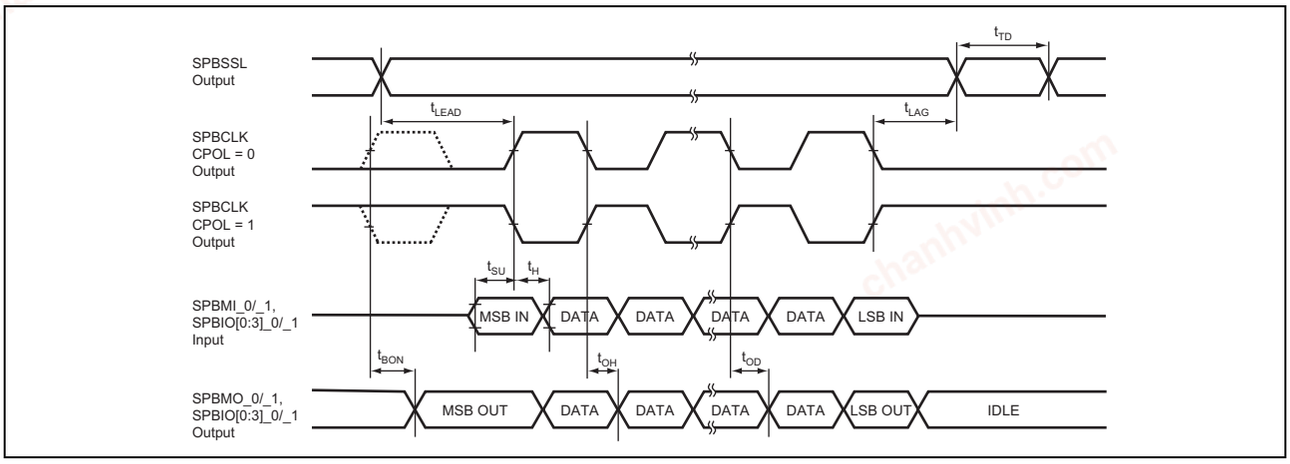


Figure 47.49 DDR Transfer Format Transmission and Reception Timing (CPHAT = 0, CPHAR = 0) (RZ/A1LU only)

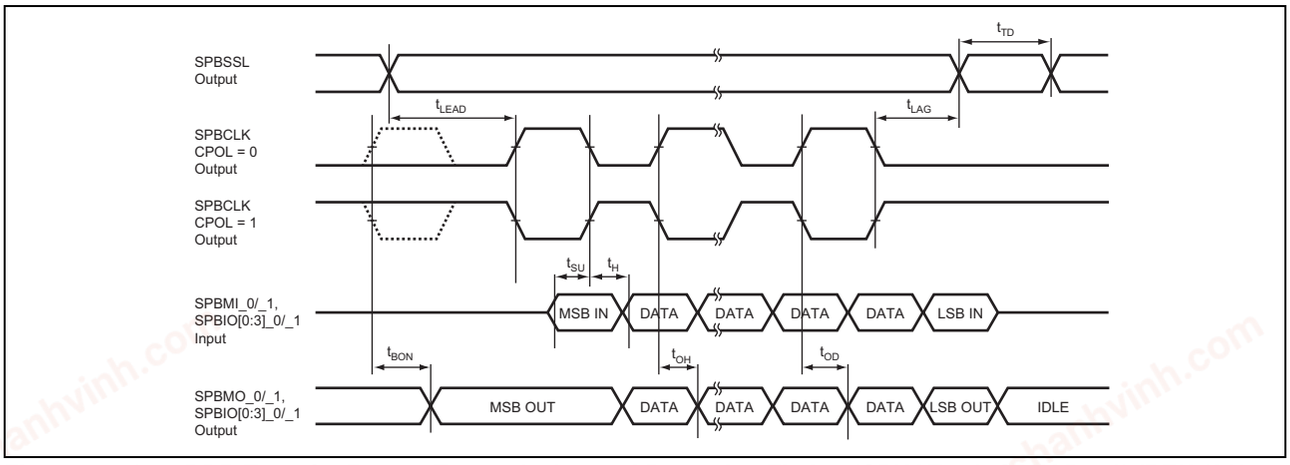


Figure 47.50 DDR Transfer Format Transmission and Reception Timing (CPHAT = 1, CPHAR = 1) (RZ/A1LU only)

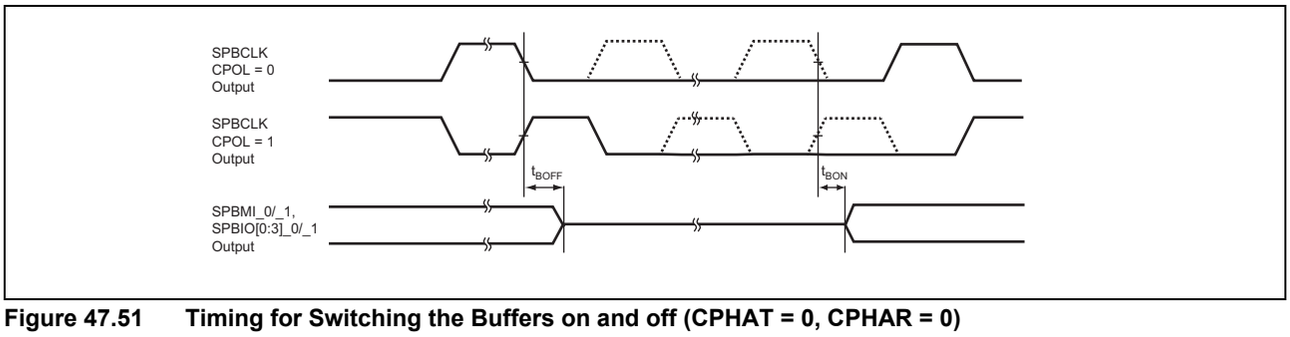


Figure 47.51 Timing for Switching the Buffers on and off (CPHAT = 0, CPHAR = 0)

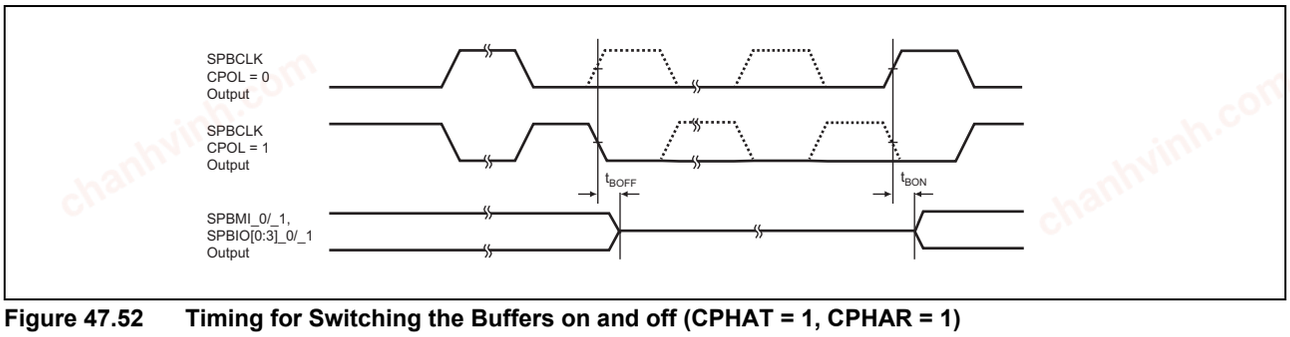


Figure 47.52 Timing for Switching the Buffers on and off (CPHAT = 1, CPHAR = 1)

chanhvinh.com

47.4.11 I²C Bus Interface Timing

Table 47.15 I²C Bus Interface Timing

Item	Symbol	I/O	Standard mode (Sm) Min.	Standard mode (Sm) Max.	Fast mode (Fm) Min.	Fast mode (Fm) Max.	Unit
SCL clock frequency	f_{CLK}	I/O	0	100	0	400	kHz
Bus free time (between stop and start condition)	t_{BUF}	I/O	4.7	-	1.3	-	µs
Hold time*1	$t_{HD:STA}$	I/O	4.0	-	0.6	-	µs
Low period of SCL clock	t_{LOW}	I/O	4.7	-	1.3	-	µs
High period of SCL clock	t_{HIGH}	I/O	4.0	-	0.6	-	µs
Setup time for start/restart condition	$t_{SU:STA}$	I/O	4.7	-	0.6	-	µs
Data hold time (I²C bus device)	$t_{HD:DAT}$	I/O	0*2	-	0*2	-	µs
Data setup time	$t_{SU:DAT}$	I/O	250	-	100*3	-	ns
SDA and SCL signal rise time	t_R	Input	-	1000	20	300	ns
SDA and SCL signal fall time*3	t_F	Input	-	300	20 × (PV_{CC}/5.5V)	300	ns
		Output	-	250	20 × (PV_{CC}/5.5V)	250	ns
Setup time for STOP condition	$t_{SU:STO}$	I/O	4.0	-	0.6	-	µs
Capacitive load for each bus line	C_b	-	-	400*4	-	400*4	pF
Pulse width of spikes that must be suppressed by the input filter	t_{SP}	Input	-	-	0	50*5	ns

In the above table and subsequently, SCL and SDA refer to the RIICnSCL and RIICnSDA signals, respectively.

Note 1. The first clock pulse is generated on the SCL line after the start condition has been issued and the hold time has elapsed.

Note 2. This module requires a minimum of 300 ns hold time internally for the SDA signal to handle the period over which the falling edge of SCL has not reached a defined level (time until the SCL signal reaches V_{IL} (max.) from V_{IH} (min.)).

Note 3. The fast-mode I²C bus device can be used in the standard mode I²C bus system. In this case, the minimum value of the data setup time ($t_{SU:DAT}$ 250 [ns]) must be satisfied.
If the system does not extend the low period of SCL clock (t_{LOW}), this condition is automatically satisfied. If the system extends the low period of SCL clock (t_{LOW}), transmit the subsequent data bit to the SDA line before the SCL line is released ($t_R$max. + $t_{SU:DAT}$ = 1000 + 250 = 1250 [ns]: (standard mode I²C bus specification)).

Note 4. Total capacitance of one bus line. The allowable maximum bus capacitance may differ from this specification, depending on the actual operating voltage and frequency of an application. For techniques to cope with a large bus capacitance, see the I²C bus specification provided by NXP Semiconductors.

Note 5. Noise is removed by the analog and digital input filters. The level of noise reduction of the digital input filter is determined by the period of internal reference clock (IICφ) and the NF[1:0] bits in RIICnMR3. For details, refer to section 18, I²C Bus Interface.

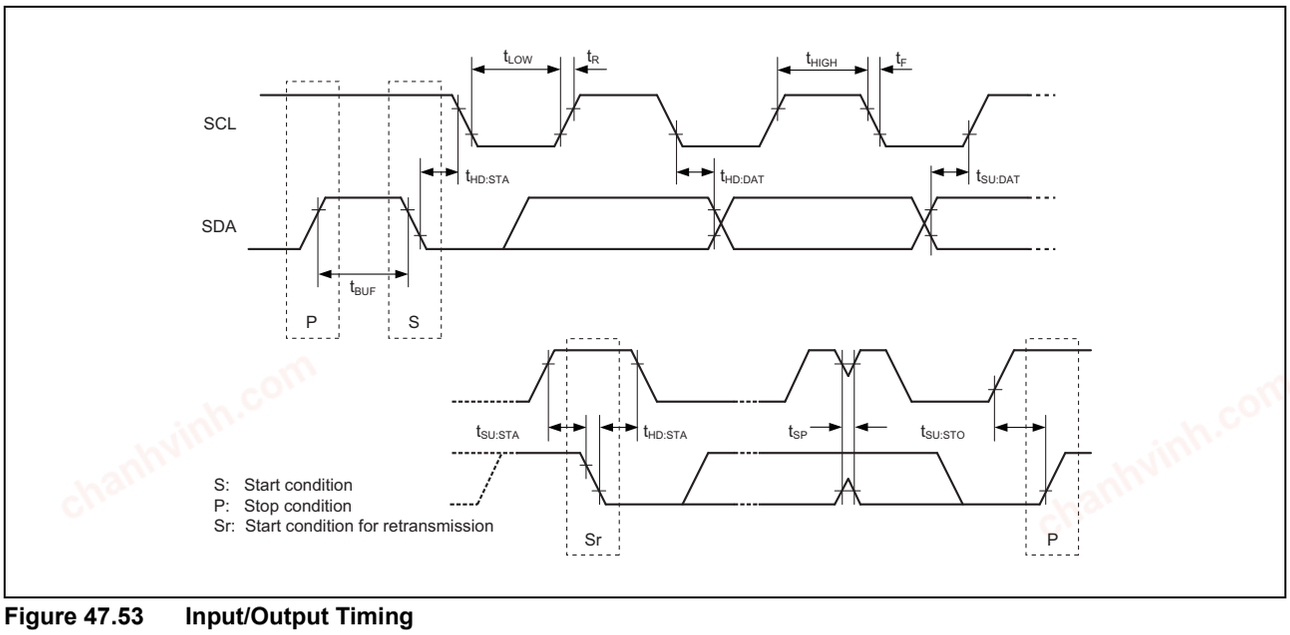


Figure 47.53 Input/Output Timing

chanhvinh.com

47.4.12 Serial Sound Interface Timing

Table 47.16 Serial Sound Interface Timing

Item		Symbol	Min.	Max.	Unit	Remarks	Figure
Output clock cycle		t_O	80	64000	ns	Output	Figure 47.54
Input clock cycle		t_I	80	64000	ns	Input	
Clock high		t_{HC}	32	—	ns	Bidirectional	
Clock low		t_{LC}	32	—	ns		
Clock rise time		t_{RC}	—	25	ns	Output	
Delay	Noise canceler not in use	t_{DTR}	–5	25	ns		Figure 47.55, Figure 47.56, Figure 47.57, Figure 47.58, and Figure 47.59
	Noise canceler in use		10	45			
SSIWS delay (RZ/A1LU and RZ/A1LC only)*			–5	15			
Setup time		t_{SR}	25	—	ns		
Hold time		t_{HTR}	5	—	ns		

Note: * For SSIWS delay of the RZ/A1L, refer to the values in the entry on “Delay (Noise canceler not in use)”.

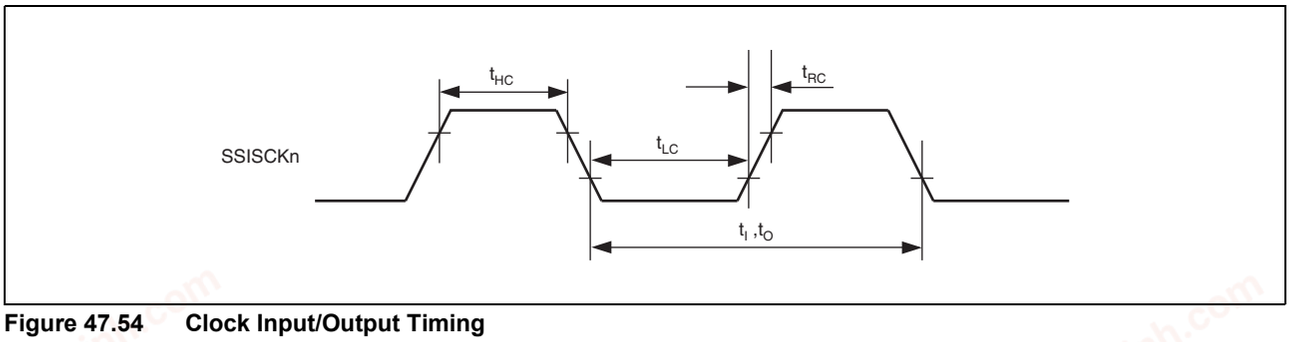


Figure 47.54 Clock Input/Output Timing

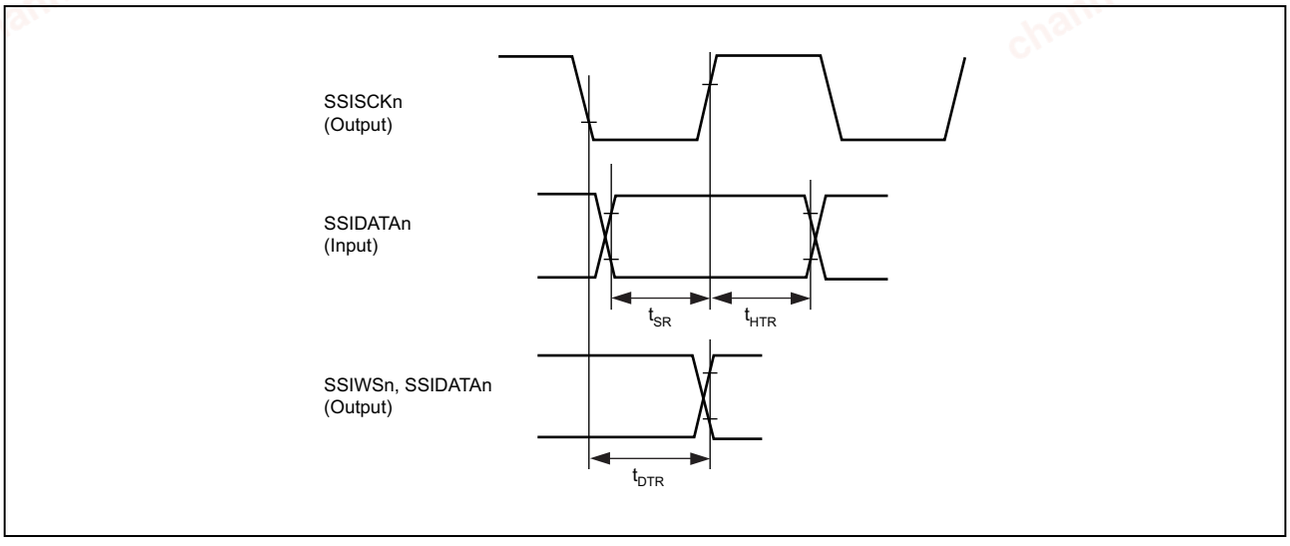


Figure 47.55 Transmission and Reception Timing (Master, SSICR_n.SCKP = 0)

chanhvinh.com

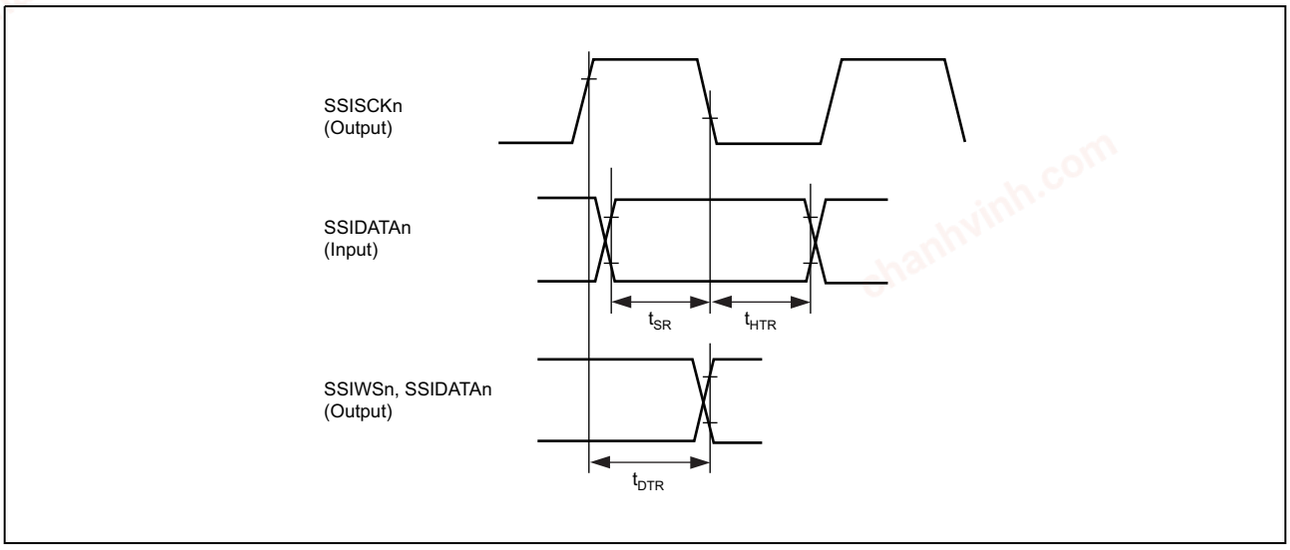


Figure 47.56 Transmission and Reception Timing (Master, SSICR_n.SCKP = 1)

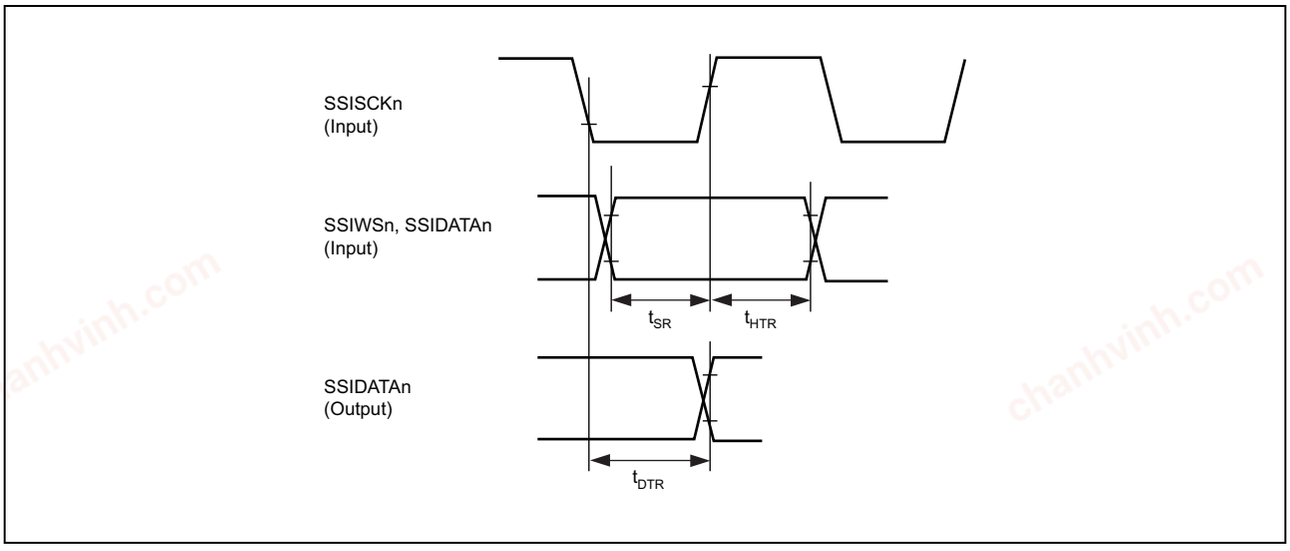


Figure 47.57 Transmission and Reception Timing (Slave, SSICR_n.SCKP = 0)

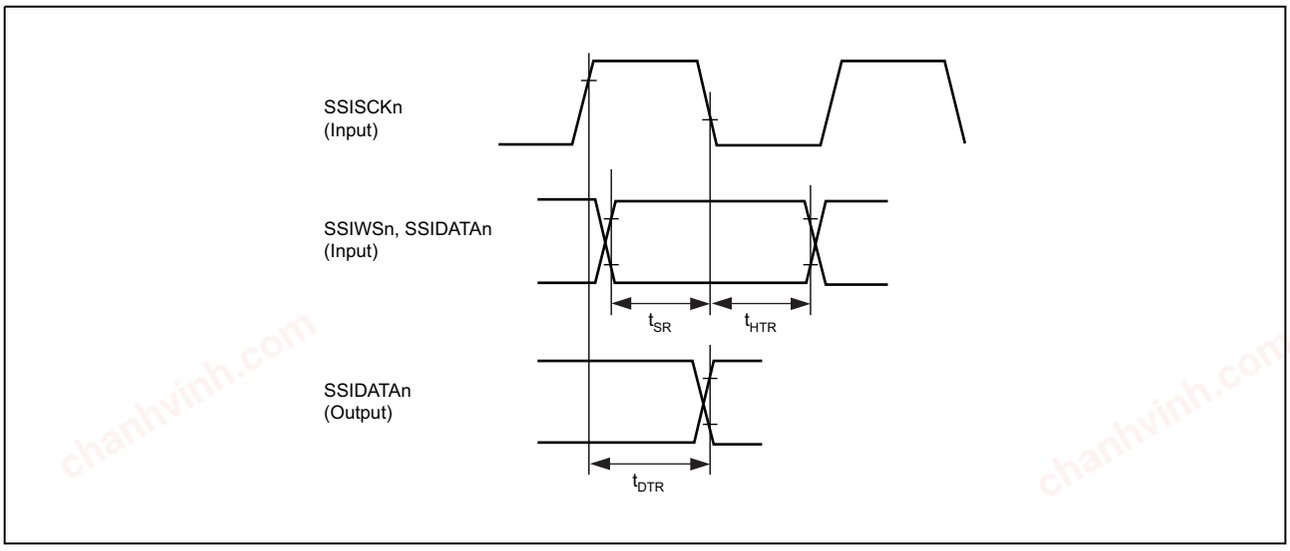
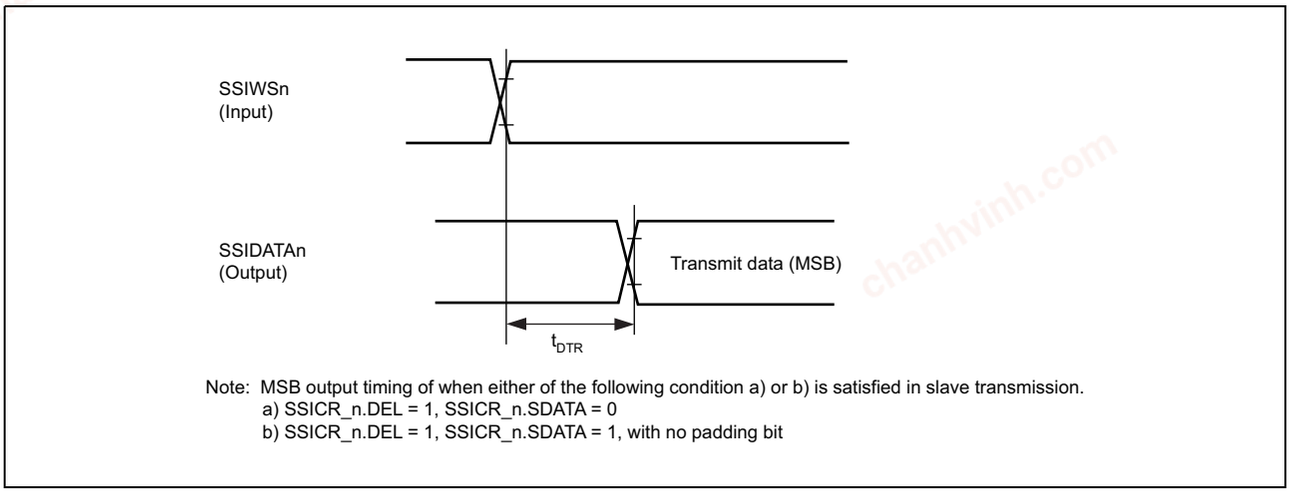


Figure 47.58 Transmission and Reception Timing (Slave, SSICR_n.SCKP = 1)

chanhvinh.com



Note: MSB output timing of when either of the following condition a) or b) is satisfied in slave transmission.
a) SSICR_n.DEL = 1, SSICR_n.SDATA = 0
b) SSICR_n.DEL = 1, SSICR_n.SDATA = 1, with no padding bit

Figure 47.59 Transmission Timing (Slave, in Synchronization with SSIWSn)

chanhvinh.com

47.4.13 Media Local Bus Timing

This module is only incorporated in the RZ/A1L.

Table 47.17 Media Local Bus Timing

Item	Symbol	Min.	Typ	Max.	Unit	Remarks	Figure
Input clock frequency (256 × FS)	f_I	11.2640	12.2880	12.3136	MHz		Figure 47.60
Input clock cycle (256 × FS)	t_I	—	81	—	ns		
Input clock high level (256 × FS)	t_{HC}	30	36.5	—	ns		
Input clock low level (256 × FS)	t_{LC}	30	35.5	—	ns		
Input clock frequency (512 × FS)	f_I	22.5280	24.5760	24.6272	MHz		
Input clock cycle (512 × FS)	t_I	—	40	—	ns		
Input clock high level (512 × FS)	t_{HC}	14	16.5	—	ns		
Input clock low level (512 × FS)	t_{LC}	14	16.5	—	ns		
Input clock frequency (1024 × FS)	f_I	45.0560	49.1520	49.2544	MHz		
Input clock cycle (1024 × FS)	t_I	—	20.3	—	ns		
Input clock high level (1024 × FS)	t_{HC}	9.3	10.2	—	ns		
Input clock low level (1024 × FS)	t_{LC}	6.1	7.3	—	ns		
Input clock rise time	t_{RC}	—	—	1	ns	V_{IL} to V_{IH}	
Input clock fall time	t_{FC}	—	—	1	ns	V_{IH} to V_{IL}	
Delay time (clock signal rising)	t_{DTR}	—	—	8.0	ns	Output load: 20 pF	
Delay time (clock signal falling)	t_{DTF}	0	—	t_{LC}	ns		
Setup time	t_{SR}	1	—	—	ns		
Hold time	t_{HTR}	2	—	—	ns		

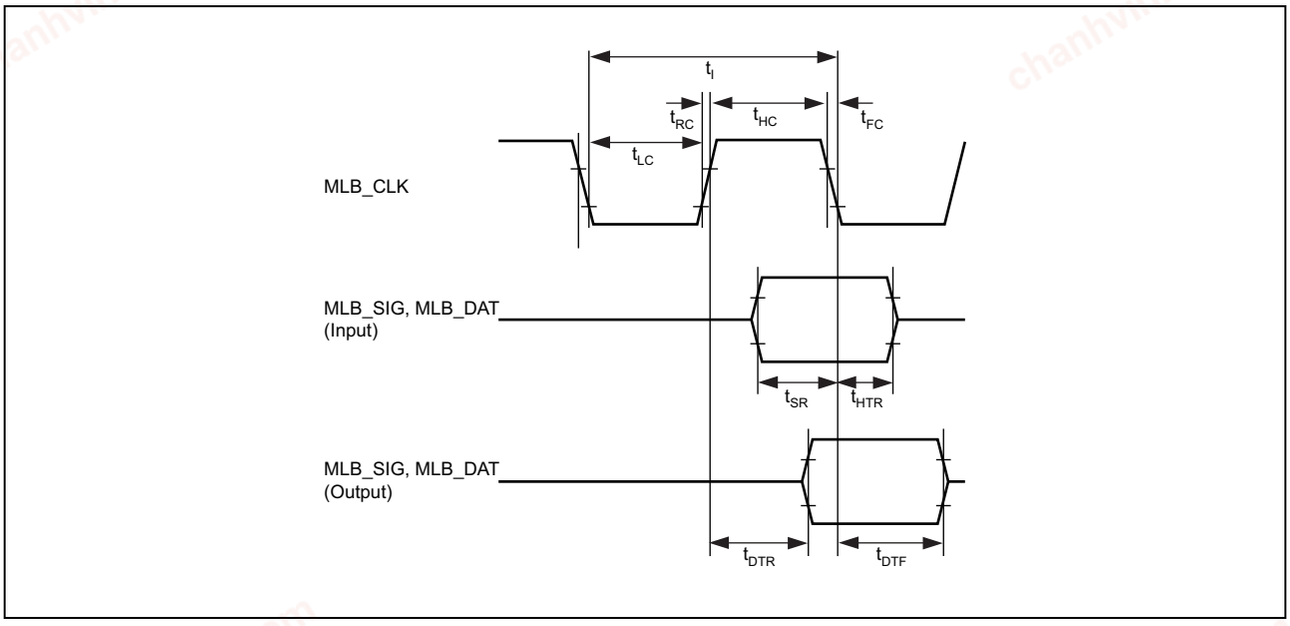


Figure 47.60 Interface Timing

chanhvinh.com

47.4.14 CAN Interface Timing

Table 47.18 CAN Interface Timing

Item	Symbol	Min.	Max.	Unit	Figure
Internal delay time	tnode	—	100	ns	Figure 47.61
Transmission rate		—	1	Mbps	

Internal delay time (tnode) = Internal transfer delay time (toutput) + Internal receive delay time (tinput)

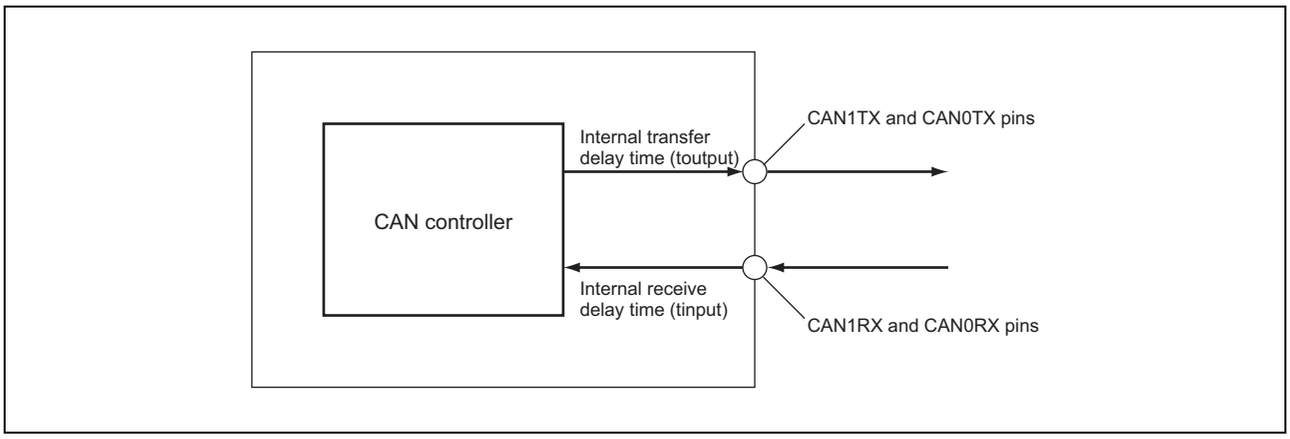


Figure 47.61 CAN Interface Condition

chanhvinh.com

47.4.15 Ethernet Controller and EthernetAVB Timing

The EthernetAVB is only incorporated in the RZ/A1LU.

Table 47.19 Ethernet Controller and EthernetAVB Timing

Item	Symbol	Min.	Max.	Unit	Figure
ET_TXCLK cycle time	t_{Tcyc}	40	—	ns	Figure 47.62, Figure 47.63, Figure 47.64, Figure 47.65, and Figure 47.66
ET_TXCLK high level width	t_{TCKWH}	0.35 × t_{Tcyc}	—	ns	
ET_TXCLK low level width	t_{TCKWL}	0.35 × t_{Tcyc}	—	ns	
ET_TXEN output delay time	t_{TEND}	0	25	ns	
ET_TXD[3:0] output delay time	t_{TDD}	0	25	ns	
ET_RXCLK cycle time	t_{Rcyc}	40	—	ns	
ET_RXCLK high level width	t_{RCKWH}	0.35 × t_{Rcyc}	—	ns	
ET_RXCLK low level width	t_{RCKWL}	0.35 × t_{Rcyc}	—	ns	
ET_RXDV setup time	t_{RDVS}	10	—	ns	
ET_RXDV hold time	t_{RDVH}	10	—	ns	
ET_RXD[3:0] setup time	t_{RDDS}	10	—	ns	
ET_RXD[3:0] hold time	t_{RDDH}	10	—	ns	
ET_RXER setup time	t_{RERS}	10	—	ns	
ET_RXER hold time	t_{RERH}	10	—	ns	
AVB_GPTP_EXTERN cycle time (RZ/A1LU only)	t_{Gcyc}	40	—	ns	
AVB_GPTP_EXTERN high level width (RZ/A1LU only)	t_{GCKWH}	0.35 × t_{Gcyc}	—	ns	
AVB_GPTP_EXTERN low level width (RZ/A1LU only)	t_{GCKWL}	0.35 × t_{Gcyc}	—	ns	
AVB_CAPTURE high level width (RZ/A1LU only)	t_{CAPWH}	2 × t_{Ccyc}*	—	ns	

Note: * This is the cycle time of the clock selected by the CSEL bit in the AVB-DMAC mode register (CCC).

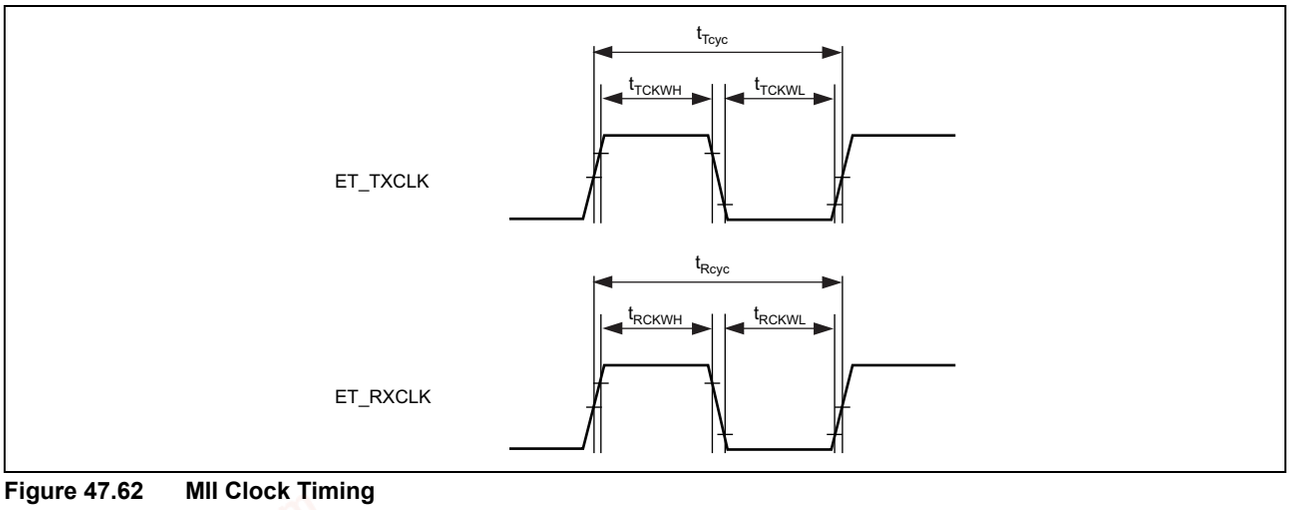


Figure 47.62 MII Clock Timing

chanhvinh.com

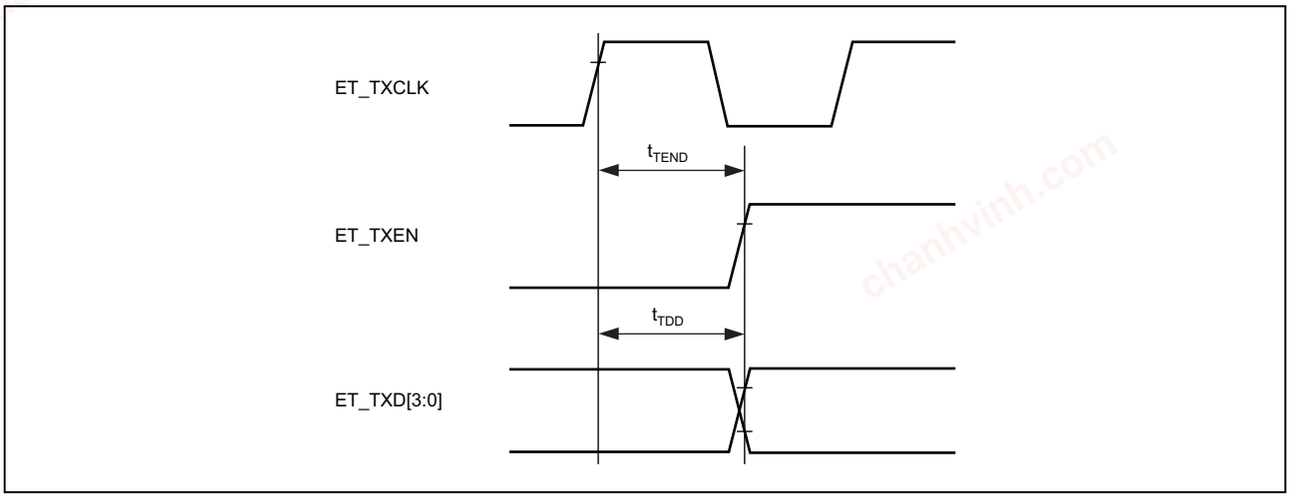


Figure 47.63 MII Transmit Data Timing

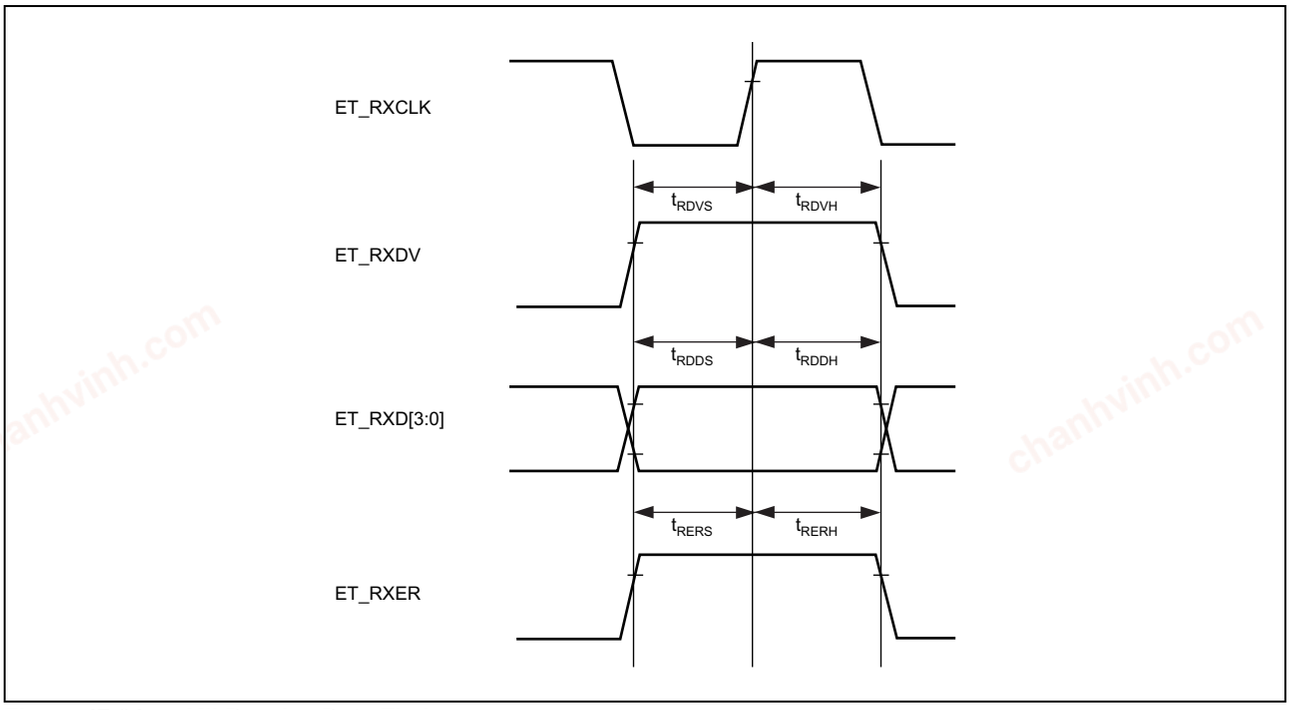


Figure 47.64 MII Receive Data Timing

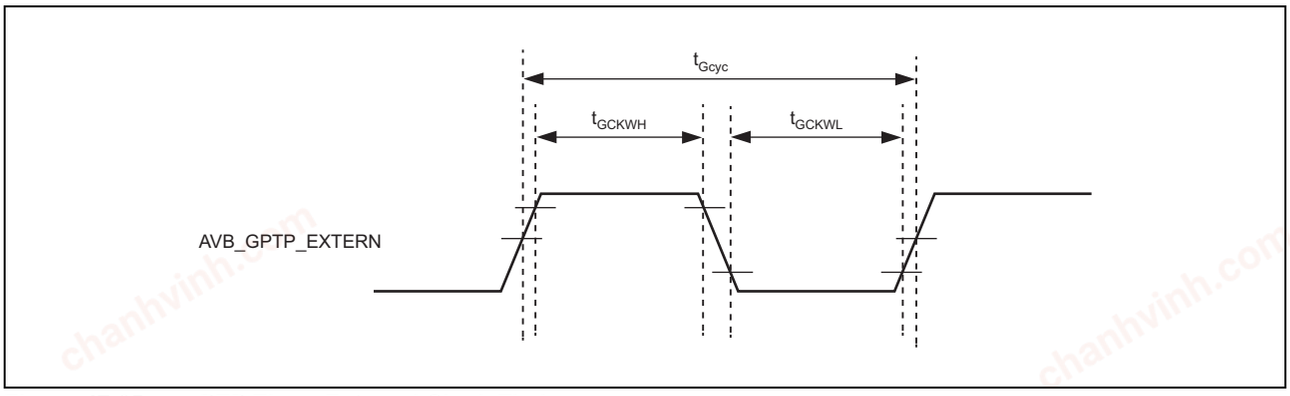


Figure 47.65 gPTP Timer External Clock Timing

chanhvinh.com

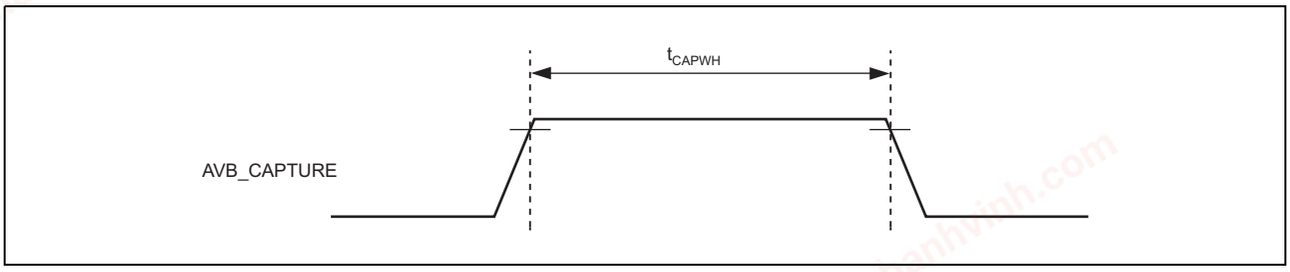


Figure 47.66 Timer Capture Signal Timing

chanhvinh.com

47.4.16 A/D Converter Timing

Table 47.20 A/D Converter Timing

Module	Item	Symbol	Min.	Max.	Unit	Figure
A/D converter	Trigger input setup time	t_{TRGS}	17	—	ns	Figure 47.67

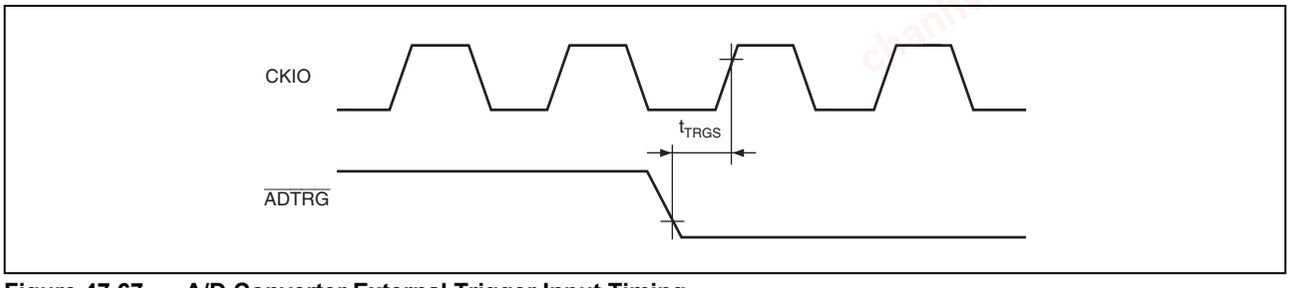


Figure 47.67 A/D Converter External Trigger Input Timing

47.4.17 USB 2.0 Host/Function Module Timing

Table 47.21 USB Transceiver Timing (Low-Speed)

Item	Symbol	Min.	Max.	Unit	Figure
Rise time	t_{LR}	75	300	ns	Figure 47.68
Fall time	t_{LF}	75	300	ns	
Rise/fall time lag	t_{LR}/t_{LF}	80	125	%	

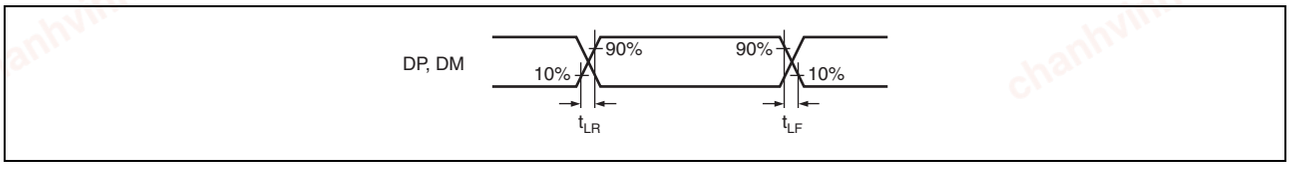


Figure 47.68 DP1, DP0, DM1, and DM0 Output Timing (Low-Speed)

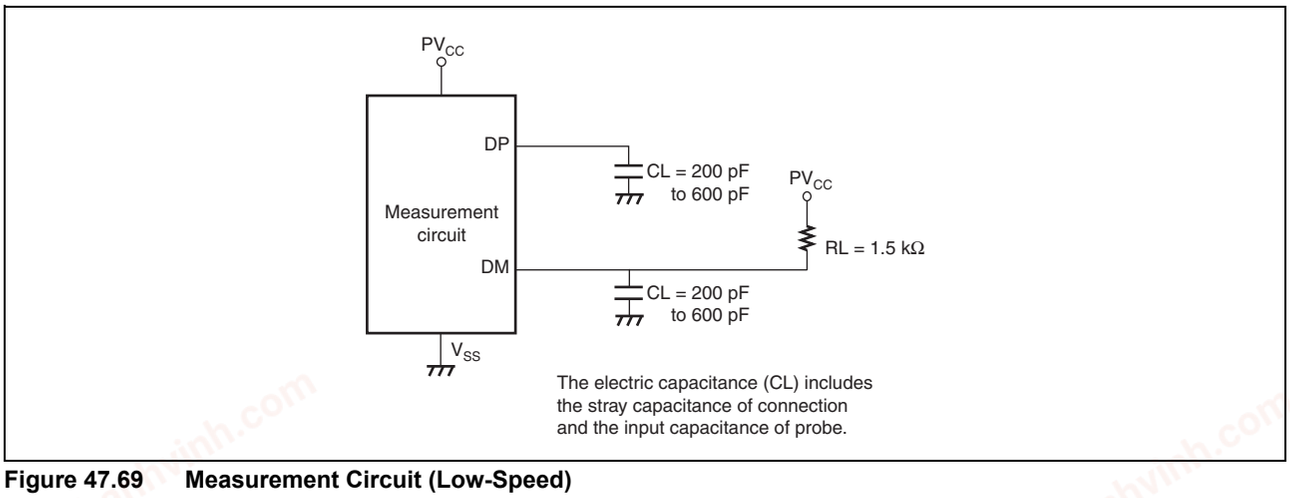


Figure 47.69 Measurement Circuit (Low-Speed)

chanhvinh.com

Table 47.22 USB Transceiver Timing (Full-Speed)

Item	Symbol	Min.	Max.	Unit	Figure
Rise time	t_{FR}	4	20	ns	Figure 47.70
Fall time	t_{FF}	4	20	ns	
Rise/fall time lag	t_{FR}/t_{FF}	90	111.11	%	

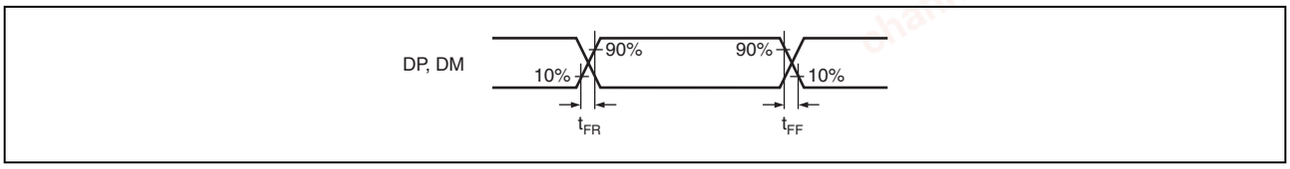
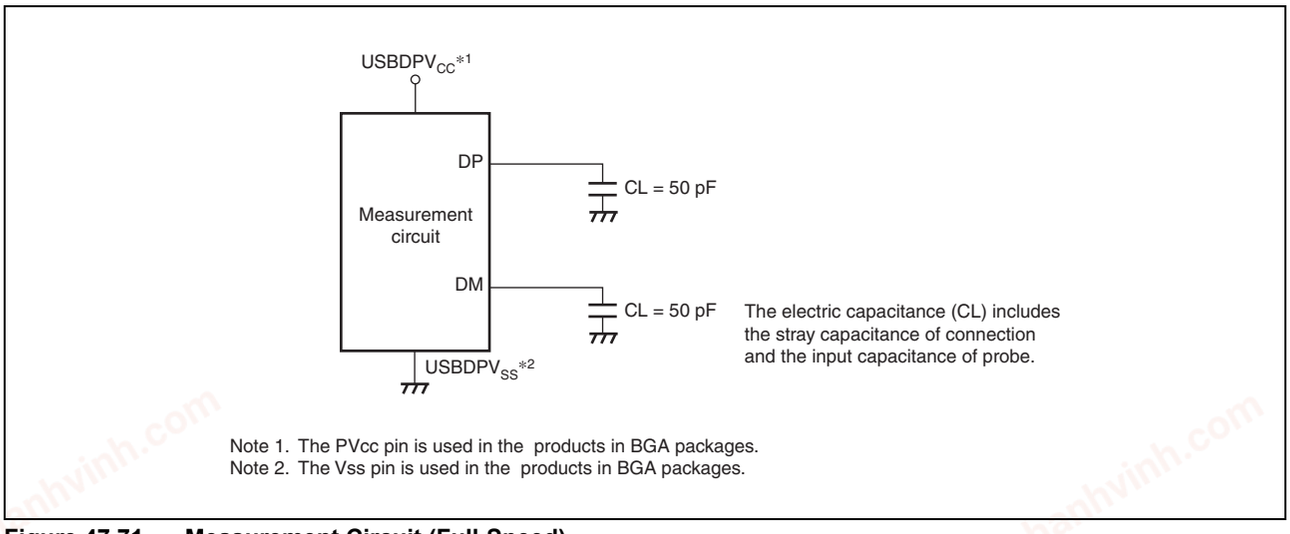


Figure 47.70 DP1, DP0, DM1, and DM0 Output Timing (Full-Speed)



Note 1. The PVcc pin is used in the products in BGA packages.
Note 2. The Vss pin is used in the products in BGA packages.

Figure 47.71 Measurement Circuit (Full-Speed)

chanhvinh.com

Table 47.23 USB Transceiver Timing (High-Speed)

Item	Symbol	Min.	Max.	Unit	Figure
Rise time	t_{HSR}	500	—	ps	Figure 47.72
Fall time	t_{HSF}	500	—	ps	
Output driver resistance	Z_{HSDRV}	40.5	49.5	Ω	

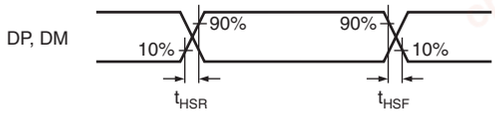
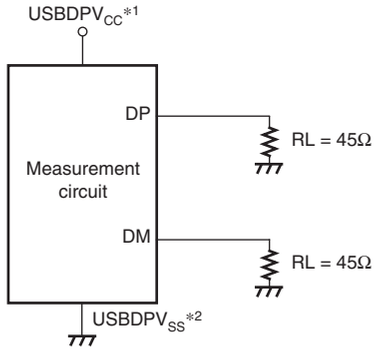


Figure 47.72 DP1, DP0, DM1, and DM0 Output Timing (High-Speed)



Note 1. The PVcc pin is used in the products in BGA packages.
Note 2. The Vss pin is used in the products in BGA packages.

Figure 47.73 Measurement Circuit (High-Speed)

chanhvinh.com

47.4.18 Video Display Controller 5 Timing

Table 47.24 Video Display Controller 5 Timing

Item	Symbol	Min.	Max.	Unit	Figure
DV0_CLK input clock frequency	t_{Dcyc}	—	87.00	MHz	Figure 47.74
DV0_CLK input clock low pulse width	t_{WIL}	0.4	—	t_{Dcyc}	
DV0_CLK input clock high pulse width	t_{WIH}	0.4	—		
CD0_EXTCLK input clock frequency	t_{Ecyc}	—	87.00	MHz	
LCD0_EXTCLK input clock low pulse width	t_{WIL}	0.4	—	t_{Ecyc}	
LCD0_EXTCLK input clock high pulse width	t_{WIH}	0.4	—		
LCD0_CLK output clock frequency	t_{Lcyc}	—	87.00	MHz	Figure 47.75
LCD0_CLK clock output low pulse width*1	t_{LOL}	t_{WIL} - 0.95	t_{WIL} + 0.95	ns	
LCD0_CLK clock output high pulse width*1	t_{LOH}	t_{WIH} - 0.95	t_{WIH} + 0.95	ns	
LCD0_CLK clock output low pulse width*2	t_{LOL}	$t_{Lcyc}/2$ - 1.06	$t_{Lcyc}/2$ + 1.06	ns	
LCD0_CLK clock output high pulse width*2	t_{LOH}	$t_{Lcyc}/2$ - 1.06	$t_{Lcyc}/2$ + 1.06	ns	
LCD0_CLK clock output rise time	t_{LOR}	—	3	ns	
LCD0_CLK clock output fall time	t_{LOF}	—	3	ns	
Input data setup time	t_{VS}	2	—	ns	Figure 47.76
Input data hold time	t_{VH}	4	—	ns	
Output data delay time	t_{DD}	–3	3	ns	Figure 47.77

Note 1. This is the case when the video image clock or an external clock is selected as the clock for frequency division and the division ratio is set to 1/1.

Note 2. This is for cases other than when the video image clock or an external clock is selected as the clock for frequency division and the division ratio is set to 1/1.

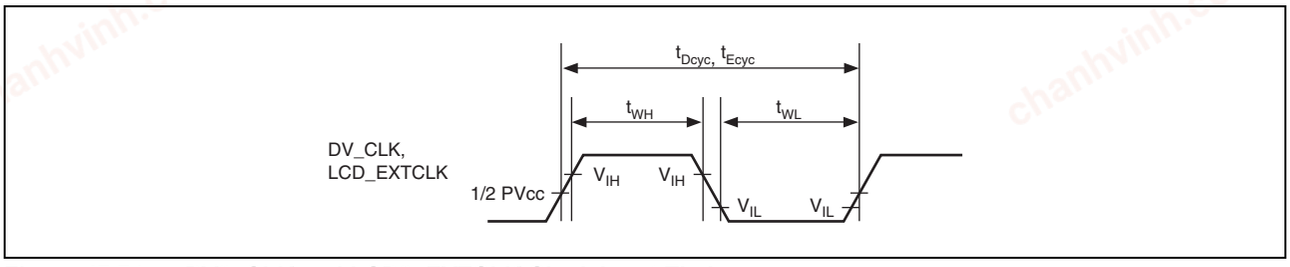


Figure 47.74 DV0_CLK and LCD0_EXTCLK Clock Input Timing

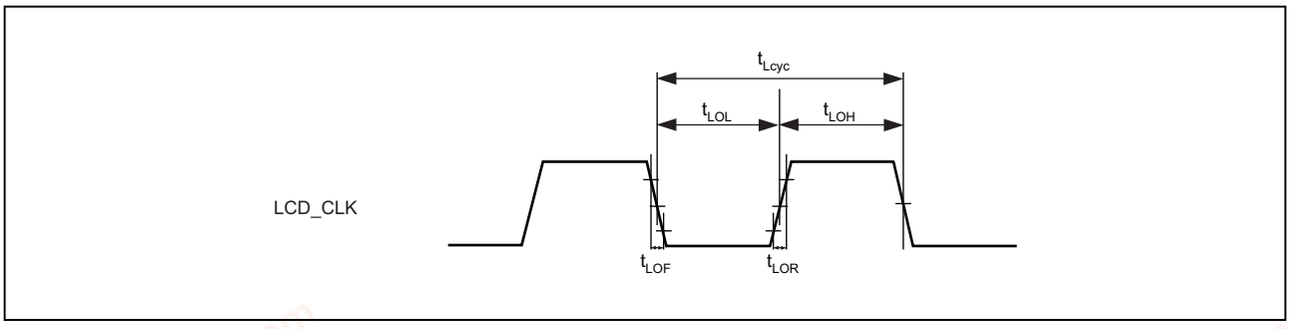


Figure 47.75 LCD0_CLK Clock Output Timing

chanhvinh.com

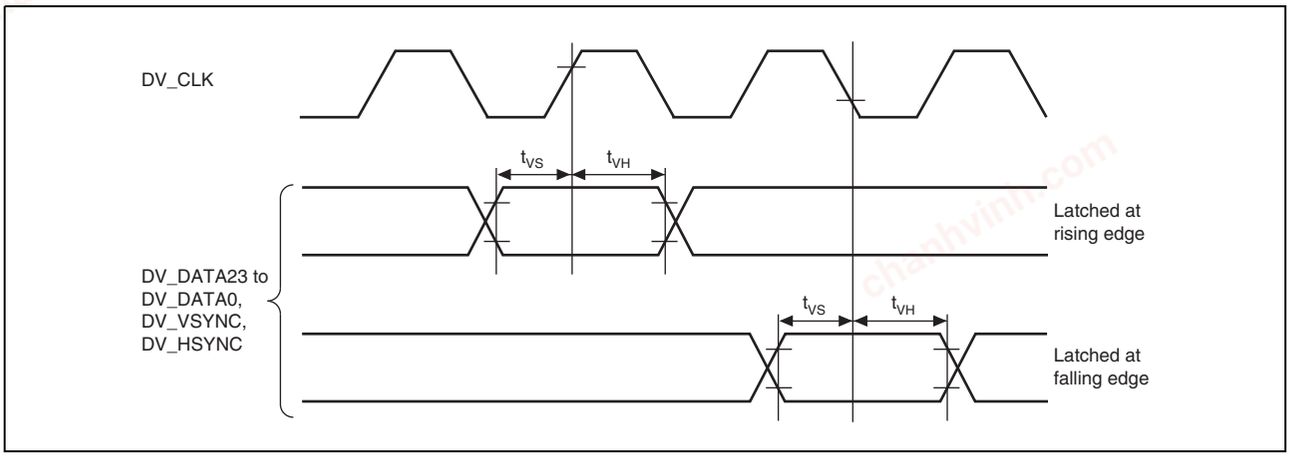


Figure 47.76 Video Input Timing

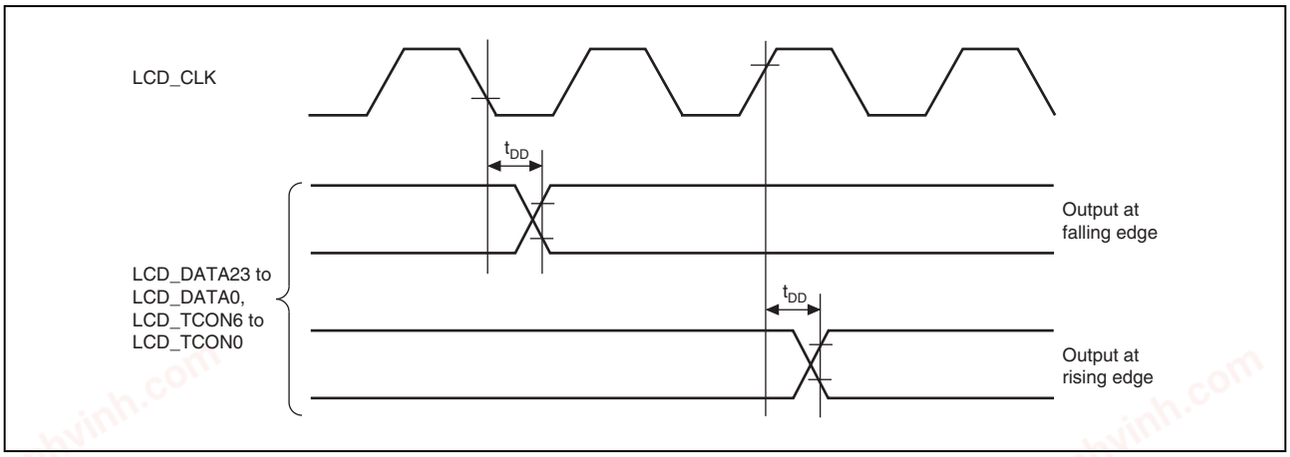


Figure 47.77 Display Output Timing

chanhvinh.com

47.4.19 Capture Engine Unit Module Signal Timing

Table 47.25 Capture Engine Unit Module Signal Timing

Item	Symbol	Min.	Max.	Unit	Figure
Vertical sync (VIO_VD) setup time (when data is captured at the rising edges of the camera clock)	t_{VVDS}	2	—	ns	Figure 47.78 (1) and Figure 47.78 (2)
Vertical sync (VIO_VD) setup time (when data is captured at the falling edges of the camera clock) (RZ/A1LU and RZ/A1LC only)	t_{VVDS}	2.5	—	ns	
Vertical sync (VIO_VD) hold time	t_{VVDH}	3.5	—	ns	
Horizontal sync (VIO_HD) setup time (when data is captured at the rising edges of the camera clock)	t_{VHDS}	2	—	ns	
Horizontal sync (VIO_HD) setup time (when data is captured at the falling edges of the camera clock) (RZ/A1LU and RZ/A1LC only)	t_{VHDS}	2.5	—	ns	
Horizontal sync (VIO_HD) hold time	t_{VHDH}	3.5	—	ns	
Capture image data (VIO_D) setup time (when data is captured at the rising edges of the camera clock)	t_{VDTS}	2	—	ns	
Capture image data (VIO_D) setup time (when data is captured at the falling edges of the camera clock) (RZ/A1LU and RZ/A1LC only)	t_{VDTS}	2.5	—	ns	
Capture image data (VIO_D) hold time	t_{VDTH}	3.5	—	ns	
Camera clock cycle	t_{VCYC}	—	87	MHz	
Camera clock high level	t_{VHW}	$0.4 \times t_{VCYC}$	—	ns	
Camera clock low level	t_{VLW}	$0.4 \times t_{VCYC}$	—	ns	
Field identification signal (VIO_FLD) setup time (when data is captured at the rising edges of the camera clock)	t_{VFDS}	2	—	ns	
Field identification signal (VIO_FLD) setup time (when data is captured at the falling edges of the camera clock) (RZ/A1LU and RZ/A1LC only)	t_{VFDS}	2.5	—	ns	
Field identification signal (VIO_FLD) hold time	t_{VFDH}	3.5	—	ns	

chanhvinh.com

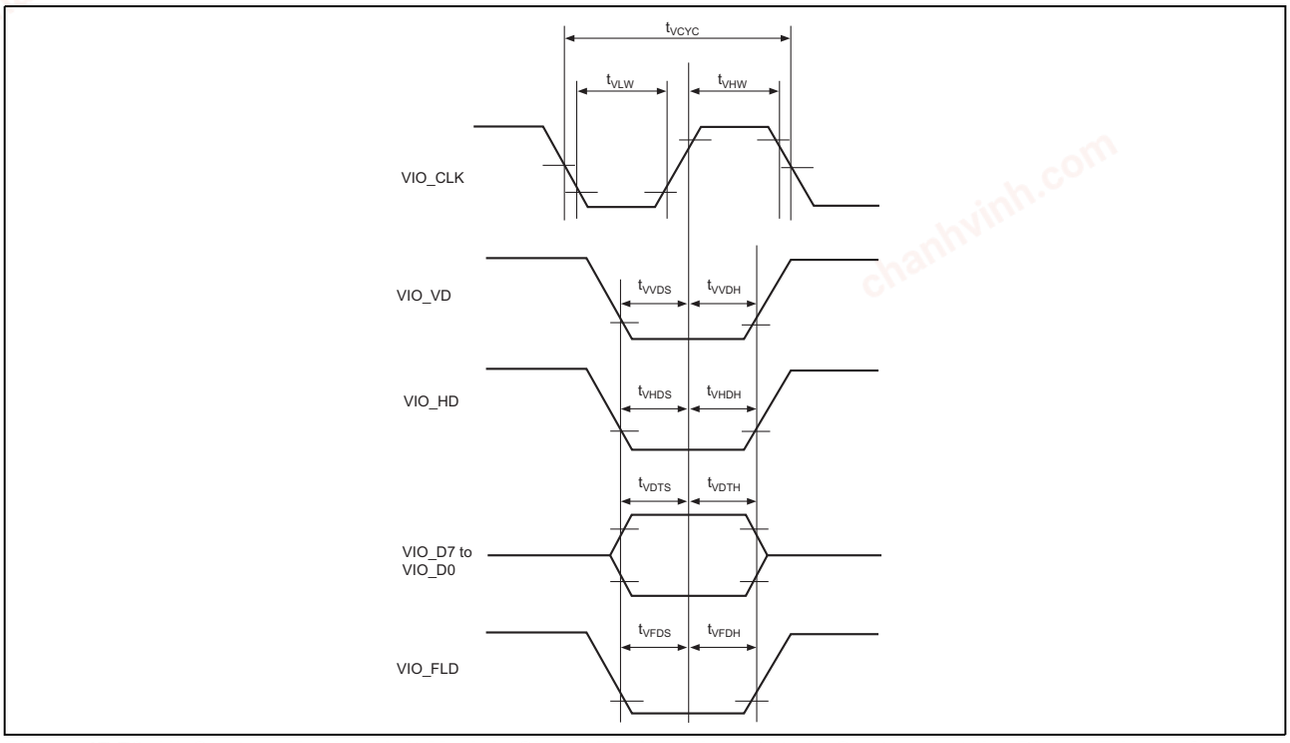


Figure 47.78 (1) Capture Engine Unit Module Signal Timing when Data is Captured at the Rising Edges of VIO_CLK

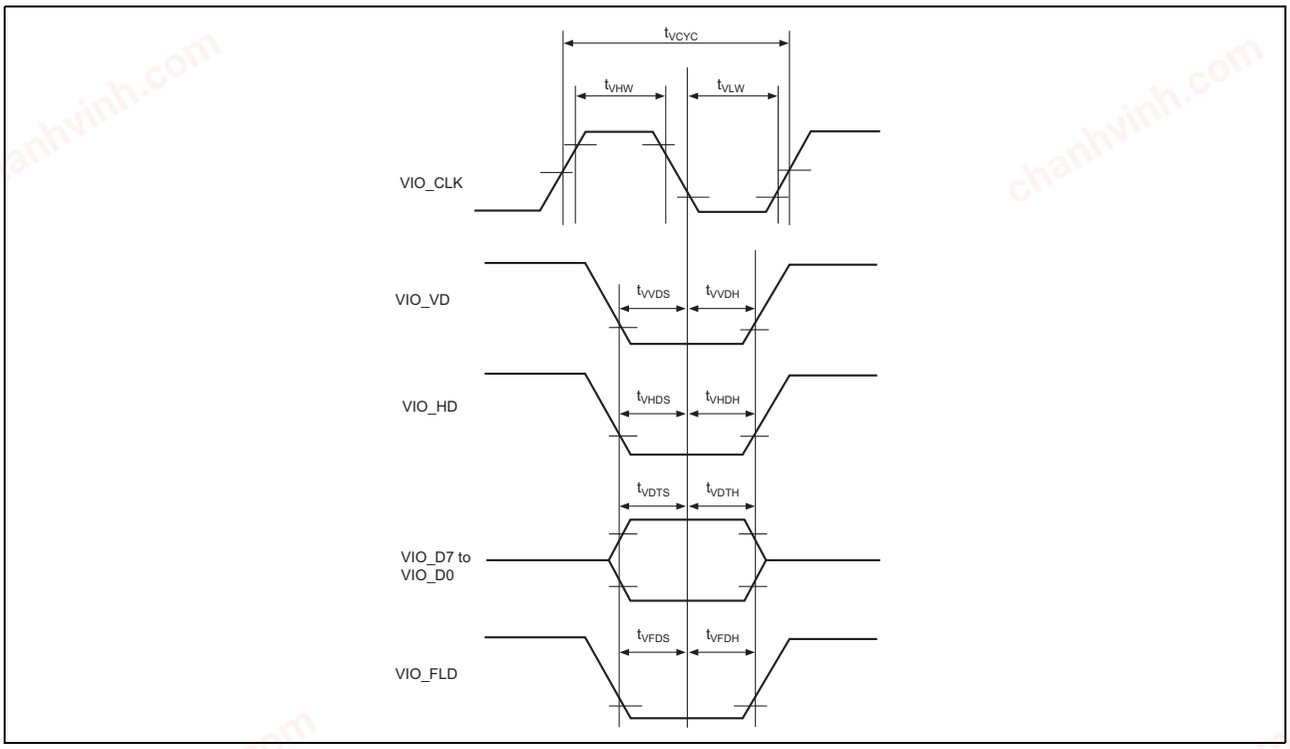


Figure 47.78 (2) Capture Engine Unit Module Signal Timing when Data is Captured at the Falling Edge of VIO_CLK (RZ/A1LU and RZ/A1LC Only)

chanhvinh.com

47.4.20 SD Host Interface Timing

Table 47.26 SD Host Interface Timing

Item	Symbol	Min.	Max.	Unit	Figure
SD_CLK clock cycle	t_{SDPP}	2 × t_{p1cyc}	—	ns	Figure 47.79
SD_CLK clock high level width	t_{SDWH}	0.4 × t_{SDPP}	—	ns	
SD_CLK clock low level width	t_{SDWL}	0.4 × t_{SDPP}	—	ns	
SD_CLK clock rise time	t_{SDLH}	—	3	ns	
SD_CLK clock fall time	t_{SDHL}	—	3	ns	
SD_CMD, SD_D3 to SD_D0 output data delay time (data transfer mode)	t_{SDODLY}	—	4	ns	
SD_CMD, SD_D3 to SD_D0 input data setup time	t_{SDISU}	5	—	ns	
SD_CMD, SD_D3 to SD_D0 input data hold time	t_{SDIH}	2	—	ns	

Note: • t_{p1cyc} indicates peripheral clock 1 (P1ϕ) cycle.

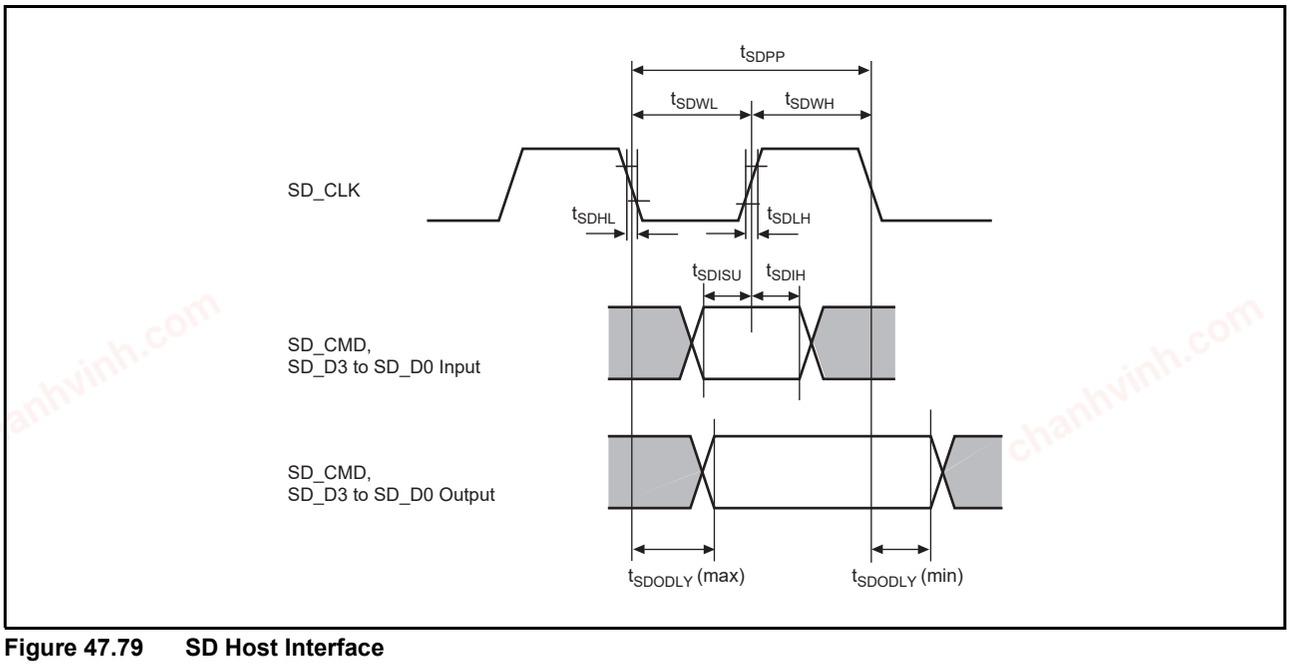


Figure 47.79 SD Host Interface

chanhvinh.com

47.4.21 MMC Host Interface Timing

Table 47.27 MMC Host Interface Timing

Item	Symbol	Min.	Max.	Unit	Figure
MMC_CLK clock cycle	t_{MMCPP}	$2 \times t_{p1cyc}$	—	ns	Figure 47.80
MMC_CLK clock high level width	t_{MMCWH}	6.5	—	ns	
MMC_CLK clock low level width	t_{MMCWL}	6.5	—	ns	
MMC_CLK clock rise time	t_{MMCLH}	—	3	ns	
MMC_CLK clock fall time	t_{MMCHL}	—	3	ns	
MMC_CMD, MMC_D7 to MMC_D0 output data delay time (data transfer mode)	$t_{MMCODLY}$	– 6.5	6.5	ns	
MMC_CMD, MMC_D7 to MMC_D0 input data setup time	t_{MMCISU}	4.5	—	ns	
MMC_CMD, MMC_D7 to MMC_D0 input data hold time	t_{MMCIH}	2	—	ns	

Note: • t_{p1cyc} indicates peripheral clock 1 (P1ϕ) cycle.

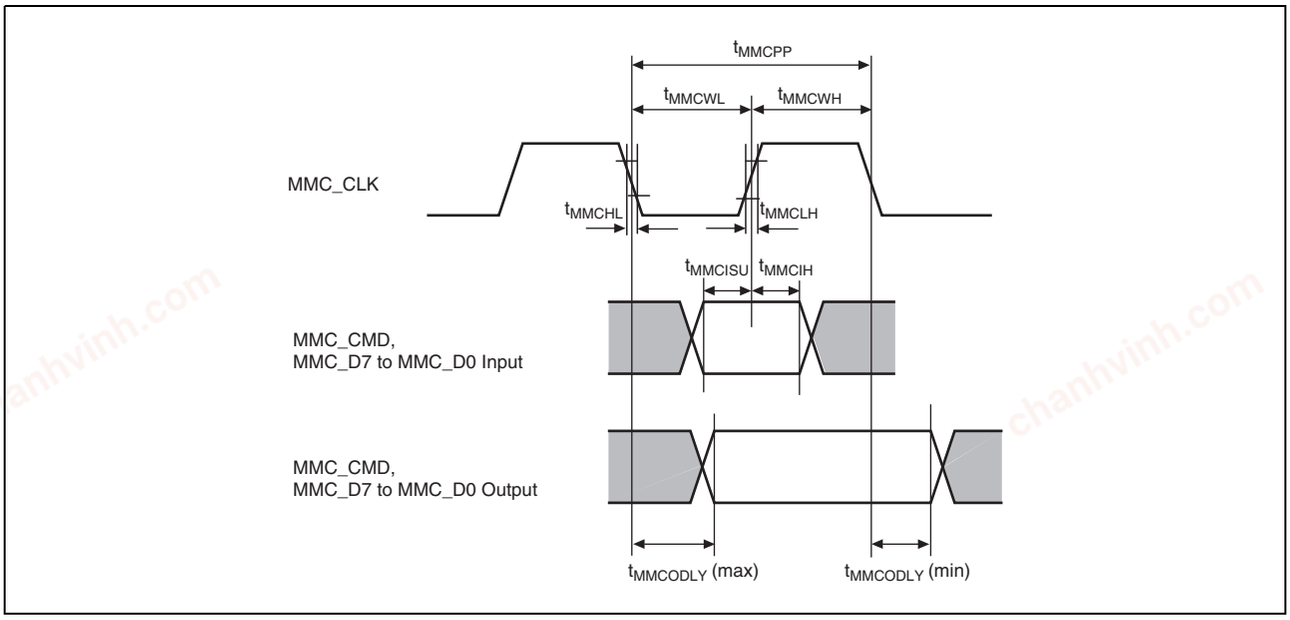


Figure 47.80 MMC Interface

chanhvinh.com

47.4.22 General Purpose I/O Ports Timing

Table 47.28 General Purpose I/O Ports Timing

Item	Symbol	Min.	Max.	Unit	Figure
Output data delay time	t_{PORTD}	—	100	ns	Figure 47.81
Input data setup time	t_{PORTS}	100	—		
Input data hold time	t_{PORTH}	100	—		

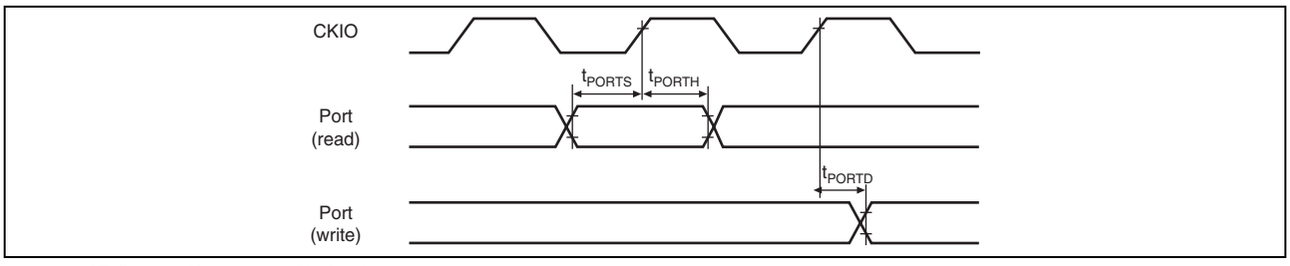


Figure 47.81 General I/O Ports Timing

47.4.23 Debugger Interface Timing

Table 47.29 Debugger Interface Timing

Item	Symbol	Min.	Max.	Unit	Figure
TCK cycle time	t_{TCKcyc}	50*1	—	ns	Figure 47.82
TCK high pulse width	t_{TCKH}	0.4	0.6	t_{TCKcyc}	
TCK low pulse width	t_{TCKL}	0.4	0.6	t_{TCKcyc}	
TDI setup time	t_{TDIS}	10	—	ns	Figure 47.83
TDI hold time	t_{TDIH}	10	—	ns	
TMS/SWDIO setup time	t_{TMSS}	10	—	ns	
TMS/SWDIO hold time	t_{TMSH}	10	—	ns	
SWDIO delay time	t_{SWDO}	—	16	ns	
TDO delay time	t_{TDOD}	—	16	ns	
Capture register setup time	t_{CAPTS}	10	—	ns	Figure 47.84
Capture register hold time	t_{CAPTH}	10	—	ns	
Update register delay time	$t_{UPDATED}$	—	20	ns	
Trace clock cycle	t_{TCYC}	30*2	—	ns	Figure 47.85 Output load: 15 pF
Trace clock high level	t_{THC}	12	—	ns	
Trace clock low level	t_{TLC}	12	—	ns	
Trace data delay time	t_{TDT}	3	0.3 × t_{TCYC} + 3	ns	

Note 1. Should be greater than the peripheral clock 0 (P0ϕ) cycle time.
Note 2. Generated by dividing the frequency of the peripheral clock (P1ϕ) by 2.

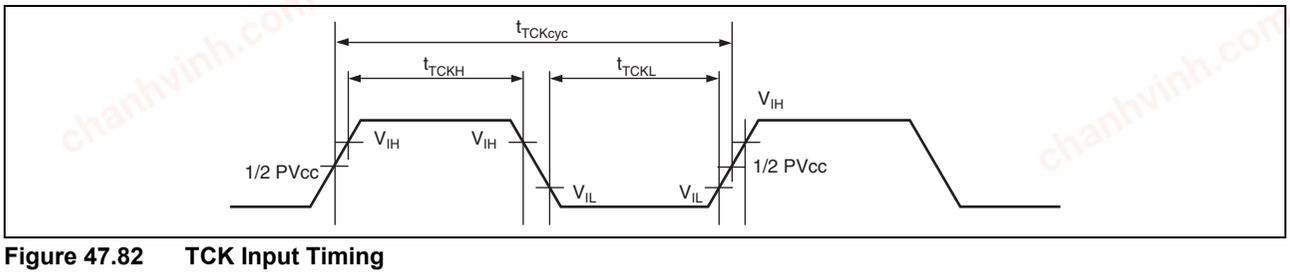


Figure 47.82 TCK Input Timing

chanhvinh.com

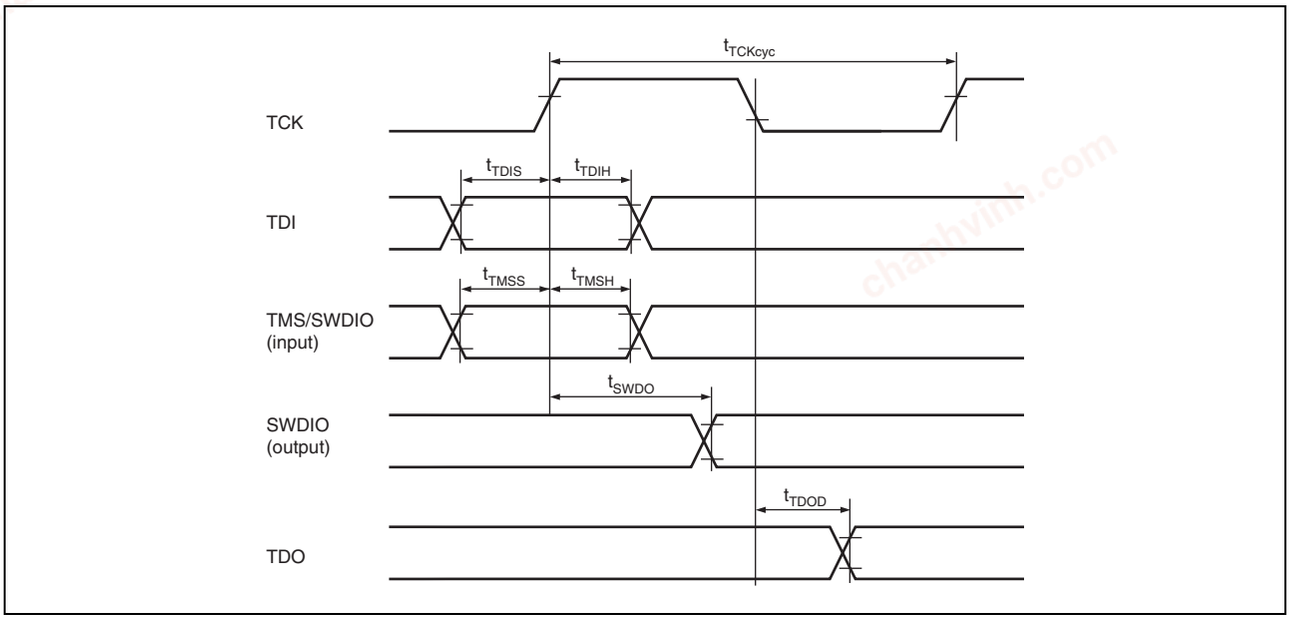


Figure 47.83 Data Transfer Timing

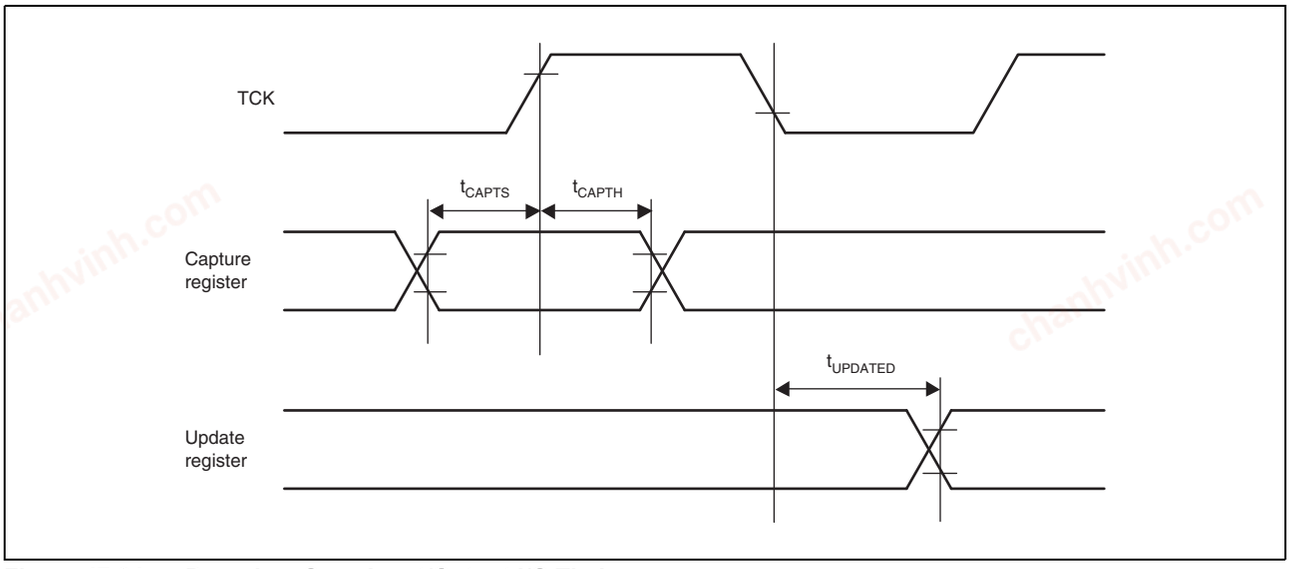


Figure 47.84 Boundary Scan Input/Output I/O Timing

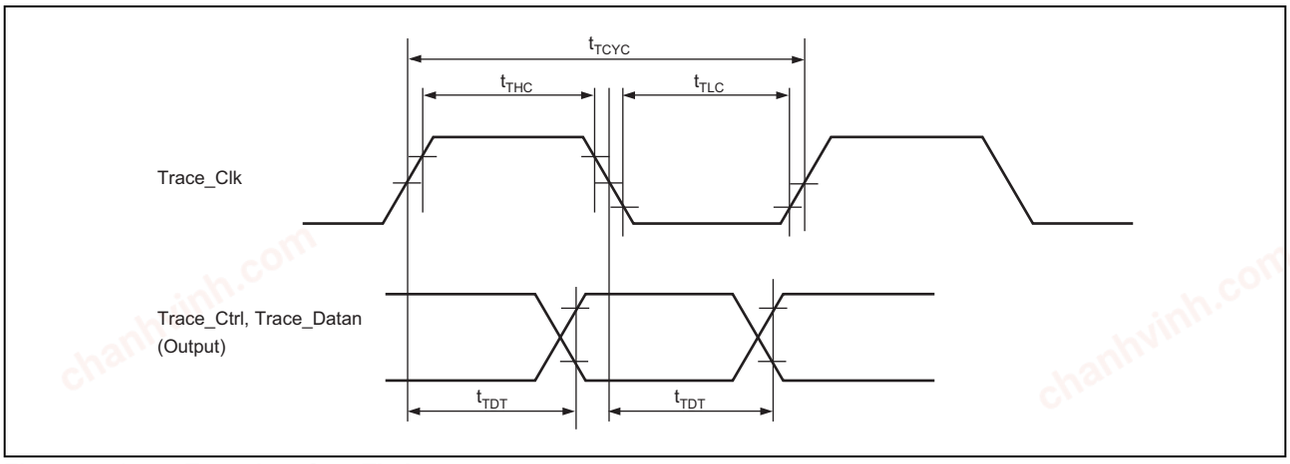


Figure 47.85 Trace Interface Timing

chanhvinh.com

47.4.24 AC Characteristics Measurement Conditions

- I/O signal reference level: $PV_{CC}/2$, the minimum values of V_{IH}, V_{T+}, and V_{OH}, and the maximum values of V_{IL}, V_{T-}, and V_{OL} (refer to the individual timing chart)
- Input pulse level: PV_{CC}
- Input rise and fall times: 1 ns

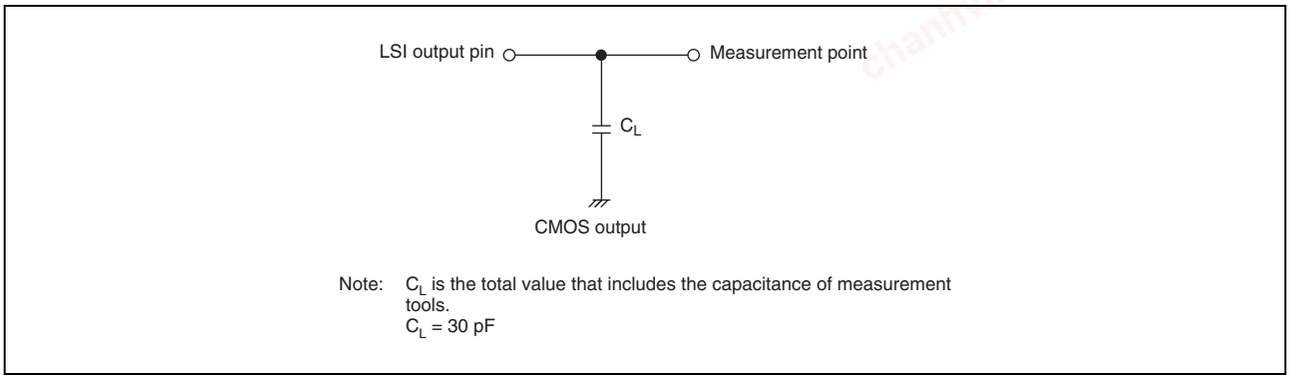


Figure 47.86 Output Load Circuit

chanhvinh.com

47.5 A/D Converter Characteristics

Conditions: V_{CC} = $USBDV_{CC}$ = $USBUV_{CC}$ = 1.10 to 1.26 V, PV_{CC} = $USBDPV_{CC}$ = 3.0 to 3.6 V,
$PLLV_{CC}$ = 1.10 to 1.26 V, AV_{CC} = 3.0 to 3.6 V, $USBAPV_{CC}$ = 3.0 to 3.6 V, $USBAV_{CC}$ = 1.10 to 1.26 V,
V_{SS} = AV_{SS} = $USBDV_{SS}$ = $USBAV_{SS}$ = $USBDPV_{SS}$ = $USBAPV_{SS}$ = $USBUV_{SS}$ = 0 V,
Ta = –40 to 85 °C

Note: Products in BGA packages do not have $USBDV_{CC}$, $USBUV_{CC}$, $USBDPV_{CC}$, $USBDV_{SS}$, $USBAV_{SS}$, $USBDPV_{SS}$, $USBAPV_{SS}$, and $USBUV_{SS}$ pins.

Table 47.30 A/D Converter Characteristics

Item		Min.	Typ.	Max.	Unit
Resolution		12	12	12	bits
Conversion time	12-bit	5	—	—	µs
	10-bit				
Analog input capacitance		—	—	20	pF
Permissible signal-source impedance		—	—	3	kΩ
DNL	12-bit	—	—	±1.0	LSB
	10-bit	—	—	±1.0	LSB
INL	12-bit	—	—	±4.0	LSB
	10-bit	—	—	±4.0	LSB
Offset error	12-bit	—	—	±8.0	LSB
	10-bit	—	—	±2.0	LSB
Full-scale error	12-bit	—	—	±8.0	LSB
	10-bit	—	—	±2.0	LSB
Absolute accuracy	12-bit	—	—	±11.0	LSB
	10-bit	—	—	±5.0	LSB

chanhvinh.com

48. States and Handling of Pins

This section describes pin states in each operating mode and how to handle pins.

48.1 Pin States

Table 48.1 shows the pin states in each operating mode.

As for the input/output functions, input buffers are listed on the upper column and output buffers on the lower column.

Table 48.1 Pin States

Pin Function				Pin State					
						Pin State Retained*2			Power-Down State
						EBUSKEEPE*3(Other than States at Right)			
Type	Pin Name			Normal State (Other than States at Right)	Power-On Reset*1	0	1	Power-On Reset*4	Deep Standby Mode	Software Standby Mode
Clock	EXTAL*6			I	I	I			I/Z*5	I
	XTAL*6			O	O	O			O/Z*5	O/Z*5
	CKIO	Boot mode	0, 1	O/Z*7	O	O	O/Z*7		O/Z*7	O/Z*7
			Other than above	O/Z*7	O	O/Z*7			O/Z*7	O/Z*7
	AUDIO_CLK			I	-	-			I/Z*12	I
	AUDIO_X1*6			I	I	I			Z	Z
	AUDIO_X2*6			O	O	O			L	L
	AUDIO_XOUT, AUDIO_XOUT2, AUDIO_XOUT3			O	-	O/Z*9*15			O/Z*9*15	L/Z*9
System control	$\overline{RES}$			I	I	I			I	I
Operating mode control	MD_BOOT1, MD_BOOT0, MD_CLK, MD_CLKS			-	I	-			-	-
	BSCANP			I	I	I			I	I
Interrupts	NMI			I	I	I			I	I
	IRQ0 (P6_7), IRQ1 (P2_9), IRQ2 (P2_7), IRQ3 (P5_8, P7_3), IRQ4 (P5_9, P7_2), IRQ5 (P2_2, P5_10, P7_9), IRQ6 (P7_6), IRQ7 (P2_0)			I	-	-			I/Z*12	I
	IRQ0 (other than P6_7), IRQ1 (other than P2_9), IRQ2 (other than P2_7), IRQ3 (other than P5_8 and P7_3), IRQ4 (other than P5_9 and P7_2), IRQ5 (other than P2_2, P5_10, and P7_9), IRQ6 (other than P7_6), IRQ7 (other than P2_0)			I	-	-			Z	I
Bus state controller	A25 to A21, A0			O	-	O/Z*10			O/Z*10	O/Z*10
	A20 to A1	Boot mode	0	O	Z	O	O/Z*10		O/Z*10	O/Z*10
			Other than above	O	-	O/Z*10			O/Z*10	O/Z*10
	D0 to D7, D11 to D15	Boot mode	0	I/Z	Z	I/Z		Z	Z	Z
				O/Z	Z	O/Z	Z		Z	Z
			Other than above	I/Z	-	-			Z	Z
				O/Z	-	Z			Z	Z
	D8 (P5_8), D9 (P5_9), D10 (P5_10)	Boot mode	0	I/Z	Z	I/Z		Z	I/Z*12	I
				O/Z	Z	O/Z	Z		Z	Z
			Other than above	I/Z	-	-			I/Z*12	I
				O/Z	-	Z			Z	Z
	D16 to D22, D24 to D31			I/Z	-	-			Z	Z
				O/Z	-	Z			Z	Z
	D23 (P6_7)			I/Z	-	-			I/Z*12	I
				O/Z	-	Z			Z	Z
	$\overline{CS0}$, $\overline{RD}$	Boot mode	0	O	Z	O	H/Z*10		H/Z*10	H/Z*10
			Other than above	O	-	H/Z*10			H/Z*10	H/Z*10
	$\overline{CS5}$ to $\overline{CS1}$, RD/$\overline{WR}$, $\overline{BS}$, $\overline{WE3}$/DQMUU/$\overline{AH}$, $\overline{WE2}$/DQMUL, $\overline{WE1}$/DQMLU/$\overline{WE}$, $\overline{WE0}$/DQMLL			O	-	H/Z*10			H/Z*10	H/Z*10
	$\overline{WAIT}$			I	-	-			Z	Z

chanhvinh.com

Table 48.1 Pin States

Pin Function				Pin State					
				Normal State (Other than States at Right)	Power-On Reset*1	Pin State Retained*2			Power-Down State
						EBUSKEEPE*3(Other than States at Right)		Power-On Reset*4	Deep Standby Mode	Software Standby Mode
Type	Pin Name					0	1			
Bus state controller	$\overline{RAS}$, $\overline{CAS}$, CKE			O	-	O/Z*11			O/Z*11	O/Z*11
Direct memory access controller	DREQ0			I	-	-			Z	Z
	DACK0, TEND0			O	-	O/Z*9			O/Z*9	O/Z*9
Multi-function timer pulse unit 2	TCLKA, TCLKB, TCLKC, TCLKD			I	-	-			Z	Z
	TIOC0C (P2_2), TIOC1A (P2_7, P7_9), TIOC3A (P6_7), TIOC4A (P5_8), TIOC4B (P5_9), TIOC4C (P5_10)			I	-	-			I/Z*12	I
				O/Z	-	O/Z*9			O/Z*9	O/Z*9
	TIOC0A, TIOC0B, TIOC0C (other than P2_2), TIOC0D, TIOC1A (other than P2_7 and P7_9), TIOC1B, TIOC2A, TIOC2B, TIOC3A (other than P6_7), TIOC3B, TIOC3C, TIOC3D, TIOC4A (other than P5_8), TIOC4D			I	-	-			Z	Z
				O/Z	-	O/Z*9			O/Z*9	O/Z*9
Watchdog timer	$\overline{WDTOVF}$			O	-	H			H	H
Realtime clock	RTC_X1*6			I/Z*13	I	I/Z*13			I/Z*13	I/Z*13
	RTC_X2*6			O/Z*13	O	O/Z*13			O/Z*13	O/Z*13
Serial communication interface with FIFO	TxD0 to TxD4			O/Z	-	O/Z*9			O/Z*9	O/Z*9
	RxD0 to RxD4			I	-	-			Z	Z
	SCK0 (P2_9), SCK3 (P7_9), $\overline{CTS2}$ (P7_6)			I	-	-			I/Z*12	I
				O/Z	-	O/Z*9			O/Z*9	O/Z*9
	SCK0 (other than P2_9), SCK1, SCK2, SCK3 (other than P7_9), SCK4, $\overline{RTS0}$ to $\overline{RTS2}$, $\overline{CTS0}$, $\overline{CTS1}$, $\overline{CTS2}$ (other than P7_6)			I	-	-			Z	Z
				O/Z	-	O/Z*9			O/Z*9	O/Z*9
Serial communications interface	SCI_TXD0, SCI_TXD1			O/Z	-	O/Z*9			O/Z*9	O/Z*9
	SCI_RXD0, SCI_RXD1			I	-	-			Z	Z
	$\overline{SCI_CTS0/RTS0}$ (P7_3)			I	-	-			I/Z*12	I
				O/Z	-	O/Z*9			O/Z*9	O/Z*9
	SCI_SCK0, SCI_SCK1, $\overline{SCI_CTS1/RTS1}$			I	-	-			Z	Z
				O/Z	-	O/Z*9			O/Z*9	O/Z*9
Renesas serial peripheral interface	RSPCK0 to RSPCK2, SSL00, SSL10, SSL20, MOSI0 to MOSI2, MISO0 to MISO2			I	-	-			Z	Z
				O/Z	-	O/Z*9			O/Z*9	O/Z*9
SPI multi I/O bus controller	SPBCLK_0, SPBSSL_0	Boot mode	1	O/Z	-	-	O/Z*9		O/Z*9	O/Z*9
			Other than above	O/Z	-	O/Z*9			O/Z*9	O/Z*9
	SPBIO00_0, SPBIO10_0			I	-	-			Z	Z
		Boot mode	1	O/Z	-	-	O/Z*9		O/Z*9	O/Z*9
			Other than above	O/Z	-	O/Z*9			O/Z*9	O/Z*9
	SPBIO20_0, SPBIO30_0, SPBIO01_0, SPBIO11_0, SPBIO21_0, SPBIO31_0			I	-	-			Z	Z
				O/Z	-	O/Z*9			O/Z*9	O/Z*9
I²C bus interface	RIIC0SCL to RIIC3SCL, RIIC0SDA to RIIC3SDA			I	-	-			Z	Z
				L/Z	-	Z			Z	Z
Serial sound interface	SSITxD0, SSITxD1, SSITxD3			O	-	O/Z*9			O/Z*9	O/Z*9
	SSIRxD3 (P6_7)			I	-	-			I/Z*12	I
	SSIRxD0, SSIRxD1, SSIRxD3 (other than P6_7)			I	-	-			Z	Z
	SSIDATA2			I	-	-			Z	Z
				O/Z	-	O/Z*9			O/Z*9	O/Z*9
	SSISCK3 (P2_7), SSIWS1 (P7_9), SSIWS3 (P2_9)			I	-	-			I/Z*12	I
				O/Z	-	O/Z*9			O/Z*9	O/Z*9
	SSISCK0 to SSISCK2, SSISCK3 (other than P2_7), SSIWS0, SSIWS1 (other than P7_9), SSIWS2, SSIWS3 (other than P2_9)			I	-	-			Z	Z
				O/Z	-	O/Z*9			O/Z*9	O/Z*9
Media local bus*16	MLB_CLK			O/Z	-	O/Z*9			O/Z*9	O/Z*9
	MLB_DAT, MLB_SIG			I	-	-			Z	Z
				O/Z	-	O/Z*9			O/Z*9	O/Z*9

chanhvinh.com

Table 48.1 Pin States

Pin Function		Pin State						
		Normal State (Other than States at Right)	Power-On Reset*1	Pin State Retained*2			Power-Down State	
				EBUSKEEPE*3(Other than States at Right)		Power-On Reset*4	Deep Standby Mode	Software Standby Mode
Type	Pin Name			0	1			
CAN interface	CAN0TX, CAN1TX	O	-	O/Z*9			O/Z*9	O/Z*9
	CAN0RX (P2_9, P5_8, P7_2), CAN1RX (P2_2)	I	-	-			I/Z*12	I
	CAN1RX (other than P2_2), CAN_CLK	I	-	-			Z	Z
IEBus™ controller*16	IETxD	O	-	O/Z*9			O/Z*9	O/Z*9
	IERxD (P5_10)	I	-	-			I/Z*12	I
	IERxD (other than P5_10)	I	-	-			Z	Z
Renesas SPDIF interface	SPDIF_OUT	O	-	O/Z*9			O/Z*9	O/Z*9
	SPDIF_IN (P2_0)	I	-	-			I/Z*12	I
	SPDIF_IN (other than P2_0)	I	-	-			Z	Z
LIN interface*16	RLIN30TX	O	-	O/Z*9			O/Z*9	O/Z*9
	RLIN30RX (P2_0, P6_7)	I	-	-			I/Z*12	I
Ethernet controller, EthernetAVB*17	ET_TXER, ET_TXEN, ET_TXD3 to ET_TXD0, ET_MDC	O	-	O/Z*9			O/Z*9	O/Z*9
	ET_TXCLK, ET_RXCLK, ET_RXDV, ET_RXER, ET_RXD3 to ET_RXD0, ET_CRS, ET_COL	I	-	-			Z	Z
	ET_MDIO	I	-	-			Z	Z
		O	-	O/Z*9			O/Z*9	O/Z*9
EthernetAVB*17	AVB_CAPTURE	I	-	-			Z	Z
	AVB_GPTP_EXTERN	I	-	-			Z	Z
A/D converter	AN7 to AN0	I	-	-			Z	Z
	$\overline{\text{ADTRG}}$	I	-	-			Z	Z
USB2.0 host/ function module	DP0, DP1, DM0, DM1	I/Z	Z	I/Z			Z	I/Z
		O/Z	Z	O/Z			Z	O/Z
	VBUS0, VBUS1	I	I	I			I	I
	REFRIN	I	I	I			I	I
	USB_X1*6	I	I	I			Z	Z
	USB_X2*6	O	O	O			L	L
Video display controller 5	LCD0_CLK	O	-	O/Z*9			O/Z*9	O/Z*9
	LCD0_DATA0 to LCD0_DATA23, LCD0_TCON0 to LCD0_TCON6	O	-	O/Z*9			O/Z*9	O/Z*9
	LCD0_EXTCLK	I	-	-			Z	Z
	DV0_DATA6 (P7_2), DV0_DATA7 (P7_3), DV0_VSYNC (P7_9)	I	-	-			I/Z*12	I
	DV0_CLK, DV0_DATA0 to DV0_DATA5, DV0_DATA6 (other than P7_2), DV0_DATA7 (other than P7_3), DV0_DATA8 to DV0_DATA23, DV0_HSYNC	I	-	-			Z	Z
Capture engine unit	VIO_CLK, VIO_VD, VIO_HD, VIO_FLD, VIO_D0 to VIO_D7	I	-	-			Z	Z

chanhvinh.com

Table 48.1 Pin States

Pin Function				Pin State					
						Pin State Retained*2			Power-Down State
						EBUSKEEPE*3(Other than States at Right)			
Type	Pin Name			Normal State (Other than States at Right)	Power-On Reset*1	0	1	Power-On Reset*4	Deep Standby Mode	Software Standby Mode
SD host interface	SD_CLK_0 (P3_3)	Boot mode	2	O	-	-	O/Z*9		O/Z*9	O/Z*9
			Other than above	O	-	O/Z*9			O/Z*9	O/Z*9
	SD_CLK_0 (other than P3_3)			O	-	O/Z*9			O/Z*9	O/Z*9
	SD_CMD_0 (P3_2), SD_D0_0 (P3_4), SD_D1_0 (P3_5), SD_D2_0 (P3_0), SD_D3_0 (P3_1)			I	-	-			Z	Z
		Boot mode	2	O	-	-	O/Z*9		O/Z*9	O/Z*9
			Other than above	O/Z	-	O/Z*9			O/Z*9	O/Z*9
	SD_CMD_0 (other than P3_2), SD_D0_0 (other than P3_4), SD_D1_0 (other than P3_5), SD_D2_0 (other than P3_0), SD_D3_0 (other than P3_1)			I	-	-			Z	Z
				O/Z	-	O/Z*9			O/Z*9	O/Z*9
	SD_CLK_1			O/Z	-	O/Z*9			O/Z*9	O/Z*9
	SD_D0_1 (P7_3), SD_D1_1 (P7_2), SD_D3_1 (P7_6)			I	-	-			I/Z*12	I
				O/Z	-	O/Z*9			O/Z*9	O/Z*9
	SD_CMD_1, SD_D2_1			I	-	-			Z	Z
				O/Z	-	O/Z*9			O/Z*9	O/Z*9
	SD_CD_0, SD_CD_1, SD_WP_0, SD_WP_1			I	-	-			Z	Z
MMC host interface	MMC_CLK	Boot mode	3	O	-	-	O/Z*9		O/Z*9	O/Z*9
			Other than above	O	-	O/Z*9			O/Z*9	O/Z*9
	MMC_D0 (P7_3), MMC_D1 (P7_2), MMC_D3 (P7_6)			I	-	-			I/Z*12	I
		Boot mode	3	O/Z	-	-	O/Z*9		O/Z*9	O/Z*9
			Other than above	O/Z	-	O/Z*9			O/Z*9	O/Z*9
	MMC_CMD, MMC_D2			I	-	-			Z	Z
		Boot mode	3	O/Z	-	-	O/Z*9		O/Z*9	O/Z*9
			Other than above	O/Z	-	O/Z*9			O/Z*9	O/Z*9
	MMC_D4 to MMC_D7			I	-	-			Z	Z
				O/Z	-	O/Z*9			O/Z*9	O/Z*9
	MMC_CD			I	-	-			Z	Z
Ports	JP0_0, JP0_1, P0_0 to P0_3			I/Z	-	-			Z	Z
	P1_8 to P1_15			I/Z	Z	Z			Z	Z
	P1_0 to P1_7			I/Z*8	Z	Z			Z	Z
				L/Z	Z	Z			Z	Z
	P2_0, P2_2, P2_7, P2_9, P5_8, P5_9, P5_10, P6_7, P7_2, P7_3, P7_6, P7_9			I	Z	Z			I/Z*12	I
				O/Z	Z	O/Z*9			O/Z*9	O/Z*9
	Other than above			I/Z*8	Z	Z			Z	Z
				O/Z	Z	O/Z*9			O/Z*9	O/Z*9
Debugger interface	TDI			I	I	I			Z	I
	TDO			O/Z*14	O/Z*14	O/Z*14			Z	O/Z*14
	TMS/SWDIO			I	I	I			Z	I
				O/Z	O/Z	O/Z			Z	O/Z
	TCK/SWDCLK			I	I	I			Z	I
	$\overline{\text{TRST}}$			I	I	I			Z	I
	TRACEDATA3 to TRACEDATA0, TRACECLK, TRACECTL			O	-	O/Z*9			O/Z*9	O/Z*9

[Legend]
I: Input
O: Output
H: High-level output
L: Low-level output
Z: High-impedance
-: Condition under which the pin function is not selectable

chanhvinh.com

Note 1. Indicates the power-on reset by low-level input to the $\overline{RES}$ pin. The pin states after a power-on reset by the watchdog timer overflow is the same as the initial pin states at normal operation (see section 41, Ports).
Note 2. After the chip has been released from deep standby mode by the input on pins for canceling the standby mode such as NMI or by the realtime clock alarm interrupt, the pins retain the state until the IOKEEP bit in the deep standby cancel source flag register (DSFR) is cleared (see section 42, Power-Down Modes).
Note 3. The EBUSKEEPE bit in deep standby control register (DSTCR) (see section 42, Power-Down Modes).
Note 4. This LSI enters the power-on reset state for a certain period after recovery from deep standby control mode (see section 42, Power-Down Modes).
Note 5. Depends on the setting of the RCKSEL bit in the realtime clock control register 5 (RCR5) (see section 13, Realtime Clock).
Note 6. When pins for the connection with a crystal resonator are not used, the input pins (EXTAL, RTC_X1, AUDIO_X1, and USB_X1) must be fixed (pull-up/down resistor, power supply, or ground.) and the output pins (XTAL, RTC_X2, AUDIO_X2, and USB_X2) must be open.
Note 7. Depends on the setting of the CKOEN bit in the frequency control register (FRQCR) of the clock pulse generator (see section 6, Clock Pulse Generator).
Note 8. Depends on the setting of the PIPCnm bit in the port IP control register (PIPCn) of the general I/O ports.
Note 9. Depends on the setting of the HIZ bit in the standby control register 2 (STBCR2) (see section 42, Power-Down Modes).
Note 10. Depends on the setting of the HIZMEM bit in the common control register (CMNCR) of the bus state controller (see section 8, Bus State Controller).
Note 11. Depends on the setting of the HIZCNT bit in the common control register (CMNCR) of the bus state controller (see section 8, Bus State Controller).
Note 12. Depends on the setting of the corresponding bit in the deep standby cancel source select register (DSSSR) and the RCKSEL bit in the realtime clock control register 5 (RCR5) (see section 42, Power-Down Modes).
Note 13. Depends on the setting of the RTCEN bit in the realtime clock control register 2 (RCR2) and the RCKSEL bit in the realtime clock control register 5 (RCR5) (see section 13, Realtime Clock).
Note 14. O in serial wire debug (SWD) mode. In modes other than serial wire debug (SWD), Z when the TAP controller is neither the Shift-DR nor Shift-IR state.
Note 15. When this is an output, the output is fixed to either the High or Low level. There is no oscillation.
Note 16. RZ/A1L only
Note 17. RZ/A1LU only

chanhvinh.com

48.2 Treatment of Unused Pins

How unused pins are to be handled is indicated below.

Table 48.2 Handling of Unused Pins (Except for Debugger Interface Pins)

Pin Name	Handling
NMI	Fix this pin at a high level (pull up or connect to a power supply).
DP1, DP0, DM1, DM0, VBUS1, VBUS0	- Connect these pins to USBDPVss (QFP package). - Connect them to Vss (BGA package).
REFRIN	Connect this pin, via a 5.6 kΩ ± 20 % resistor, to USBAPVcc.
1.18-V power dedicated to the USB (USBAVcc, USBDVcc, USBUVcc)	Supply power at 1.18 V. Note: Products in BGA packages do not have pins USBDVcc and USBUVcc.
3.3-V power dedicated to the USB (USBAPVcc, USBDPVcc)	Supply power at 3.3 V. Note: Products in BGA packages do not have the USBDPVcc pin.
Dedicated USB ground (USBAPVss, USBDPVss, USBAVss, USBDVss, USBUVss)	Connect to ground. Note: Products in BGA packages do not have the dedicated USB ground pins.
AVref	Connect this pin to AVcc.
Dedicated A/D power (AVcc)	Supply power at 3.3 V.
Dedicated A/D ground (AVss)	Connect to ground.
Dedicated input pins other than those listed above	Fix the level on the pins (pull them up or down, or connect them to the power supply or ground level)*2.
Input/output pins other than those listed above	Make the input-pin settings and then fix the level (pull them up or down)*2; alternatively, make the output-pin settings and leave the pins open-circuit.
Dedicated output pins	Open-circuit

Note 1. We recommend that the values of pull-up or pull-down resistors are in the range from 4.7 kΩ to 100 kΩ.
Note 2. By setting the ports in accord with section 41, Ports, setting of a fixed level can be made unnecessary for some pins. For details, see section 41.3.12, Port Input Buffer Control Register (PIBCn/JPIBC0).

Table 48.3 Handling of Debugger Interface Pins (when Emulator is not Used)

Pin	Handling
BSCANP	Fix this pin at a low level (pull down or connect to the ground level).
$\overline{\text{TRST}}$*3	Fix this pin at a low level (pull down or connect to the ground level). Or connect to another pin which operates in the same manner as the $\overline{\text{RES}}$ pin
TCK, TMS, TDI	Fix the level on the pins (pull them up or down, or connect them to the power supply or ground level).
TDO	Open-circuit

Note 1. When using the emulator, handle these pins as described in the manual for the emulator.
Note 2. We recommend that the values of pull-up or pull-down resistors are in the range from 4.7 kΩ to 100 kΩ.
Note 3. If the pin is not fixed to the low level, whether an emulator is or is not in use, the specifications in Table 43.13, Reset Signal Setting, or $\overline{\text{RES}}$ input rise time (t_{RSr}) or $\overline{\text{RES}}$ negating hold time (t_{RSNH}) in Table 47.6, Control Signal Timing, must be satisfied.

chanhvinh.com

48.3 Handling of Pins in Deep Standby Mode

How pins are to be handled in deep standby mode is indicated below.

For the states of pins in deep standby mode, refer to the corresponding items under section 48.1, Pin States. Handling of unused pins as described under section 48.2, Treatment of Unused Pins, also applies in deep standby mode.

Table 48.4 Handling of Pins in Deep Standby Mode

Pin	Handling
1.2-V power (Vcc, USBDVcc, USBUVcc, USBAVcc, PLLVcc)	Supply power at 1.2 V. Note: Products in BGA packages do not have pins USBDVcc and USBUVcc.
3.3-V power (PVcc, AVcc, USBDPVcc, USBAPVcc)	Supply power at 3.3 V. Note: Products in BGA packages do not have the USBDPVcc pin.
Ground (Vss, USBDVss, USBUVss, USBAVss, AVss USBDPVss, USBAPVss)	Connect to ground. Note: Products in BGA packages do not have the USBDVss, USBAVss, USBUVss, USBDPVss, and USBAPVss pins.
VBUS1, VBUS0	Fix the level on this pin (pull it up or down, or connect it to the power supply or ground level) or open circuit. However, note that current as indicated in table 45.2, DC Characteristics (2) [Current Consumption] will be drawn by the pin fixed to the high level.
REFRIN	Connect this pin to USBAPVss via 5.6 kΩ ± 1 % resistor (QFP package). Connect this pin to Vss via 5.6 kΩ ± 1% resistor (BGA package).
DP1, DP0, DM1, DM0	Fix the level on the pins (pull them up or down, or connect them to the power supply or ground level) or open circuit.
EXTAL, RTC_X1, AUDIO_X1, USB_X1	Connect the pins to the crystal oscillator or the clock-input signal, or to a fixed level (pull them up or down, or connect them to the power supply or ground level)
XTAL, RTC_X2, AUDIO_X2, USB_X2	Connect the pins to the crystal oscillator or open circuit.
Dedicated input pins other than those listed above	Fix the level on the pins (pull them up or down, or connect them to the power supply or ground level).
Input/output pins (other than those listed above) in the input state	Fix the level on the pins (pull them up or down).
Input/output pins (other than those listed above) in the high-impedance state	Fix the level on the pins (pull them up or down) or open circuit.
Input/output pins (other than those listed above) in the output state	Open-circuit
Dedicated output pins other than those listed above	Open-circuit

Note: We recommend that the values of pull-up or pull-down resistors are in the range from 4.7 kΩ to 100 kΩ.

chanhvinh.com

48.4 Recommended Combination of Bypass Capacitor

Mount a multilayer ceramic capacitor between a pair of the power supply pins as a bypass capacitor. These capacitors must be placed as close as the power supply pins of the LSI. The capacitance of the capacitors should be used 0.1 µF to 0.33 µF (recommended values). For details of the capacitor related to the crystal resonator, see section 6, Clock Pulse Generator.

Figure 48.1 and Figure 48.2 are examples of the externally allocated capacitor in the products in 208-pin and 176-pin QFP packages, respectively.

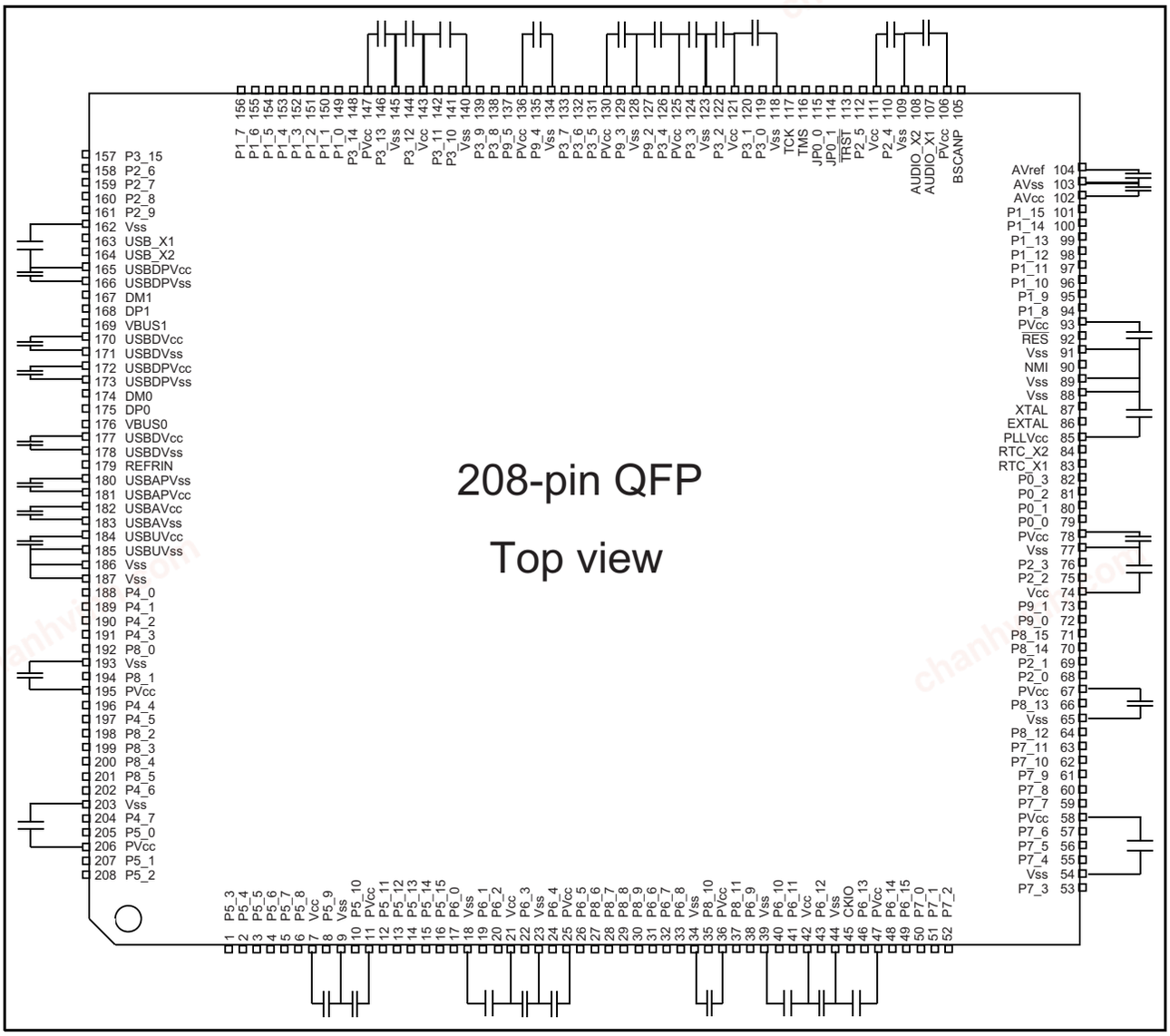


Figure 48.1 Example of Externally Allocated Capacitors in the Products in 208-Pin QFP Packages

chanhvinh.com

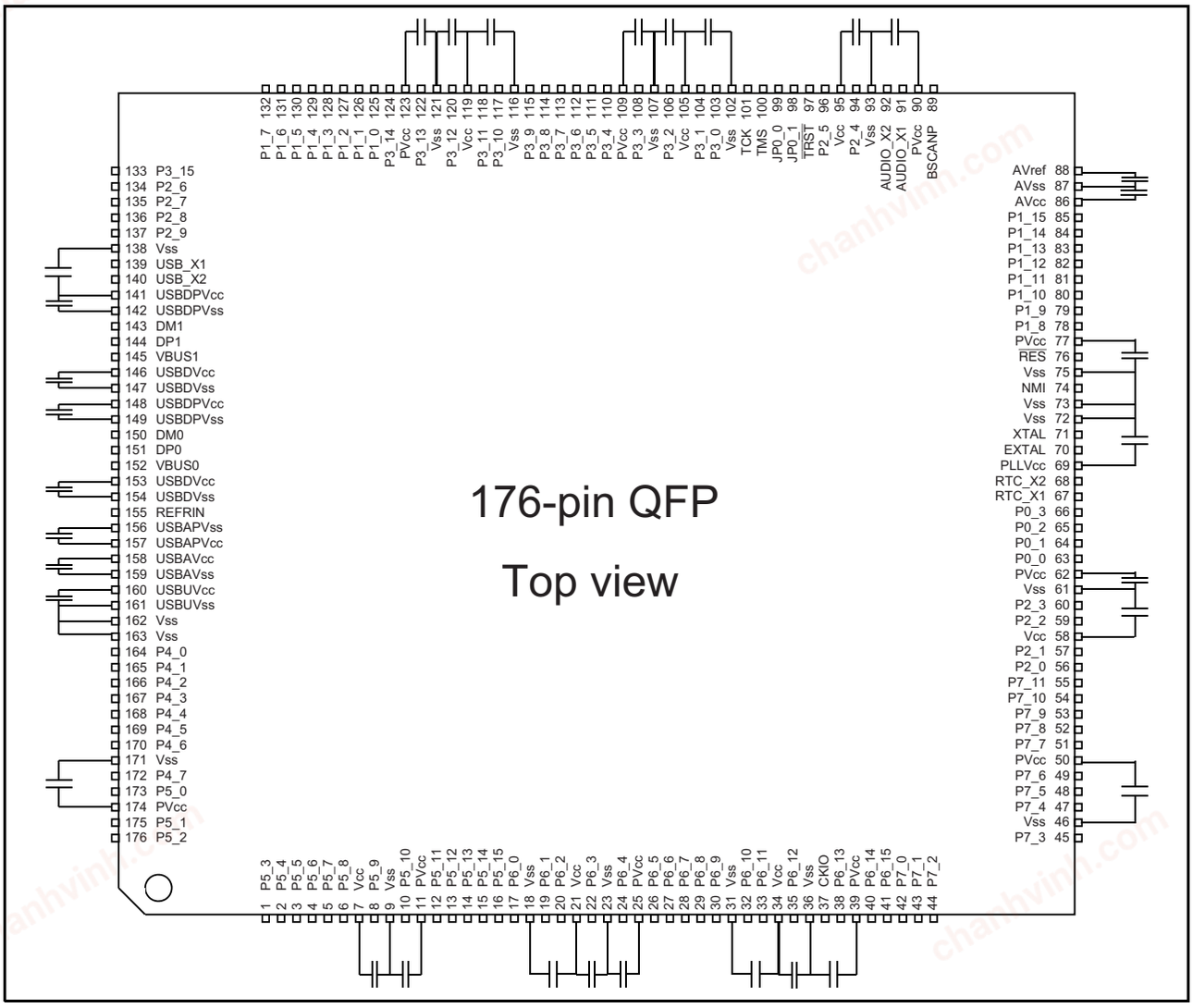


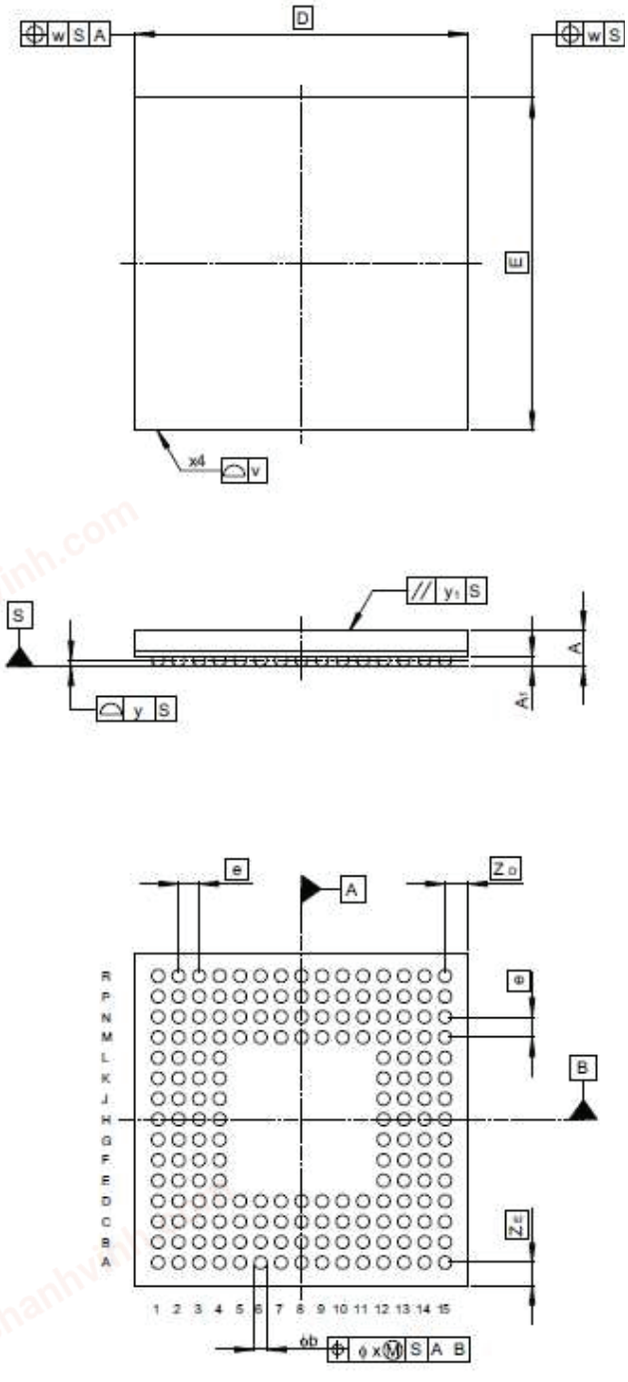
Figure 48.2 Example of Externally Allocated Capacitors in the Products in 176-Pin QFP Packages

chanhvinh.com

Appendix

A. Package Dimensions

JEITA Package code	RENESAS code	Previous code	MASS(TYP.)[g]
P-LFBGA176-8x8-0.50	PLBG0176KA-A	-	0.14

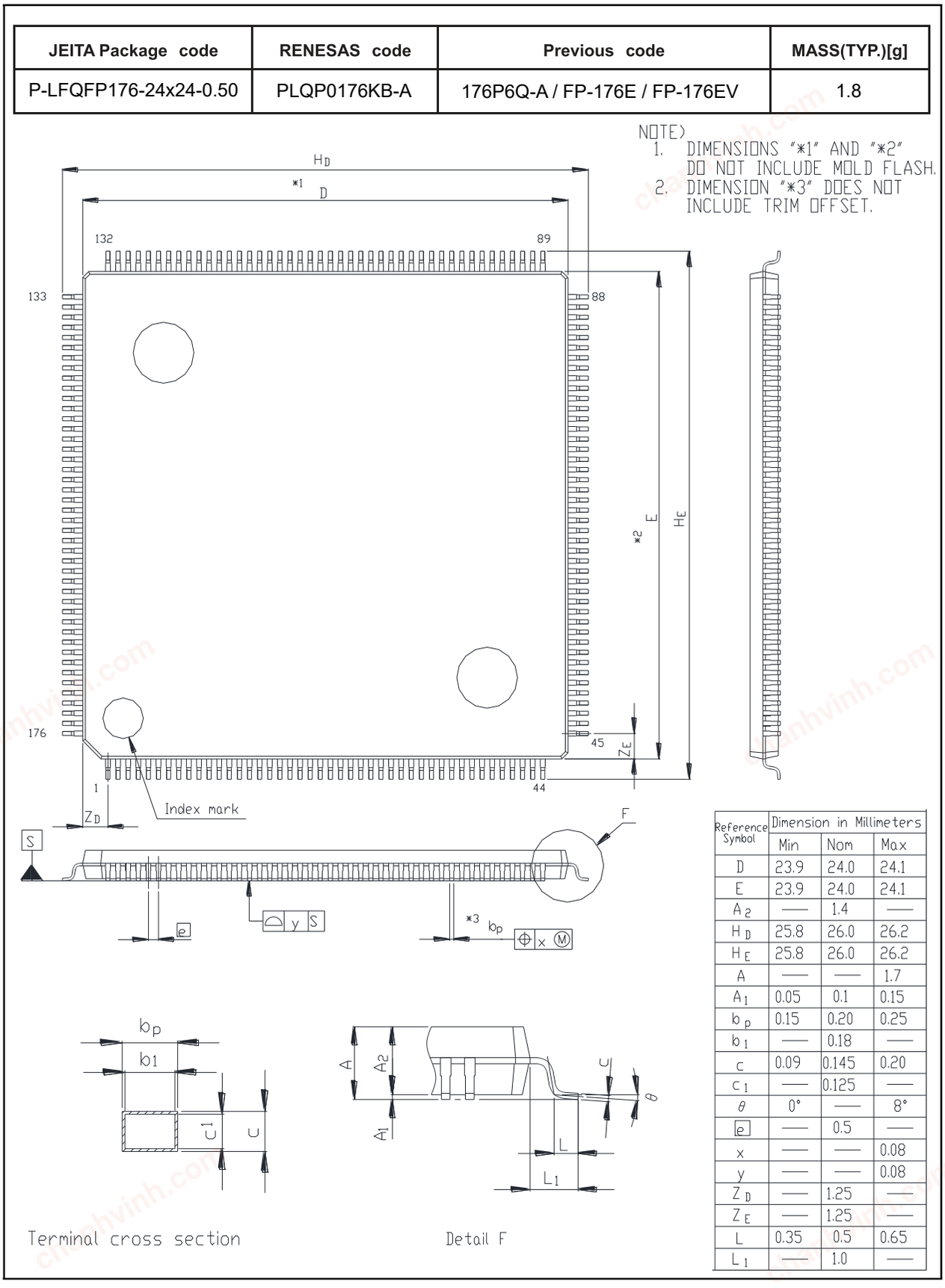


Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	8.0	—
E	—	8.0	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.40
A_1	0.20	0.25	0.30
e	—	0.50	—
b	0.25	0.30	0.35
x	—	—	0.05
y	—	—	0.08
y_1	—	—	0.20
S_D	—	—	—
S_E	—	—	—
Z_D	—	0.50	—
Z_E	—	0.50	—

Figure A.1 Dimensions of 176-Pin BGA Package

chanhvinh.com

JEITA Package code	RENESAS code	Previous code	MASS(TYP.)[g]
P-LFQFP176-24x24-0.50	PLQP0176KB-A	176P6Q-A / FP-176E / FP-176EV	1.8

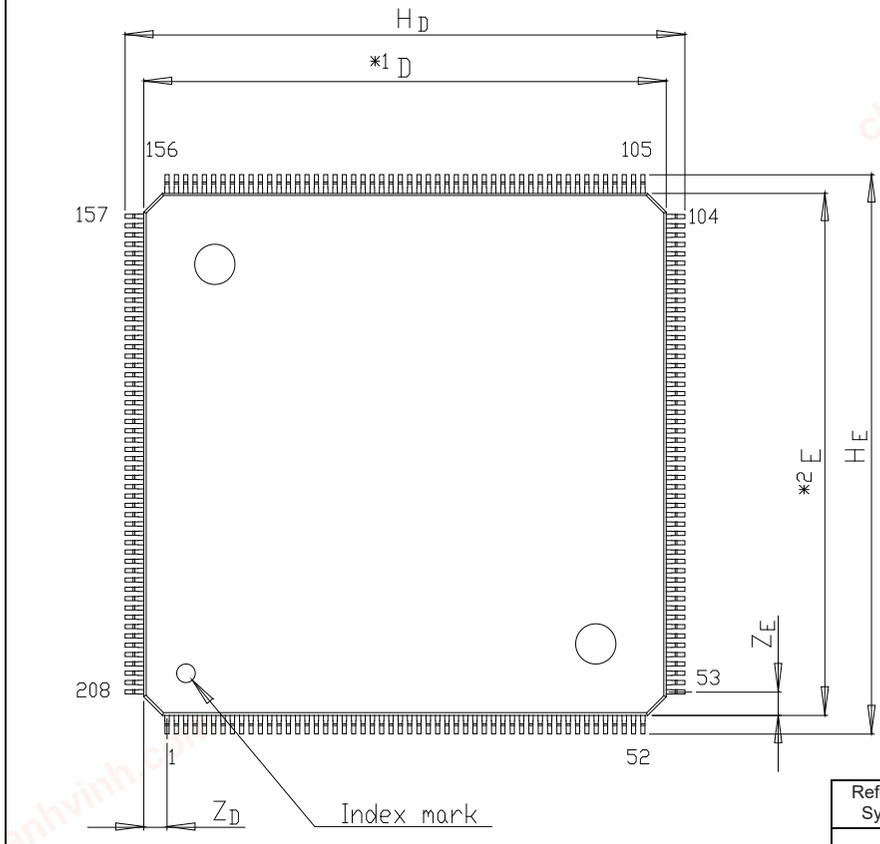


Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	23.9	24.0	24.1
E	23.9	24.0	24.1
A_2	—	1.4	—
H_D	25.8	26.0	26.2
H_E	25.8	26.0	26.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.15	0.20	0.25
b_1	—	0.18	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.35	0.5	0.65
L_1	—	1.0	—

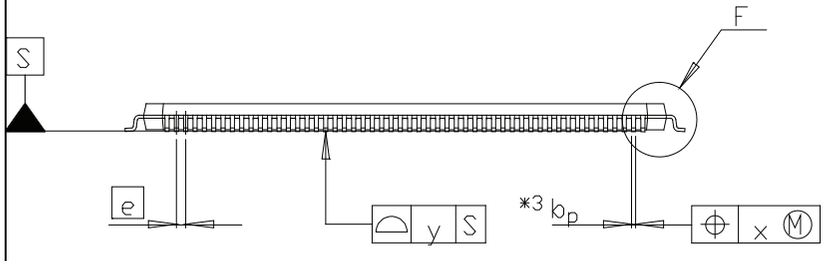
Figure A.2 Dimensions of 176-Pin QFP Package

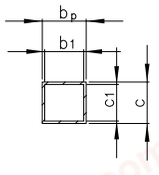
chanhvinh.com

JEITA Package code	RENESAS code	Previous code	MASS(TYP.)[g]
P-LQFP208-2828-0.50	PLQP0208KB-A	–	2.7g

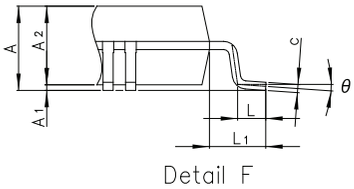


NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.





Terminal cross section



Detail F

Unit:mm

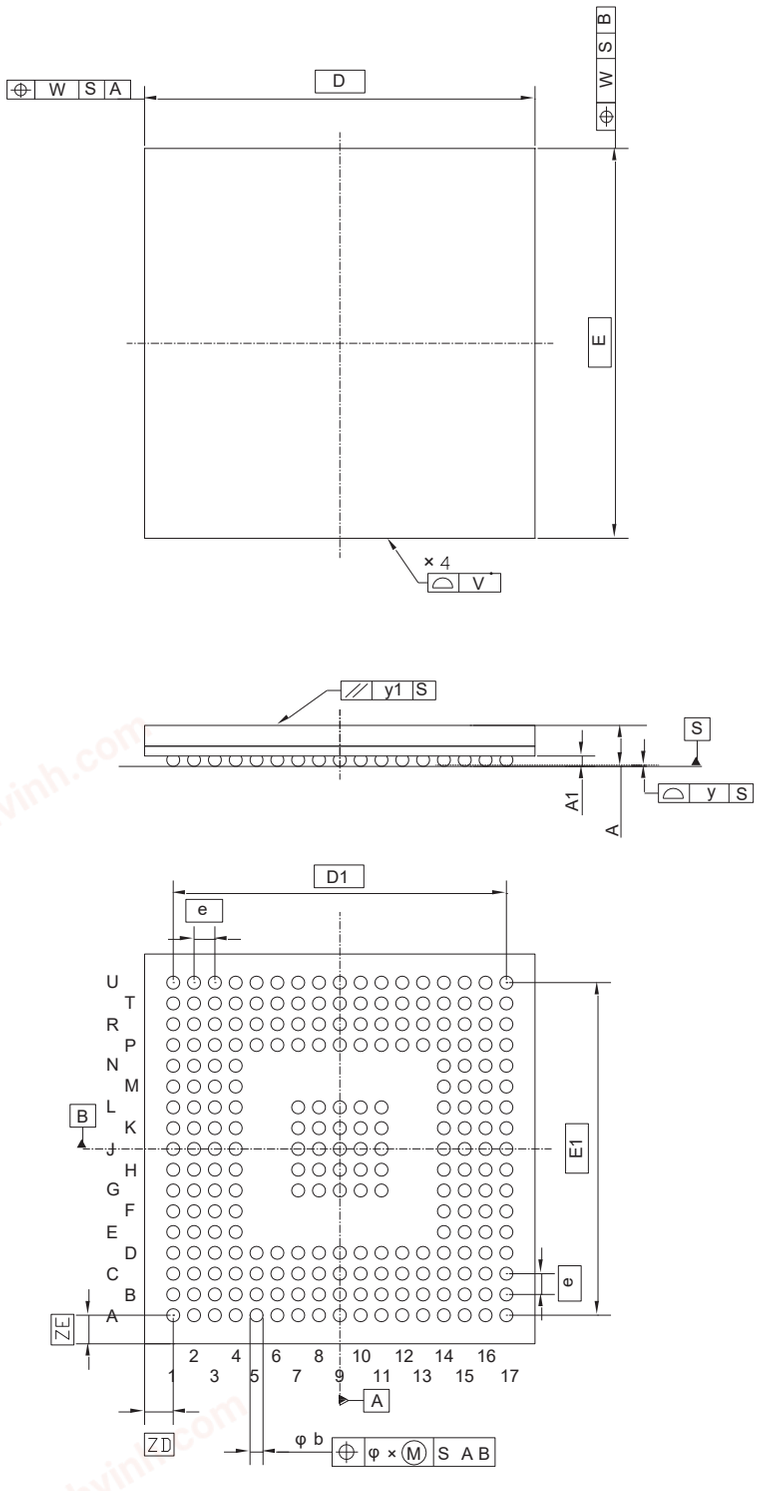
Reference Symbol	Dimension in Millimeters		
	MIN	NOM	MAX
D	–	28.0	–
E	–	28.0	–
A2	–	1.40	–
HD	29.8	30.0	30.2
HE	29.8	30.0	30.2
A	–	–	1.70
A1	0.05	0.10	0.15
b	0.17	0.22	0.27
b1	–	0.20	–
c	0.095	0.145	0.195
c1	–	0.125	–
θ	0°	–	8°
e	–	0.50	–
x	–	–	0.08
y	–	–	0.08
ZD	–	1.25	–
ZE	–	1.25	–
L	0.35	0.50	0.65
L1	–	1.00	–

Figure A.3 Dimensions of 208-Pin QFP Package

chanhvinh.com

JEITA Package code	RENESAS code	Previous code	MASS(TYP.)[g]
P-FBGA233-15x15-0.80	PRBG0233GA-A	—	0.75

Unit:mm



Reference Symbol	Dimension in Millimeters		
	Min	Mon	Max
D	—	15.00	—
D1	—	12.80	—
E	—	15.00	—
E1	—	12.80	—
v	—	—	0.15
w	—	—	0.20
e	—	0.80	—
A	—	1.58	1.90
A1	0.30	0.35	0.40
b	0.49	0.54	0.59
x	—	—	0.08
y	—	—	0.10
y1	—	—	0.20
ZD	—	1.10	—
ZE	—	1.10	—

Figure A.4 Dimensions of 233-Pin BGA Package

chanhvinh.com

Revision History

RZ/A1L Group, RZ/A1LU Group, RZ/A1LC Group User's Manual: Hardware

Rev.	Date	Description	
		Page	Summary
1.00	Aug 30, 2013	—	First edition, issued
2.00	May 25, 2015	All	Erroneous pin names, register names, and bit names corrected
		1. Overview	
		1-2 to 1-6	Table 1.1 RZ/A1L Features: Specification of CPU, IEBus™ controller, and capture engine unit corrected
		1-9	Figure 1.1 Pin Assignment of the 176-pin BGA (Upper Perspective View) corrected
		1-10	Figure 1.2 Pin Assignment of the 176-pin QFP (Top View) corrected
		1-11	Figure 1.3 Pin Assignment of the 208-pin QFP (Top View) corrected
		1-14, 1-16, 1-17	Table 1.3 Pin Functions: Note added to TIOC0A, TIOC0B, TIOC0C, and TIOC0D, function of TIOC4A, TIOC4B, TIOC4C, and TIOC4D corrected, function of REFRIN corrected, function of A/D converter corrected, and name of the symbol of general I/O port corrected
		1-18, 1-19, 1-21	Table 1.4 List of Pins: I/O of D31 to D0 corrected (I/O corrected to I(s)/O), I/O of $\overline{\text{WAIT}}$ corrected (O corrected to I(s)), symbol of Port function/dedicated function corrected (VBUSIN1 corrected to VBUS1, VBUSIN0 corrected to VBUS0)
		1-22	Figure 1.4 (2) Simplified Circuit Diagram (TTL AND Input Buffer) corrected
		2. CPU	
		2-1	2.1 Features: Note 2 added
		4. Secondary Cache	
		4-1	Table 4.1 Setting Values of Configuration Signals: Setting value of ASSOCIATIVITY corrected and note 2 corrected
		5. LSI Internal Bus	
		5-2	Figure 5.2 North Main Bus Configuration corrected (write buffer added)
		5-7	Table 5.5 Address Map: "Upper" in the area name deleted
		5-12	5.8 AXI Protocol Control Signal: Body corrected
		5-13	5.9 Write Buffers: Body corrected
		6. Clock Pulse Generator	
		6-3	(1) Crystal Oscillator: Description corrected
		6-5	Table 6.2 Input/Output Clock Frequency: Pin name corrected (MD_CLK0 → MD_CLK)
		6-7	6.4.1 Frequency Control Register (FRQCR): Description of CKOEN[1:0] bits corrected (Table 6.6 corrected to Table 6.7)
		6-15 to 6-19	6.10 Clock Signals: Newly added
		7. Interrupt Controller	
		7-1	7.1 Features: Body corrected and ARM PrimeCell® generic interrupt controller (PL390) added
		7-3	7.3 Register Descriptions: Body corrected
		7-3, 7-12	Table 7.2 Register Configuration: Superscript "*3" added to the initial value of the interrupt controller type register Table 7.2 Register Configuration: Note 1 corrected and Note 3 added
		7-15	7.3.3 IRQ Interrupt Request Register (IRQRR): Description of bits IRQ7F to IRQ0F corrected
		7-16	7.4 Interrupt Sources: Erroneous description corrected
		7-16	7.4.1 NMI Interrupt: Body corrected
		7-16	7.4.2 IRQ Interrupts: Body corrected
		7-17	7.4.3 On-Chip Peripheral Module Interrupts: Body corrected
		7-17	7.4.4 Pin Interrupts: Body corrected
		7-18	7.5 Interrupt IDs: Body corrected
		7-25	Table 7.3 List of Interrupt IDs: Entry under "ICDICFRn" for interrupt ID 240 corrected
		7-28	Table 7.3 List of Interrupt IDs: Request source names of LIN interface corrected
		7-35	7.6.1 Initial Settings: Body corrected
		7-37	7.6.2 Flow of Interrupt Operations: Body corrected
		7-39	7.8.3 Notes on Reading Interrupt ID Values from Interrupt Acknowledge Register (ICCIAR): Body corrected

chanhvinh.com

Rev.	Date	Description	
		Page	Summary
2.00	May 25, 2015	8. Bus State Controller	
		8-1	8.1 Features: Erroneous description of item 10 corrected
		8-4	Table 8.2 Address Map: Space of H'40000000 to H'407FFFFF corrected (Other → CS0 mirror)
		8-7	8.4.1 Common Control Register (CMNCR): Erroneous description of bits 28 and 24 corrected
		8-11	8.4.3 CSn Space Wait Control Register (CSnWCR) (n = 0 to 5): Description of HW[1:0] bits corrected
		8-34	8.4.10 Timeout Enable Register (TOENR): Erroneous description of CS5TOEN to CS0TOEN bits corrected
		8-77	Table 8.17 Relationship between Bus Width, Access Size, and Number of Bursts: Erroneous description in note 1 corrected
		8-78	8.5.8 SRAM Interface with Byte Selection: Body corrected
		8-86	Figure 8.40 Idle Cycle Conditions corrected
		9. Direct Memory Access Controller	
		9-2	9.3 Register Configuration: Register name corrected (into TC) in (e) DMA Register Set
		9-15	9.4.7 Channel Status Register n (CHSTAT_n): Description of the DER bit corrected
		9-22	9.4.9 Channel Configuration Register n (CHCFG_n): Description of bits 2 to 0 corrected
		9-23	9.4.11 Channel Extension Register n (CHEXT_n): DPR[2:0] and SPR[2:0] bits added, and notes corrected
		9-25	9.4.14 DMA Control Register (DCTRL_0_7, DCTRL_8_15): LWPR and LDPR bits added
		9-32	9.4.25 DMA Extension Resource Selectors 0 to 7 (DMARS0 to DMARS7): Bit 14 of DMARS2 corrected
		9-34	9.5.1 Transfer Flow: Body corrected
		9-34	9.5.2 DMA Transfer Requests: Body in (1) Auto-Request Mode corrected
		9-34	9.5.2 DMA Transfer Requests: Body in (2) External Request Mode corrected
		9-37	Table 9.4 On-Chip Peripheral Module Requests: Setting of AM[2:0] for the serial sound interface channels 0 to 3 corrected
		9-44	Figure 9.5 Register Mode Setting Example 2 corrected
		9-47	9.6.3 Link Mode: Body of <Explanation of the link mode flow>, 8. Link end analysis in (1) Operation Flow corrected
		9-57	9.7.5 DMA Acknowledge Output Function: Erroneous description corrected
		9-68, 9-69	9.8.2 Setting Example 2 (Register Mode/Software Request) changed
		9-70, 9-71	9.8.3 Setting Example 3 (Register Mode/Continuous Execution) changed
		9-73	Table 9.29 Descriptor Setting changed
		9-76	9.9.1 Divided Output of DACK0 and TEND0: Erroneous description corrected
		9-77	9.9.3 Atomic Access (ARLOCK[1:0] and AWLOCK[1:0]) added
		10. Multi-Function Timer Pulse Unit 2	
		10-1	10.1 Features: The number of interrupt sources corrected (28 corrected to 25)
		10-5	Table 10.2 Pin Configuration: Note corrected
		10-12	Table 10.10 Setting of Operation Mode by Bits MD0 to MD3: Description corrected
		10-31	10.3.4 Timer Interrupt Enable Register (TIER): Description of TTGE and TTGE2 bits corrected
		10-36	10.3.6 Timer Buffer Operation Transfer Mode Register (TBTM): Description added to description of TTSE, TTSB, and TTSA bits
		10-46	10.3.17 Timer Output Control Register 1 (TOCR1): Note 2 corrected
		10-54	10.3.23 Timer Cycle Data Register (TCDR): Body corrected
		10-55	10.3.25 Timer Interrupt Skipping Set Register (TITCR): Note corrected
		10-57	10.3.26 Timer Interrupt Skipping Counter (TITCNT): Body corrected and description of 4VCNT[2:0] bits corrected
		10-60	10.3.29 Timer Waveform Control Register (TWCR): Note corrected
		10-89	Figure 10.35 Procedure for Selecting Reset-Synchronized PWM Mode: Description of [6] and note corrected
		10-94	Figure 10.38 Example of Complementary PWM Mode Setting Procedure: Description of [9] corrected
		10-126	Figure 10.73 Example of Procedure for Specifying A/D Converter Start Request Delaying Function: Description of [2] corrected
		10-154	10.8.2 Reset Start Operation: Body corrected
		11. OS Timer	
		11-1	Table 11.2 Register Base Addresses corrected

chanhvinh.com

Rev.	Date	Description	
		Page	Summary
2.00	May 25, 2015	11-2	11.2.1 Registers Overview: Table corrected
		11-5	Table 11.8 OSTMnTS register contents: Function of the OSTMnTS bit corrected (0: This setting has no effect.)
		11-6	Table 11.9 OSTMnTT register contents: Function of the OSTMnTT bit corrected (0: This setting has no effect.)
		11-7	11.3.1 Block Diagram added, and Figure 11.1 Block Diagram of OSTM added
		11-8	11.3.3 Generation of Interrupt Request: Body corrected, and Figure 11.2 Generating an Interrupt when Counting Starts (in Interval Timer Mode) added
		11-9	11.3.4 Starting and Stopping the Timer: Description of Initializing deleted
		11-9 to 11-11	11.3.5.1 Basic Operation in Interval Timer Mode: Body corrected, Figure 11.3 Timing Diagram of OSTM in Interval Timer Mode added, and Figure 11.4 Timing Diagram of Forced Restart in Interval Timer Mode added
		11-12	11.3.5.2 Operation when OSTMnCMP = 0000 0000_H: Body corrected, and Figure 11.5 Timing Diagram when OSTMnCMP = 0000 0000_H in Interval Timer Mode added
		11-13, 11-14	11.3.6.1 Basic Operation in Free-Running Comparison Mode: Body corrected, Figure 11.6 Timing Diagram of OSTM in Free-Run Compare Mode added, and Table 11.11 OSTMTINT Generation Timing added
		11-14, 11-15	11.3.6.2 Operation when OSTMnCMP = 0000 0000_H added, and Figure 11.7 Timing Diagram when OSTMnCMP = 0000 0000_H in Free-Run Compare Mode added
		12. Watchdog Timer	
		12-1	12.1 Features: Body corrected
		12-5	12.3.3 Watchdog Reset Control/Status Register (WRCSR): Bit 5 corrected
		12-6	Figure 12.2 Writing to WTCNT and WTCSR: Addresses corrected
		12-6	12.3.4 (2) Writing to WRCSR: Body corrected
		12-6	Figure 12.3 Writing to WRCSR: Addresses corrected
		12-8	12.4.1 Canceling Software Standby Mode: Body corrected
		12-8	12.4.2 Using Watchdog Timer Mode: Body corrected
		14. Serial Communication Interface with FIFO	
		14-19	14.3.9 FIFO Control Register (SCFCR): Description of MCE bit corrected
		15. Serial Communications Interface	
		15-1	Table 15.1 Specifications of SCI: Specifications in "Clock source" under "Asynchronous mode" corrected
		15-2	Table 15.2 Input and Output Pins of the SCIs: Note corrected
		15-65	Table 15.18 Pin Configuration corrected
		16. Renesas Serial Peripheral Interface	
		16-3	16.2 Input/Output Pins: Note corrected (RSPCK)
		16-8	16.3.4 Status Register (SPSR): Note added to the description of TEND bit
		16-28	Figure 16.10 MSB First Transfer (8-Bit Data): Note corrected
		17. SPI Multi I/O Bus Controller	
		17-9	17.4.3 (1) Bit Rate: Calculation formula added
		17-9	Table 17.3 Relationship between SPBR[7:0] and BRDV[1:0] Settings: Description corrected
		18. I^2C Bus Interface	
		18-1	18.1.2 Register Base Addresses: Body corrected
		18-1	Table 18.3 Register Base Address corrected
		18-2	18.1.3 External I/O Signals: Title corrected
		18-2	Table 18.4 RIICn Pin Configuration corrected
		18-3	18.2.1 Functional Overview, Issuing and detecting conditions: Body corrected
		18-4	18.2.1 Functional Overview, Noise removal: Body corrected
		18-6	Figure 18.2 Connections to the External Circuit by the I/O Pins (I^2C Bus Configuration Example): Description corrected
		18-7 to 18-38	Bit position of each register corrected
		18-8	18.3.1 RIICnCR1 - I^2C Bus Control Register 1: Body of IICRST Bit (I^2C Bus Interface Internal Reset) corrected
		18-9	Table 18.6 RIIC Resets corrected
		18-9	18.3.1 RIICnCR1 - I^2C Bus Control Register 1: Body of ICE Bit (I^2C Bus Interface Enable) corrected
		18-11, 18-12	18.3.2 RIICnCR2 - I^2C Bus Control Register 2: Body and cautions of RS Bit (Restart Condition Issuance Request) and body of TRS Bit (Transmit/Receive Mode) corrected

chanhvinh.com

Rev.	Date	Description	
		Page	Summary
2.00	May 25, 2015	18-14	Table 18.8 RIICnMR1 register contents: Note 1 corrected
		18-17	18.3.4 RIICnMR2 - I2C Bus Mode Register 2: Note 1 of SDDL[2:0] Bits (SDA Output Delay Setup Counter) corrected (fm → Fm)
		18-19	18.3.5 RIICnMR3 - I2C Bus Mode Register 3: Body of NF[1:0] Bits (Noise Filter Stage Selection) corrected
		18-19	18.3.5 RIICnMR3 - I2C Bus Mode Register 3: Body of ACKBT Bit (Transmit Acknowledge) corrected
		18-20	18.3.5 RIICnMR3 - I2C Bus Mode Register 3: Name and body of SMBE Bit (SMBus Select) corrected
		18-29	18.3.9 RIICnSR1 - I2C Bus Status Register 1: Body of DID Flag (Device-ID Address Detection) and HOA Flag (Host Address Detection) corrected
		18-31, 18-33	18.3.10 RIICnSR2 - I2C Bus Status Register 2: Body of TMOF Flag (Timeout), AL Flag (Arbitration-Lost), and RDRF Flag (Receive Data Full) corrected
		18-35	Table 18.17 RIICnSARy register contents: Function of SVA[9:1] bits corrected
		18-36	18.3.11 RIICnSARy - I2C Slave Address Register y (y = 0 to 2): Name of SVA[9:1] Bits (7-Bit Address/10-Bit Address Upper Bits) corrected
		18-37	18.3.12 RIICnBRL - I2C Bus Bit Rate Low-Level Register: Note 1 corrected ((Sm), (Fm))
		18-38, 18-39	18.3.13 RIICnBRH - I2C Bus Bit Rate High-Level Register: Description corrected
		18-39, 18-40	Table 18.20 Minimum Specifiable Value for RIICnBRL and RIICnBRH: Title and description corrected
		18-40	Table 18.21 Examples of RIICnBRH/RIICnBRL Settings for Transfer Rate (when RIICnFER.SCLE = 1 and RIICnFER.NFE = 0) corrected
		18-40	Table 18.22 Examples of RIICnBRH/RIICnBRL Settings for Transfer Rate (when RIICnFER.SCLE = 1, RIICnFER.NFE= 1, and Number of NF Stages = 4) corrected
		18-41	18.3.14 RIICnDRT - I2C Bus Transmit Data Register: Body corrected
		18-45	18.5.1 Communication Data Format: Body corrected
		18-45	Figure 18.3 I2C Bus Format corrected
		18-45	18.5.1 Communication Data Format: Description of A# added
		18-47, 18-48	18.5.3 Master Transmit Operation: Description of (3), (5), and (7) corrected, and caution added
		18-52	18.5.4 Master Receive Operation: Body corrected
		18-54	Figure 18.10 Example Flowchart for the Master Reception of 3 or More Bytes (7-Bit Address Format): Title corrected
		18-56	Figure 18.13 Master Receive Operation Timing (3) (when RDRFS = 0): Level of TRS and TDRE corrected
		18-57	Figure 18.14 Example Flowchart for the Master Reception of 1 or 2 Bytes (7-Bit Address Format) added
		18-58	18.5.5 Slave Transmit Operation: Description of (7) corrected and caution added
		18-59	Figure 18.15 Example of Slave Transmission Flowchart corrected
		18-64	18.6 SCL Synchronization Circuit: Body corrected
		18-65	Figure 18.22 SDA Output Delay Facility corrected
		18-66	18.8 Digital Noise-Filter Circuits: Body corrected
		18-69	18.9.2 Detection of the General Call Address: Body corrected
		18-70	18.9.3 Device-ID Address Detection: Body corrected
		18-71	Figure 18.28 AASy/DID Flag Set/Clear Timing during Reception of Device-ID corrected
		18-74	Figure 18.31 Suspension of Data Transfer when NACK is Received (NACKE = 1) corrected
		18-79	18.11.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit): Body corrected
		18-80	18.11.3 Slave Arbitration-Lost Detection (SALE Bit): Body corrected
		18-82	Figure 18.37 Start Condition/Restart Condition Issue Timing (ST and RS Bits) corrected
		18-83	Figure 18.38 Stop Condition Issue Timing (SP Bit): Erroneous description corrected (IICϕ)
		18-84	18.13.1 Timeout Function: Body and description of <1> to <3> corrected
		18-87	18.13.3 RIIC Reset and Internal Reset: Reference section corrected
		18-88	18.14 SMBus Operation: Body corrected
		18-90	18.15 Reset Function of RIIC: Title corrected
		18-90, 18-91	Table 18.24 RIIC Reset Functions: Title of the table and description corrected

chanhvinh.com

Rev.	Date	Description	
		Page	Summary
2.00	May 25, 2015	19. Serial Sound Interface	
		19-39	19.5.3 Limits on TDM mode and WS Continue Mode: Body corrected
		21. CAN Interface	
		21-2	21.1.2 Register addresses: Body corrected
		21-2	Table 21.4 Register base address corrected
		21-8	Table 21.10 List of RS-CAN Module Registers (2/12): Name of transmit/receive FIFO buffer receive interrupt flag status register and transmit/receive FIFO buffer transmission interrupt flag status register corrected
		21-18	Table 21.11 Transmit Buffer p Allocated to Each Channel: CANn corrected to CANm
		21-19	Table 21.12 Transmit/Receive FIFO Buffer k Allocated to Each Channel: CANn corrected to CANm
		21-24	21.3.2 RSCAN0CmCTR - Channel Control Register (m = 0 or 1): Erroneous description of OLIE bit corrected (OLF corrected to OVLF)
		21-26	Table 21.17 RSCAN0CmSTS register contents: Function of TEC[7:0] bits corrected
		21-26	Table 21.17 RSCAN0CmSTS register contents: Function of REC[7:0] bits corrected
		21-27	21.3.3 RSCAN0CmSTS - Channel Status Register (m = 0 or 1): Body of EPSTS flag corrected
		21-52	21.3.18 RSCAN0RMNDy - Receive Buffer New Data Register y (y = 0): Body of RMNSq Flags (q = 0 to 31) corrected
		21-53	21.3.21 RSCAN0RMIDq - Receive Buffer ID Register (q = 0 to 31): Name of RMIDE bit corrected
		21-87	21.3.43 RSCAN0TMCp - Transmit Buffer Control Register (p = 0 to 31): Body corrected
		21-102	Table 21.74 RSCAN0TXQSTSm register contents: Function of bits 31 to 3 corrected
		21-122	21.5.2.1 Channel Stop Mode: Body corrected (GSLPR bit corrected to CSLPR bit)
		21-142	Figure 21.16 CAN Setting Procedure after the this LSI is Reset corrected
		21-164	21.11 Notes on the RS-CAN Module: Description corrected
		22. IEBus Controller	
		22-3	22.2.1 Function overview: Description of effective transmission speed corrected
		22-50	Table 22-29 Received control data and data written to the IEBBnDR register: Function of 5H corrected
		22-76	Note 1 corrected
		22-92	(c) Slave status transmission request processing flow example: Title corrected
		22-100	Figure 22-45 Master transmission (FIFO mode): Description corrected
		22-102	Figure 22-47 Master reception (FIFO mode): Description corrected
		22-106	Figure 22-51 Slave transmission (FIFO mode): When the control bit 3H or 7H is received: Description corrected
		22-107	Figure 22-52 Slave transmission (FIFO mode): When the control bit 0H or 6H is received (or when 4H or 5H is received from the locked master while the unit is locked): Reference section corrected
		22-111	22.7.1 (3) Effective transmission speed: Description corrected
		22-112	Table 22-56 Transfer rate and maximum number of transfer bytes in each communication mode: Effective transfer rate corrected
		22-119	Table 22-60 Control field $\overline{\text{ACK}}$ signal response conditions (when the received control data is 0H, 3H, 4H, 5H, 6H, or 7H): Title corrected (H → 0H)
		22-123	22.7.6 (9) (b) Last acknowledge bit of the control field: Body corrected
		22-125	Figure 22-62 Slave status bit configuration: Note b) corrected
		24. CD-ROM Decoder	
		24-6	Table 24.1 Register Configuration: Access size of CD-ROM decoder stream data input register corrected (R: 16, W: 16, 32)
		25. LIN Interface	
		—	LIN slave mode deleted
		25-1	25. LIN Interface: Body corrected
		25-1	25.1 Features: Title corrected
		25-1	25.1.2 Register Addresses: Body corrected
		25-1	Table 25.2 Register Base Addresses corrected
		25-2	Table 25.4 RLIN3n Interrupt and DMA Requests corrected
		25-3	25.2 Function: Body corrected
		25-4	Table 25.6 LIN Interface Specifications corrected (prohibition of the frame separated mode setting added)
		25-5	Figure 25.1 LIN Interface Block Diagram: n = 0, 1 deleted

chanhvinh.com

Rev.	Date	Description	
		Page	Summary
2.00	May 25, 2015	25-6	Table 25.7 List of LIN Interface Registers corrected (LIN master and LIN slave deleted)
		25-7 to 25-32	25.3.1 LIN Master Related Registers: Address and bit position of each register corrected
		25-7	Table 25.8 RLN3nLWBR register contents: Functions of LPRS[2:0] and LWBR0 bits corrected
		25-7	25.3.1.1 RLN3nLWBR - LIN Wake-up Baud Rate Select Register: Description of NSPB[3:0] bits (bit sampling count select bits) and LWBR0 bit (wake-up baud rate select bit) corrected
		25-8	Table 25.9 RLN3nLBRP0 register contents: Function corrected (setting range corrected)
		25-8	25.3.1.2 RLN3nLBRP0 — LIN Baud Rate Prescaler 0 Register: Body corrected
		25-8	Table 25.10 RLN3nLBRP1 register contents: Function corrected (setting range corrected)
		25-9	Table 25.11 RLN3nLSTC register contents: Function of reserved bits corrected
		25-9	25.3.1.4 RLN3nLSTC - LIN Self-Test Control Register: Body corrected
		25-10, 25-11	25.3.1.5 RLN3nLMD - LIN Mode Register: Description of LCKS[1:0] bits (LIN system clock select bits) corrected
		25-14	25.3.1.8 RLN3nLWUP - LIN Wake-up Configuration Register: Description of WUTL[3:0] bits (wake-up transmission low level width select bits) corrected
		25-15	Table 25.16 RLN3nLIE register contents: Function of SHIE, ERRIE, FRCIE, and FTCIE bits corrected
		25-15, 25-16	25.3.1.9 RLN3nLIE - LIN Interrupt Enable Register: Description of SHIE bit (successful header transmission interrupt enable bit), ERRIE bit (error detection interrupt request enable bit), FRCIE bit (successful frame/wake-up reception interrupt request enable bit), and FTCIE bit (successful frame/wake-up transmission interrupt request enable bit) corrected
		25-18	25.3.1.10 RLN3nLEDE -LIN Error Detection Enable Register: Description of PBERE bit (physical bus error detection enable bit) corrected
		25-19	25.3.1.11 RLN3nLCUC - LIN Control Register: Description of OM1 bit (LIN mode select bit) corrected
		25-20	25.3.1.12 RLN3nLTRC - LIN Transmission Control Register: Description of RTS bit (response transmission/ reception start bit) and FTS bit (frame transmission/wake-up transmission/reception start bit) corrected
		25-20	Table 25.19 RLN3nLTRC register contents: Note added (prohibition of the frame separated mode setting) in the RTS bit function
		25-22, 25-23	25.3.1.14 RLN3nLST - LIN Status Register: Body and description of HTRC flag (successful header transmission flag), D1RC flag (successful data 1 reception flag), ERR flag (error detection flag), FRC flag (successful frame/ wake-up reception flag) and FTC flag (successful frame/wake-up transmission flag) corrected
		25-24, 25-25	25.3.1.15 RLN3nLEST - LIN Error Status Register: Body and description of RPER flag (response preparation error flag), CSER flag (checksum error flag), FER flag (framing error flag), FTER flag (timeout error flag), PBER flag (physical bus error flag), and BER flag (bit error flag) corrected
		25-26 to 25-28	25.3.1.16 RLN3nLDFC - LIN Data Field Configuration Register: Description of LSS bit (transmission/reception continuation select bit), FSM bit (frame separate mode select bit), RFT bit (response field communication direction select bit), and RFDL[3:0] bits (response field length select bits) corrected
		25-26	Table 25.23 RLN3nLDFC register contents: Note added (prohibition of the frame separated mode setting) in the FSM bit function
		25-29	25.3.1.17 RLN3nLIDB — LIN ID Buffer Register: Body corrected
		25-30	25.3.1.18 RLN3nLCBR - LIN Checksum Buffer Register: Access and body corrected
		25-32	25.3.1.19 RLN3nLDBRm - LIN Data Buffer m Register (m = 1 to 8): Description of For transmission of response data of 9 bytes or more corrected
		25-34	Table 25.28 Transition Condition of Each Mode corrected
		25-38	Table 25.30 Transition Conditions of Operation Modes corrected
		25-40	Table 25.32 Processing in Response Transmission corrected
		25-41	Table 25.33 Processing in Response Reception corrected
		25-42	25.7.2.1 Data Transmission: Body corrected
		25-42	Figure 25.7 Example of Data Transmission Timing (LIN Master Mode) corrected
		25-43	Figure 25.8 Example of Data Reception Timing corrected
		25-46	25.7.3.3 Multi-Byte Response Transmission/Reception Function: Body corrected
		25-47	25.7.4.1 Wake-up Transmission: Body corrected
		25-48	25.7.4.2 Wake-up Reception: Body corrected
		25-48	25.7.4.3 Wakeup Collision added
		25-49	Table 25.34 Types of Statuses in LIN Master Mode corrected
		25-50, 25-51	Table 25.35 Types of Error Statuses in LIN Master Mode corrected
		25-50	25.7.6.1 LIN Master Mode (1) Types of Error Statuses: Body corrected

chanhvinh.com

Rev.	Date	Description	
		Page	Summary
2.00	May 25, 2015	25-53	Figure 25.14 Connection in LIN Reset Mode and LIN Mode and Figure 25.15 Connection in LIN Self-Test Mode: n = 0, 1 deleted
		25-54	25.8.1 Change to LIN Self-Test Mode: Description corrected
		25-55, 25-56	25.8.2 Transmission in LIN Master Self-Test Mode: Description corrected
		25-56, 25-57	25.8.3 Reception in LIN Master Self-Test Mode: Description corrected
		25-58	25.9 Baud Rate Generator: Body corrected
		25-58	Figure 25.16 Block Diagram of Baud Rate Generation in LIN Master Mode corrected
		25-58	25.9.1 LIN Master Mode: Title and body corrected
		—	Table 25.36 Examples of Baud Rate (19200, 10417, 9600, and 2400 bps) Generation in LIN Master Mode deleted
		25-59	25.9.2 Noise Filter: Body corrected
		25-59	Figure 25.17 Configuration of Noise Filter corrected
		25-59	Figure 25.18 Example of Noise Filter Circuit corrected
		25-60	Figure 25.19 Determination of Received Data when Noise Filter is Used corrected
		26. Ethernet Controller	
		26-15	26.3.9 CRC Error Frame Receive Counter Register (CEFCR) corrected
		26-15	26.3.10 Frame Receive Error Counter Register (FRECR) corrected
		26-16	26.3.11 Too-Short Frame Receive Counter Register (TSFRCR) corrected
		26-16	26.3.12 Too-Long Frame Receive Counter Register (TLFRCR) corrected
		26-17	26.3.13 Residual-Bit Frame Receive Counter Register (RFCR) corrected
		26-17	26.3.14 Multicast Address Frame Receive Counter Register (MAFCR) corrected
		26-55	26.4.1 (1) (b) Transmit Descriptor 1 (TD1): Description of TDL[15:0] bits corrected
		26-55	26.4.1 (1) (c) Transmit Descriptor 2 (TD2): Body corrected and description of TBA[31:0] bits corrected
		26-58	26.4.1 (2) (a) Receive Descriptor 0 (RD0): Description of RCSE bit corrected (Table 26.4 corrected to Table 26.3)
		26-59	26.4.1 (2) (b) Receive Descriptor 0 (RD0): Description of RBL[15:0] bits corrected
		26-60	Figure 26.5 Relationship between Transmit Descriptor and Transmit Buffer corrected
		27. A/D Converter	
		27-19	Table 27.5 A/D Conversion Time (Single Mode) corrected
		27-22	27.6 Definitions of A/D Conversion Accuracy: Body corrected
		27-23	27.7.1 Module Standby Mode Setting: Body corrected
		28. USB2.0 Host/Function Module	
		—	Erroneous description corrected: DxFIFO corrected to DnFIFO, DEVADDx corrected to DEVADDn, PIPExCTR corrected to PIPEnCTR, PIPExTRE corrected to PIPEnTRE, PIPExTRN corrected to PIPEnTRN, PIPExCFG corrected to PIPECFG
		28-3	Table 28.1 Pin Configuration: Function of REFRIN corrected
		28-11	28.3.1 System Configuration Control Register (SYSCFG0): Description of UCKSEL bit corrected
		28-18, 28-19	28.3.6 DMAn-FIFO Bus Configuration Registers (D0FBCFG, D1FBCFG): Description of DFACC[1:0] and TENDE bits corrected
		28-20	28.3.7 FIFO Port Registers (CFIFO, D0FIFO, D1FIFO): Description of FIFOPORT[31:0] bits corrected
		28-23	28.3.8 (1) CFIFOSEL: Description of MBW[1:0], BIGEND, ISEL, and CURPIPE[3:0] bits corrected
		28-24, 28-25	29.3.8 (2) D0FIFOSEL, D1FIFOSEL: Description of DREQE and MBW[1:0] bits corrected
		28-44, 28-45	28.3.28 DCP Configuration Register (DCPCFG): Description of bits 7 and 8 added
		28-46	Table 28.10 Relationship between Transfer Mode Settings by CNTMD Bit and Timings at which Reading Data or Transmitting Data from FIFO Buffer is Enabled added
		28-53	28.3.32 Pipe Configuration Register (PIPECFG): Description of TYPE[1:0] bits corrected
		28-56	Table 28.11 Relationship between Transfer Mode Settings by CNTMD Bit and Timings at which Reading Data or Transmitting Data from FIFO Buffer is Enabled added
		28-70	28.3.37 PIPEn Transaction Counter Enable Registers (PIPEnTRE) (n = 1 to 5, 9, and A to F): Description of TRENB bit corrected
		28-89	Table 28.19 Pipe Setting Items: Remarks of CNTMD and SHTNAK bits corrected

chanhvinh.com

Rev.	Date	Description	
		Page	Summary
2.00	May 25, 2015	28-96	Table 28.19 Relationship between Transfer Mode Settings by CNTMD Bit and Timings at which Reading Data or Transmitting Data from FIFO Buffer is Enabled deleted
		28-97	Table 28.21 FIFO Port Function Settings: Function of DREQE bit corrected
		28-98	Figure 28.6 Procedure for Pipe Switching during Access to the FIFO Port corrected
		30. Video Display Controller 5 (2): Input Controller	
		30-17	Table 30.14 Timing of Delay for Output of Vertical Synchronization Signal corrected
		30-41, 30-42	The order of 30.2.9, 30.2.10, and 30.2.11 changed
		31. Video Display Controller 5 (3): Scaler	
		31-23	Table 31.20 Settings for Field Determination Signal Control corrected
		31-26	Table 31.25 Frame Buffer Writing Mode Setting corrected
		31-27	Table 31.30 Frame Buffer Writing Control: Description corrected
		31-34	Table 31.37 Field Specification for Frame Buffer Reading corrected
		31-36	Table 31.38 Frame Buffer Control corrected
		31-51	31.2.17 Vertical Scaling Register (SC0_SCL0_DS6): Note corrected
		31-64	31.2.33 Frame Sub-Sampling Register (SC0_SCL1_WR5): Description of SC0_RES_WENB bit corrected
		31-71, 31-72	31.2.45 Pointer Buffer and Field Information Register (SC0_SCL1_PBUF_FLD): Description corrected
		31-74	31.2.49 Frame Buffer Control Register 1 (Graphics 0) (GR0_FLM1): Description of GR0_FLM_SEL [1:0] bits corrected
		31-76	31.2.51 Frame Buffer Control Register 3 (Graphics 0) (GR0_FLM3): Description of GR0_FLD_NXT bit corrected
		31-88	31.3.1 Scaling Setting Example for 525i Video Input and VGA-Size (640 x 480) Video Output: Title corrected
		31-88	Table 31.40 Input and Output Angles for 525i Video Input and VGA-Size (640 x 480) Video Output: Title corrected and output signal corrected (640 x 480)
		31-88	(2) Horizontal Scaling (Horizontal Scale Down, Scaling Filter: 2-Tap Linear): Body corrected
		31-88	(3) Horizontal Scaling (Horizontal Scale Up, Scaling Filter: 2-Tap Linear) deleted
		31-88	(3) Vertical Scaling (Vertical Scale Up, Scaling Filter: 2-Tap Linear): Body corrected
		34. Video Display Controller 5 (7): Output Controller	
		34-6, 34-7	Table 34.5 Gamma Correction: Description corrected
		34-34 to 34-41	34.2.4 Area Setting Register G1 in Gamma Correction Block (GAM_G_AREA1) to 34.2.11 Area Setting Register G8 in Gamma Correction Block (GAM_G_AREA8): Description corrected
		34-45 to 34-52	34.2.14 Area Setting Register B1 in Gamma Correction Block (GAM_B_AREA1) to 34.2.21 Area Setting Register B8 in Gamma Correction Block (GAM_B_AREA8): Description corrected
		34-56 to 34-63	34.2.24 Area Setting Register R1 in Gamma Correction Block (GAM_R_AREA1) to 34.2.31 Area Setting Register R8 in Gamma Correction Block (GAM_R_AREA8): Description corrected
		36. Capture Engine Unit	
		36-2	Table 36.1 Functional Overview of CEU corrected
		36-7	36.4.1 Capture Start Register (CAPSR): Note added
		36-13	36.4.3 Capture Interface Control Register (CAMCR): Description of JPG[1:0] bits corrected and note added to body
		36-18	36.4.6 Capture Interface Width Register (CAPWR): Description of VWDTH[11:0] and HWDTH[12:0] bits corrected
		36-36	36.4.17 Capture Bundle Destination Size Register (CBDSR): Description of CBVS[22:0] corrected
		36-36	36.4.17 (a) Image capture and data synchronous fetch: Body corrected
		36-49	Figure 36.47 Operating Status during Capturing: Erroneous description corrected (CSPSR.CE corrected to CAPSR.CE)
		36-59	Table 36.11 Restrictions on CEU Input/Output Functions corrected
		37. SCUX	
		37-2	Figure 37.1 Block Diagram of SCUX corrected (MTU2 added)
		37-3	Table 37.1 Pin Configuration: Function corrected
		37-15	37.3.9 FFD0_n FIFO Download Boot Register (FFDBR_FFD0_n) (n = 0, 1, 2, 3): Description of BOOT bit corrected
		37-16	37.3.10 FFD0_n FIFO Download Event Mask Register (DEVMR_FFD0_n) (n = 0, 1, 2, 3): Description of DEVMRQ bit corrected

chanhvinh.com

Rev.	Date	Description	
		Page	Summary
2.00	May 25, 2015	37-22	37.3.17 FFU0_n FIFO Upload Event Clear Register (UEVCR_FFU0_n) (n = 0, 1, 2, 3): Description of UEVCUF, UEVCOF, and UEVCOL bits corrected
		37-70	37.3.68 FFD0_n Timing Select Register (FDTSELn_CIM) (n = 0, 1, 2, 3): Description of MTUSEL bit added and description of SCKSEL bit corrected
		37-71	Figure 37.4 Configuration Diagram of Input Timing Signal Selector corrected
		37-72	37.3.69 FFU0_n Timing Select Register (FDTSELn_CIM) (n = 0, 1, 2, 3): Description of MTUSEL bit added and description of SCKSEL bit corrected
		37-73	Figure 37.5 Configuration Diagram of Output Timing Signal Selector corrected
		37-87	Figure 37.7 Initial Setting Procedure of SCUX (2) corrected
		37-87	Figure 37.8 Initial Setting Procedure of SCUX (3) corrected
		37-88	Figure 37.9 Transfer Start Procedure and Stop Procedure of SCUX (1) corrected
		37-90	Figure 37.11 Transfer Start Procedure and Stop Procedure of SCUX (3) corrected
		37-95	37.4.5 Data Transfer Route: Body corrected
		37-101 to 37-103	Table 37.18 Input/Output Timing Setting (1) to Table 37.20 Input/Output Timing Setting (3) corrected, and notes corrected
		37-109	Figure 37.16 Volume Ramp Operation: Note added to description of operation (2)
		37-112, 37-113	Figure 37.20 Volume Ramp Operation for 4-System Data corrected, and note added to description of operation (2)
		41. Ports	
		41-1	41.1.2 Port group index n: Body corrected
		41-2	Table 41.3 Pin Function Configuration Overview corrected, note deleted, and caution added
		41-3	41.2.2 Pin Data Input/Output: Body corrected
		41-3	Table 41.4 PPRn.PPRnm Bit Read Values corrected, and note changed to caution
		41-4	41.3.1 Port Register (Pn): Body corrected
		41-5	41.3.2 Port Set and Reset Register (PSRn): Body corrected
		41-5	41.3.3 Port Pin Read Register (PPRn/JPPR0): Body corrected and function of PPRn[15:0] bits corrected
		41-6	41.3.4 Port Mode Register (PMn): Caution added and note corrected
		41-7	Table 41.6 Alternative Mode Selection corrected and note changed to caution
		41-8	41.3.8 Port NOT Register (PNOTn): Function of PNOTn[15:0] bits corrected
		41-9	41.3.9 Port Mode Set and Reset Register (PMSRn): Body corrected
		41-10	41.3.10 Port Mode Control Set and Reset Register (PMCSRn/JPMCSR0): Body corrected
		41-10	41.3.11 Port Function Control Additional Expansion Register (PFCAEn): Caution corrected
		41-11	41.3.12 Port Input Buffer Control Register (PIBCn): Title and body corrected, and description added to note
		41-11	41.3.13 Port Bidirection Control Register (PBDCn/JPBDC0): Title and body corrected, and note deleted
		41-12	Table 41.7 Alternative Functions for Which the PIPCn.PIPCnm Bit Should be Set to 0 corrected
		41-13	41.3.15 Serial Sound Interface Noise Canceler Control Register (SNCR): Bits 4 to 6 corrected
		41-14	Table 41.8 Number and Function of Ports: Pin name corrected
		41-14	41.5 JTAG Port 0 (JP0): Note changed to caution
		41-14	Table 41.10 Control Registers (JP0) corrected and note changed to caution
		41-15	Table 41.11 Pin Function (P0) corrected and note changed to caution
		41-15	Table 41.12 Control Registers (P0): Note changed to caution
		41-16	Table 41.13 Pin Function (P1) corrected and note changed to caution
		41-16	Table 41.14 Control Registers (P1) corrected and note changed to caution
		41-17	Table 41.16 Control Registers (P2): Note changed to caution
		41-18	Table 41.17 Pin Function (P3) corrected
		41-18	Table 41.18 Control Registers (P3) corrected and note changed to caution
		41-19	Table 41.20 Control Registers (P4): Note changed to caution
		41-20	Table 41.22 Control Registers (P5) corrected and note changed to caution
		41-22	Table 41.24 Control Registers (P6) corrected and caution added
		41-23	Table 41.25 Pin Function (P7) corrected

chanhvinh.com

Rev.	Date	Description	
		Page	Summary
2.00	May 25, 2015	41-23	Table 41.26 Control Registers (P7): Note changed to caution
		41-24	Table 41.28 Control Registers (P8) corrected
		41-25	Table 41.30 Control Registers (P9): Note changed to caution
		41-27	Figure 41.2 Example Flow of Port Settings (in Port Mode) (a): Title corrected
		41-28	Figure 41.3 Example Flow of Port Settings (in Software I/O Control Alternative Mode) (b): Title corrected and note changed to caution
		41-29	Figure 41.4 Example Flow of Port Settings (in Direct I/O Control Alternative Mode) (c) corrected and title corrected
		42. Power-Down Modes	
		42-1	Figure 42.1 Transitions between Processing States corrected
		42-2	Table 42.1 States of Power-Down Modes: Transition Conditions corrected and note 6 added
		42-4	42.2.1 Standby Control Register 1 (STBCR1): Description of STBY and DEEP bits corrected
		42-5	42.2.2 Standby Control Register 2 (STBCR2): Description of HIZ bit corrected
		42-10	42.2.7 Standby Control Register 7 (STBCR7): Erroneous description of MSTP74 bit corrected
		42-12	42.2.9 Standby Control Register 9 (STBCR9): Description of MSTP93 bit corrected
		42-13	42.2.10 Standby Control Register 10 (STBCR10): Note 1 deleted
		42-23	42.2.20 Standby Acknowledge Register 1 (STBACK1): Title corrected and note 1 corrected
		42-24	42.2.21 Standby Acknowledge Register 2 (STBACK2): Note 1 corrected
		42-30	42.2.26 Deep Standby Cancel Source Flag Register (DSFR): Body corrected
		42-32	42.3.1 (1) Transition to Sleep Mode: Body corrected and not added
		42-32	42.3.1 (2) Canceling Sleep Mode: Body corrected
		42-33	42.3.2 (1) Transition to Software Standby Mode: Description corrected
		42-35	42.3.3 Software Standby Mode Application Example: Note added
		42-36	42.3.4 (1) Transition to Deep Standby Mode: Body corrected
		42-37	Figure 42.3 Flowchart of Transition to Deep Standby Mode corrected
		42-38	Figure 42.4 Flowchart of Canceling Deep Standby Mode: Description corrected
		42-40	42.3.5 Module Standby Function: Body corrected
		42-41	42.3.6 Software Reset added
		43. Debugger Interface	
		43-7 to 43-10	Table 43.10 Correspondence between Pins of this LSI and Boundary Scan Registers corrected
		43-12	43.4 ICE Registers: Body corrected
		44. List of Registers	
		44-2	Table 44.1 Register Addresses: Frequency control register 2 deleted
		44-31	Table 44.1 Register Addresses: Access size of CD-ROM decoder stream data input register (STRMDIN0) corrected (16 (R/W), 32 (W))
		44-28	Table 44.1 Register Addresses: Register name of transmit/receive FIFO buffer receive interrupt flag status register and transmit/receive FIFO buffer transmit interrupt flag status register corrected
		44-3, 44-4, 44-51, 44-52, 44-53, 44-61	Table 44.1 Register Addresses: Interrupt clear-enable registers 17 and 18, Interrupt set-pending registers 17 and 18, Port register_10, Port register_11, Port set/reset register_10, Port set/reset register_11, Port pin read register_10, Port pin read register_11, Port mode register_10, Port mode register_11, Port mode control register_10, Port mode control register_11, Port function control register_10, Port function control register_11, Port function control expansion register_10, Port function control expansion register_11, Port NOT register_10, Port NOT register_11, Port mode set/reset register_10, Port mode set/reset register_11, Port mode control set/reset register_10, Port mode control set/reset register_11, Port function control additional expansion register_10, Port function control additional expansion register_11, Port input buffer control register_10, Port input buffer control register_11, Port bi-direction control register_10, Port bi-direction control register_11, Port IP control register_10, Port IP control register_11, and BSID register deleted
		44-69, 44-79, 44-88, 44-96, 44-97, 44-120, 44-121	Table 44.2 Register Bits: Registers of bus state controller corrected, DCTRL_0_7 and DCTRL_8_15 registers corrected, RIICxMR1, RIIC0SER, RIIC1SER, RIIC2SER, and RIIC3SER registers of I2C bus interface corrected, CEFCR0, FRECR0, TSFRCR0, TLFRCR0, RFCR0, and MAFCR0 registers of Ethernet controller corrected, and DCPCFG_0 and DCPCFG_1 registers of USB2.0 host/function module corrected

chanhvinh.com

Rev.	Date	Description	
		Page	Summary
2.00	May 25, 2015	44-66, 44-120, 44-207	Table 44.2 Register Bits: FRQCR2 register of clock pulse generator, all registers of LIN interface LIN slave, and BSID register of debugger interface deleted
		44-208, 44-209	Table 44.3 Register States: Register name (FRQCR2) deleted, module name of CAN interface corrected ((RS-CAN) deleted), and note 11 added
		45. Electrical Characteristics	
		45-5	Table 45.2 DC Characteristics (3) [Except I^2C Bus Interface, and USB 2.0 Host/Function Module-Related Pins]: Note corrected
		45-8	Table 45.5 Clock Timing: Description of real time clock oscillation settling time (t_{ROSC}) corrected
		45-9	Figure 45.3 Power-On Oscillation Settling Time corrected
		45-18	Title of figure 45.13 corrected
		45-19	Title of figure 45.14 corrected
		45-43	Figure 45.41 Clock Timing corrected
		45-44	Figure 45.42 Transmission and Reception Timing (Master, CPHA = 0) corrected
		45-44	Figure 45.43 Transmission and Reception Timing (Master, CPHA = 1) corrected
		45-44	Figure 45.44 Transmission and Reception Timing (Slave, CPHA = 0) corrected
		45-45	Figure 45.45 Transmission and Reception Timing (Slave, CPHA = 1) corrected
		45-46	Table 45.14 SPI Multi I/O Bus Controller Timing corrected
		45-46	Figure 45.47 Transmission and Reception Timing (CPHAT = 0, CPHAR = 0) corrected
		45-47	Figure 45.48 Transmission and Reception Timing (CPHAT = 1, CPHAR = 1) corrected
		45-61	Table 45.25 Capture Engine Unit Module Signal Timing corrected
		45-61	Figure 45.74 Capture Engine Unit Module Signal Timing corrected
		46. States and Handling of Pins	
		46-1	Table 46.1 Pin States corrected (clock and bus state controller)
		46-8	Figure 46.1 Example of Externally Allocated Capacitors in the Products in 208-Pin QFP Packages corrected
		46-9	Figure 46.2 Example of Externally Allocated Capacitors in the Products in 176-Pin QFP Packages corrected
		Appendix	
		Appendix-1	Figure A.1 Dimensions of 176-Pin BGA Package corrected
3.00	Nov 30, 2016	All	RZ/A1LU and RZ/A1LC specification added
		1. Overview	
		1-1	1.1 Features of This LSI: Body corrected
		1-2, 1-4, 1-5, 1-6, 1-7, 1-8	Table 1.1 Features of RZ/A1L, RZ/A1LU, and RZ/A1LC: Title corrected. Items of media local bus, IEBus™ controller, and CD-ROM decoder corrected. Items and specification of LIN interface corrected. EthernetAVB (RZ/A1LU only) and JPEG codec unit (RZ/A1LU only) added. Specification of on-chip RAM and package corrected.
		1-9	Table 1.2 Product Lineup: RZ/A1LU and RZ/A1LC group added
		1-15, 1-16	Table 1.3 Pin Functions: Symbol within the classification of "Renesas serial peripheral interface" corrected. Classification of media local bus, IEBus™ controller, LIN interface, and Ethernet controller corrected. EthernetAVB (RZ/A1LU only) added.
		1-19 to 1-23	Table 1.4 List of Pins: Table corrected, function 8 added, and notes 1 to 4 added
		2. CPU	
		2-2	Table 2.1 Cortex-A9 Configuration Signal Settings: Error corrected
		3. Boot Mode	
		3-4	Figure 3.1 Control Signals Output to the Serial Flash Memory Through SPI Communication Conversion corrected
		5. LSI Internal Bus	
		5-2	Figure 5.2 North Main Bus Configuration: Notes 1 and 2 added
		5-3, 5-4	Table 5.2 List of Peripheral Buses: Description of IEBus controller, LIN interface, media local bus, and CD-ROM decoder corrected. Description of JPEG codec unit (RZ/A1LU only) added to peripheral bus 4. Description of EthernetAVB (RZ/A1LU only) added to peripheral bus 6.

chanhvinh.com

Rev.	Date	Description	
		Page	Summary
3.00	Nov 30, 2016	5-5	Figure 5.3 South Main Bus Configuration: EthernetAVB added, and notes 1 and 2 added
		5-7, 5-8, 5-9	Table 5.5 Address Map: "Area" of address 0x6020_0000 to 0x602F_FFFF and address 0x2020_0000 to 0x202F_FFFF corrected. Note 5 corrected.
		5-10	Figure 5.4 Address Remapping corrected
		5-10	5.5.2 Operation: Body corrected
		5-12	5.8.1 (3) Protection unit information (ARPROT[2:0], AWPROT[2:0]): Note added
		5-13	5.8.5 (4) Atomic access (ARLOCK[1:0], AWLOCK[1:0]): Note added
		5-14	Table 5.6 Register Configuration: Register name of AXI bus control register 5 corrected
		5-15	5.10.1 Remap Register (RMPR): Description of bit 1 (AXI128) corrected
		5-16	5.10.2 AXI Bus Control Register 0 (AXIBUSCTL0): Body corrected, description of bits 27 to 24 (JCUARCACHE[3:0]) and bits 19 to 16 (JCUAWCACHE[3:0]) added, and note added
		5-17	5.10.4 AXI Bus Control Register 5 (AXIBUSCTL5): Body corrected
		5-20	5.10.7 AXI Bus Response Error Interrupt Control Register 0 (AXIRERRCTL0): Description of bit 28 (JCURERREN) added, and note added
		5-22	5.10.9 AXI Bus Response Error Status Register 0 (AXIRERRST0): Description of bits 31 and 30 (JCURRESP[1:0]) and bits 29 and 28 (JCUBRESP[1:0]) added, and note added
		5-24	5.10.11 AXI Bus Response Error Clear Register 0 (AXIRERRCLR0): Description of bit 30 (JCURRESPCLR) and bit 28 (JCUBRESPCLR) added, and note added
		6. Clock Pulse Generator	
		6-3	6.1 (7) SSCG Circuit: Body corrected
		6-5	6.3 Clock Mode: Body corrected
		6-5	Table 6.3 Settable Frequency Ranges corrected
		6-12	6.7.1 Oscillation Stabilizing Time of the On-chip Crystal Oscillator: Body corrected
		6-12	6.7.2 Oscillation Stabilizing Time of the PLL circuit: Body corrected
		6-16	Figure 6.6 Audio and USB Clock Signals corrected
		6-16	Figure 6.7 Video Image Clock Signals corrected
		6-17	Figure 6.8 Clock Signals for Other Modules corrected
		6-18	Figure 6.9 Distribution of Internal Clock Signals (1) corrected
		6-19	Figure 6.10 Distribution of Internal Clock Signals (2) corrected
		7. Interrupt Controller	
		7-2	Figure 7.1 Block Diagram corrected
		7-17	7.4.3 On-Chip Peripheral Module Interrupts: Body corrected
		7-22, 7-28, 7-29	Table 7.3 List of Interrupt IDs: Module column of IEBus™ controller, CD-ROM decoder, media local bus, and LIN interface channel 0 corrected. JPEG codec unit (RZ/A1LU only) assigned to interrupt ID 126 and 127. EthernetAVB (RZ/A1LU only) assigned to interrupt ID 355 to 358.
		8. Bus State Controller	
		8-5	Table 8.3 Initial States by Areas in Boot Modes 0 and 1 to 3: Note 1 corrected
		8-87	8.5.11 (3) On-Chip Peripheral Module Access: Body corrected
		9. Direct Memory Access Controller	
		9-1	9.1 Features, Transfer requests: Body corrected
		9-31	9.4.25 DMA Extension Resource Selectors 0 to 7 (DMARS0 to DMARS7): Body corrected
		9-36 to 9-38	Table 9.4 On-Chip Peripheral Module Requests: “DMA Transfer Request Source” of IEBus™ controller, CD-ROM decoder, media local bus, and LIN interface channel 0 corrected
		10. Multi-Function Timer Pulse Unit 2	
		10-44, 10-68, 10-88, 10-95, 10-111, 10-147, 10-149, 10-152, 10-153	TCNT0, TCNT1, TCNT2, TCNT3, and TCNT4 have been respectively changed to TCNT_0, TCNT_1, TCNT_2, TCNT_3, and TCNT_4.
		10-104	Figure 10.44 Example of Initial Output in Complementary PWM Mode (1) corrected

chanhvinh.com

Rev.	Date	Description	
		Page	Summary
3.00	Nov 30, 2016	10-105	Figure 10.45 Example of Initial Output in Complementary PWM Mode (2) corrected
		12. Watchdog Timer	
		12-1	Error in the body corrected
		12-8	12.4.1 Canceling Software Standby Mode: Body corrected
		13. Realtime Clock	
		13-16	13.4.1 Initial Settings of Registers after Power-On and Oscillation Stabilization Time: Body corrected
		14. Serial Communication Interface with FIFO	
		14-13	14.3.7 Serial Status Register (SCFSR): Description of bit 1 (RDF) corrected
		14-19	14.3.9 FIFO Control Register (SCFCR): Description of bits 10 to 8 (RSTRG[2:0]), bits 7 and 6 (RTRG[1:0]), and bits 5 and 4 (TTRG[1:0]) corrected
		14-33, 14-38	14.4.2 (3) Transmitting and Receiving Data: Body corrected
		15. Serial Communications Interface	
		15-4	15.2.2 Receive Data Register (RDR): Body corrected
		16. Renesas Serial Peripheral Interface	
		16-10	16.3.5 Data Register (SPDR): Body corrected
		16-12	16.3.9 Data Control Register (SPDCR): Description of bits 6 and 5 (SPLW1, SPLW0) corrected, and Note added
		17. SPI Multi I/O Bus Controller	
		17-1	17.1 Features, Serial Flash Memory Interface: Body corrected
		17-2	Figure 17.1 Block Diagram corrected
		17-4	Table 17.2 Register Configuration corrected
		17-5, 17-6	17.4.1 Common Control Register (CMNCR): Description of bits 23 and 22 (MOIIO3[1:0]), bits 21 and 20 (MOIIO2[1:0]), bits 19 and 18 (MOIIO1[1:0]), bits 17 and 16 (MOIIO0[1:0]), bits 15 and 14 (IO3FV[1:0]), bits 13 and 12 (IO2FV[1:0]), bits 9 and 8 (IO0FV[1:0]), bit 6 (CPHAT), and bit 5 (CPHAR) corrected
		17-22	17.4.14 SPI Mode Read Data Register 0 (SMRDR0): Description of bits 31 to 0 (RDATA0[31:0]) corrected
		17-22	17.4.15 SPI Mode Read Data Register 1 (SMRDR1): Description of bits 31 to 0 (RDATA1[31:0]) corrected
		17-23	17.4.16 SPI Mode Write Data Register 0 (SMWDR0): Description of bits 31 to 0 (WDATA0[31:0]) corrected
		17-23	17.4.17 SPI Mode Write Data Register 1 (SMWDR1): Description of bits 31 to 0 (WDATA1[31:0]) corrected
		17-25	17.4.19 SPI AC Input Characteristics Adjustment Register (CKDLY) (RZ/A1LU only) added
		17-27	17.4.21 Data Read DDR Enable Register (DRDRENR) (RZ/A1LU only) added
		17-29	17.4.23 SPI Mode DDR Enable Register (SMDRENR) (RZ/A1LU only) added
		17-30	17.4.24 SPI AC Output Characteristics Adjustment Register (SPODLY) (RZ/A1LU only) added
		17-32	17.5.3 32-bit Serial Flash Addresses: Error in note corrected (RBE bit in DRCR = 1)
		17-34	17.5.6 External Address Space Read Mode: Body corrected
		17-38	Figure 17.12 Example of Initial Setting Flow in External Address Space Read Mode corrected
		17-40	17.5.8 SPI Operating Mode: Body corrected
		17-41	Figure 17.16 Example of Initial Setting Flow in SPI Operating Mode corrected
		17-42	Figure 17.17 Example of a Data Transfer Setting Flow in SPI Operating Mode corrected
		17-43	17.5.9 (3) Data Transmission and Reception Timing: Body corrected
		17-43	Figure 17.18 SDR Transfer Format: Title corrected
		17-44	Figure 17.19 DDR Transfer Format (RZ/A1LU only) added
		17-45	Table 17.5 Data Registers: Error corrected (DRDMCR, SMDMCR)
		17-45	17.5.10 (2) Data Enable: Body corrected
		17-46	Figure 17.20 Data and Enable corrected
		17-50	Table 17.7 Pin Status (2) corrected
		17-51	Table 17.8 Pin Status (3) corrected
		18. I^2C Bus Interface	
		18-37	18.3.12 RIICnBRL — I^2C Bus Bit Rate Low-Level Register: Body corrected
		18-38, 18-39	18.3.13 RIICnBRH — I^2C Bus Bit Rate High-Level Register: Calculation formulae of frequency and duty cycle corrected, and note 1 added
		18-46	Figure 18.5 Example of RIIC Initialization Flowchart: Note 1 corrected

chanhvinh.com

Rev.	Date	Description	
		Page	Summary
3.00	Nov 30, 2016	18-67	18.9.1 Slave-Address Match Detection: Body corrected
		18-68	Figure 18.25 AASy Flag Set Timing with 10-Bit Address Format Selected: Error of SDAn signal corrected
		18-72	18.9.4 Host Address Detection: Body corrected
		18-74	18.10.2 NACK Reception Transfer Suspension Function: Body corrected
		18-74	Figure 18.31 Suspension of Data Transfer when NACK is Received (NACKE = 1) corrected
		18-88	18.14 SMBus Operation: Body corrected
		19. Serial Sound Interface	
		19-21	19.3.12 FC Status Register (SSIFCSR): Error in the table corrected
		20. Media Local Bus	
		20-1	Description added
		21. CAN Interface	
		21-6	Figure 21.1 RS-CAN Module Block Diagram corrected
		22. IEBus Controller	
		22-1	Description added
		22-1	22.1 IEBB Features: Body corrected
		22-5	22.3.1 IEBBn register overview: Body corrected
		22-30	22.3.2 (14) IEBBnTMS - IEBBn transfer mode setting register: Body corrected
		24. CD-ROM Decoder	
		24-1	Description added
		24-46	24.6.8 Note on Software Reset added
		25. LIN Interface	
		25-1	Description added
		25-12	Table 25.13 RLN3nLBFC register contents: Error in the function column of bits 5 and 4 (BDT[1:0]) corrected
		25-13	Table 25.14 RLN3nLSC register contents: Error of the function column of bits 5 and 4 (IBS[1:0]) corrected
		25-36	25.6 LIN Reset Mode: Body corrected
		26. Ethernet Controller	
		26-2	Figure 26.1 Configuration of ETHER corrected
		26-57, 26-58	26.4.1 (2) Receive Descriptor, (a) Receive Descriptor 0 (RD0): Description of bits 29 and 28 (RFP[1:0]) and bits 25 to 16 (RFS[9:0]) corrected
		26-59	26.4.1 (2) Receive Descriptor, (b) Receive Descriptor 1 (RD1): Description of bits 31 to 16 (RBL[15:0]) corrected
		26-61	26.4.1 (3) Descriptor and Transmit/Receive Buffer, (b) Reception: Body corrected
		26-61	Figure 26.6 Relationship between Receive Descriptor and Receive Buffer corrected
		26-69	Figure 26.11 Sample Reception Flowchart (Single-Frame/Single-Descriptor): Title and figure corrected
		26-70	26.4.3 (2) Reception Error Processing, (b) Receive FIFO Overflow: Error in the body corrected
		26-74	Table 26.5 List of ETHER Interrupts: Error in “Register and Bit” corrected (EESR0.TC)
		26-84	26.6.3 Software Reset added
		27. A/D Converter	
		27-25	Figure 27.9 Example of Analog Input Circuit: Note deleted
		27-25	Table 27.8 Analog Input Pin Ratings: Note added
		30. Video Display Controller 5 (2): Input Controller	
		30-24	Table 30.19 YCbCr/RGB Signal Reception Timing corrected
		31. Video Display Controller 5 (3): Scaler	
		31-1	Figure 31.1 Functional Block Diagram of Scaler 0 corrected
		31-9	31.1.4 (2) Generating a Full-Screen Enable Signal: Body corrected
		31-27	Table 31.29 Frame Buffer Transfer Mode: Description corrected
		31-61	31.2.29 Writing Mode Register (SC0_SCL1_WR1): Description of bit 0 (SC0_RES_BST_MD) corrected
		31-74	31.2.49 Frame Buffer Control Register 1 (Graphics 0) (GR0_FLM1): Note corrected
		33. Video Display Controller 5 (5): Image Synthesizer	
		33-22	33.1.14 Alpha Blending Calculation: Body corrected

chanhvinh.com

Rev.	Date	Description	
		Page	Summary
3.00	Nov 30, 2016	33-27	Table 33.28 CLUT Table Configuration: Abbreviation corrected
		34. Video Display Controller 5 (7): Output Controller	
		34-4	34.1.6 (2) Offset calculation formulas for each area: Calculation formula corrected
		34-34	34.2.4 Area Setting Register G1 in Gamma Correction Block (GAM_G_AREA1): Error corrected
		34-35	34.2.5 Area Setting Register G2 in Gamma Correction Block (GAM_G_AREA2): Error corrected
		34-45	34.2.14 Area Setting Register B1 in Gamma Correction Block (GAM_B_AREA1): Error corrected
		34-46	34.2.15 Area Setting Register B2 in Gamma Correction Block (GAM_B_AREA2): Error corrected
		34-56	34.2.24 Area Setting Register R1 in Gamma Correction Block (GAM_R_AREA1): Error corrected
		34-57	34.2.25 Area Setting Register R2 in Gamma Correction Block (GAM_R_AREA2): Error corrected
		36. Capture Engine Unit	
		36-2	Table 36.1 Functional Overview of CEU: The description of "Binary data" within the classification of "Connectable camera" corrected
		36-5	Figure 36.2 Register Plane Switching Timing (VD Polarity is High-Active in Data Enable Fetch Mode): Title corrected
		36-7	Body corrected
		36-12 to 36-13	36.4.3 Capture Interface Control Register (CAMCR): Body corrected. Description of bit 27 (VDSEL), bit 26 (HDSEL), bit 25 (FLDSEL), and bit 24 (DSEL) added. Description of bits 5 and 4 (JPG[1:0]) and bit 1 (VDPOL) corrected. Note added.
		36-14	Body corrected
		36-15	Figure 36.14 Relationship between the VIO_VD and VIO_HD signals and the VD Interrupt when VD and HD are High-Active (in Image Capture Mode or Data Synchronous Fetch Mode): Title corrected
		36-15	Figure 36.15 Relationship between the VIO_VD and VIO_HD signals and the VD Interrupt when VD and HD are High-Active (in Image Capture Mode or Data Synchronous Fetch Mode) added
		36-47	36.4.22 Capture Event Flag Clear Register (CETCR): Description of bit 9 (VD) corrected
		36-59	36.5.1 (1) Clock Frequency: Body corrected
		36-60	36.5.4 Software Reset added
		37. SCUX	
		37-3	Table 37.1 Pin Configuration: Note 1 added to MLB_CLK
		37-70	37.3.68 FFD0_n Timing Select Register (FDTSELn_CIM) (n = 0, 1, 2, 3): Description of bits 3 to 0 (SCKSEL) corrected
		37-71	Figure 37.4 Configuration Diagram of Input Timing Signal Selector corrected
		37-72	37.3.69 FFU0_n Timing Select Register (FDTSELn_CIM) (n = 0, 1, 2, 3): Description of bits 3 to 0 (SCKSEL) corrected
		37-73	Figure 37.5 Configuration Diagram of Output Timing Signal Selector corrected
		37-114	37.5 Usage Note added
		38. SD Host Interface	
		38-1 to 38-56	Some specifications that don't require an NDA have been disclosed.
		39. MMC Host Interface	
		39-42	39.8.3 Software Reset added
		40. On-Chip RAM	
		40-1	40.1 Features: Body corrected
		40-1	Table 40.1 Address Spaces of On-Chip Large-Capacity RAM: Note added
		41. Ports	
		41-12	Table 41.7 Alternative Functions for Which the PIPCn.PIPCnm Bit Should be Set to 0: SSITxD4 deleted
		41-13	41.3.15 Serial Sound Interface Noise Canceler Control Register (SNCR): Body corrected, description of bit 6 (ETSEL) added, and note added
		41-17	Table 41.15 Pin Function (P2): Table corrected, and note added
		41-19	Table 41.19 Pin Function (P4): Table corrected, and note added
		41-20	Table 41.21 Pin Function (P5): Table corrected, and note added
		41-21	Table 41.23 Pin Function (P6): Table corrected, and notes 1 to 3 added
		41-23	Table 41.25 Pin Function (P7): Table corrected, and notes 1 and 2 added

chanhvinh.com

Rev.	Date	Description	
		Page	Summary
3.00	Nov 30, 2016	42. Power-Down Modes	
		42-3	Table 42.2 Register Configuration corrected
		42-6	42.2.3 Standby Control Register 3 (STBCR3): Note 1 added to bit 7 (MSTP37) and bit 5 (MSTP35)
		42-9	42.2.6 Standby Control Register 6 (STBCR6): Description of bit 1 (MSTP61) added, and note 2 added to bit 1 (MSTP61)
		42-11	42.2.8 Standby Control Register 8 (STBCR8): Description of bit 2 (MSTP82) added, note 2 added to bit 3 (MSTP83), and note 3 added to bit 2 (MSTP82)
		42-13	42.2.10 Standby Control Register 10 (STBCR10): Note 1 added to bit 2 (MSTP102)
		42-15	42.2.12 Standby Control Register 12 (STBCR12): Note added
		42-17	42.2.14 Software Reset Control Register 2 (SWRSTCR2) added
		42-22	42.2.19 Standby Request Register 1 (STBREQ1): Description of bit 3 (STBRQ13) and bit 2 (STBRQ12) added, and note 2 added to bit 3 (STBRQ13) and bit 2 (STBRQ12)
		42-23	42.2.20 Standby Request Register 2 (STBREQ2): Note 2 added to bit 7 (STBRQ27)
		42-24	42.2.21 Standby Acknowledge Register 1 (STBACK1): Description of bit 3 (STBAK13) and bit 2 (STBAK12) added, and note 2 added to bit 3 (STBAK13) and bit 2 (STBAK12)
		42-25	42.2.22 Standby Acknowledge Register 2 (STBACK2): Note 2 added to bit 7 (STBAK27)
		42-34	42.3.2 (1) Transition to Software Standby Mode: Body corrected
		42-36	42.3.4 (1) Transition to Deep Standby Mode: Body corrected
		42-39	42.3.4 (2) Canceling Deep Standby Mode: Body corrected
		42-41	42.3.5 (1) Transition to Module Standby Function: Body corrected
		43-42	42.3.6 Software Reset: Body corrected
		42-42	42.3.7 Adjustment of XTAL Crystal Oscillator Gain: Body corrected
		42-43	42.4.3 Usage Notes Applying when the USB_X1 Pin is not to be Used added
		43. Debugger Interface	
		43-5	Table 43.7 Pin Configuration: Function of the test data output (pin name: TDO) corrected, and note corrected
		43-6	Table 43.8 List of Registers of Boundary-Scan TAP Controller: Initial value of ID register (BSID) corrected, and notes 1 and 2 added
		43-11	43.3.4 ID Register (BSID): Initial value and description of bits 31 to 0 (DID[31:0]) corrected, and notes 1 and 2 added
		43-15	Table 43.12 Reset Signal Setting: Note 2 corrected, and note 3 added
		44. JPEG Codec Unit	
		44-1 to 44-46	Added
		45. EthernetAVB	
		45-1 to 45-198	Added
		46. List of Registers	
		46-1 to 46-66	Table 46.1 Register Addresses: Register addresses of SD host interface, JPEG codec unit, and EthernetAVB added. Notes 1 and 2 added.
		46-67 to 46-227	Table 46.2 Register Bits: Bits of SD host interface, JPEG codec unit, and EthernetAVB added. Notes 1 to 6 added.
		46-228, 46-229	Table 46.3 Register States: Register states of SD host interface, JPEG codec unit, and EthernetAVB added. Notes 12 and 13 added.
		47. Electrical Characteristics	
		47-8	Table 47.5 Clock Timing: Table corrected, and note added
		47-9	Figure 47.3 Power-On Oscillation Settling Time corrected
		47-9	Figure 47.4 Oscillation Settling Time on Return from Standby (Return by Reset) corrected
		47-10	Figure 47.5 Oscillation Settling Time on Return from Standby (Return by NMI or IRQ) corrected
		47-10	Figure 47.6 On-chip Oscillation Circuit Oscillation Settling Time: Title and figure corrected
		47-11	Table 47.6 Control Signal Timing: Table corrected, and notes 1 and 2 added
		47-11	Figure 47.7 (1) Reset Input Timing 1: Title corrected
		47-11	Figure 47.7 (3) Reset Input Timing 2 added

chanhvinh.com

Rev.	Date	Description	
		Page	Summary
3.00	Nov 30, 2016	47-12	Figure 47.7 (4) Reset Input Timing 3 added
		47-47	Table 47.14 SPI Multi I/O Bus Controller Timing: Table corrected, and note 2 added
		47-48	Figure 47.47 SDR Transfer Format Transmission and Reception Timing (CPHAT = 0, CPHAR = 0): Title and figure corrected
		47-48	Figure 47.48 SDR Transfer Format Transmission and Reception Timing (CPHAT = 1, CPHAR = 1): Title and figure corrected
		47-49	Figure 47.49 DDR Transfer Format Transmission and Reception Timing (CPHAT = 0, CPHAR = 0) (RZ/A1LU only) added
		47-49	Figure 47.50 DDR Transfer Format Transmission and Reception Timing (CPHAT = 1, CPHAR = 1) (RZ/A1LU only) added
		47-49	Figure 47.51 Timing for Switching the Buffers on and off (CPHAT = 0, CPHAR = 0) corrected
		47-49	Figure 47.52 Timing for Switching the Buffers on and off (CPHAT = 1, CPHAR = 1) corrected
		47-51	Table 47.16 Serial Sound Interface Timing: Table corrected, and note added
		47-54	47.4.13 Media Local Bus Timing: Description added
		47-56	47.4.15 Ethernet Controller and EthernetAVB Timing: Title corrected, and description added
		47-56	Table 47.19 Ethernet Controller and EthernetAVB Timing: Title and table corrected, and note added
		47-57	Figure 47.65 gPTP Timer External Clock Timing added
		47-58	Figure 47.66 Timer Capture Signal Timing added
		47-62	Table 47.24 Video Display Controller 5 Timing: Table corrected, and notes 1 and 2 added
		47-64	Table 47.25 Capture Engine Unit Module Signal Timing corrected
		47-65	Figure 47.78 (1) Capture Engine Unit Module Signal Timing when Data is Captured at the Rising Edges of VIO_CLK : Title and figure corrected
		47-65	Figure 47.78 (2) Capture Engine Unit Module Signal Timing when Data is Captured at the Falling Edges of VIO_CLK (RZ/A1LU and RZ/A1LC Only) added
		47-69	47.4.23 AC Characteristics Measurement Conditions: Body corrected
		48. States and Handling of Pins	
		48-2, 48-3, 48-4, 48-5	Table 48.1 Pin States: Table corrected, [Legend] corrected, notes 2 and 14 corrected, and notes 16 and 17 added
		48-6	Table 48.2 Handling of Unused Pins (Except for Debugger Interface Pins): Note 2 added
		48-6	Table 48.3 Handling of Debugger Interface Pins (when Emulator is not Used): Note 3 added
4.00	Mar 19. 2018	All	Trademarks of Arm changed
		1. Overview	
		1-3	Table 1.1 Features of RZ/A1L, RZ/A1LU, and RZ/A1LC: Specification of SPI multi I/O bus controller corrected
		1-4	Table 1.1 Features of RZ/A1L, RZ/A1LU, and RZ/A1LC: Specification of media local bus (RZ/A1L only) corrected
		1-8	Table 1.1 Features of RZ/A1L, RZ/A1LU, and RZ/A1LC: “233-pin BGA” added to specification of Package
		1-9	Table 1.2 Product Lineup: R7S721031VCBG and R7S721031VLBG added to the RZ/A1LU group
		1-13	Figure 1.4 Pin Assignment of the 233-pin BGA (Upper Perspective View): Newly added
		1-16	Table 1.3 Pin Functions: I/O column of the $\overline{\text{RTS2}}$ to $\overline{\text{RTS0}}$ and $\overline{\text{CTS2}}$ to $\overline{\text{CTS0}}$ pins in the serial communication interface with FIFO corrected
		1-19	Table 1.3 Pin Functions: Symbol of capture engine unit corrected
		1-20 to 1-24	Table 1.4 List of Pins: Column “233 BGA” and note on pins to which the PVcc, Vcc, and Vss functions can be allocated on 233-pin BGA products added
		2. CPU	
		2-2	Table 2.1 Cortex-A9 Configuration Signal Settings: Setting value of CLUSTERID signal corrected
		6. Clock Pulse Generator	
		6-20	6.11 Usage Note added and 6.11.1 Notes on the SSCG added
		7. Interrupt Controller	
		7-36	Figure 7.2 Flow of Initial Settings: Description corrected
		7-40	7.8.2 Notes on Selecting IRQ Interrupt Pin Functions: Body corrected
		7-41	7.8.4 Notes on Using IRQ Pins as Triggers for Release from Standby when Software Standby is in Use: Newly added

chanhvinh.com

Rev.	Date	Description	
		Page	Summary
4.00	Mar 19. 2018	8. Bus State Controller	
		8-27	8.4.4 SDRAM Control Register (SDCR): Description of bit 10 (RMODE) corrected
		16. Renesas Serial Peripheral Interface	
		16-12	Table 16.3 Relationship between SPBR and BRDV1 and BRDV0 Settings: Note 1 added
		18. I^2C Bus Interface	
		18-40	Table 18.21 Examples of RIICnBRH/RIICnBRL Settings for Transfer Rate (when RIICnFER.SCLE = 1 and RIICnFER.NFE = 0): Description corrected
			Table 18.22 Examples of RIICnBRH/RIICnBRL Settings for Transfer Rate (when RIICnFER.SCLE = 1, RIICnFER.NFE= 1, and Number of NF Stages = 4): Description corrected
		20. Media Local Bus	
		20-1	20.1 Features: Body corrected
		20-2	20.3 Register Description: Body corrected
		20-2	Table 20.2 Register Configuration: Note 1 corrected
		25. LIN Interface	
		25-31	25.3.1.19 RLN3nLDBRm — LIN Data Buffer m Register (m = 1 to 8): Description under "For response reception" corrected
		26. Ethernet Controller	
		26-4	Table 26.2 Register Configuration: CAM entry table specification enable register (common), CAM entry table POST 1 register, CAM entry table POST 2 register, CAM entry table POST 3 register, and CAM entry table POST 4 register added
		26-22	26.3.21 CAM Entry Table Specification Enable Register (Common) (TSU_FWSLC): Newly added
		26-27	26.3.25 CAM Entry Table POST 1 Register (TSU_POST1): Newly added
		26-28	26.3.26 CAM Entry Table POST 2 Register (TSU_POST2): Newly added
		26-29	26.3.27 CAM Entry Table POST 3 Register (TSU_POST3): Newly added
		26-30	26.3.28 CAM Entry Table POST 4 Register (TSU_POST4): Newly added
		26-66	26.4.1 (2) (b) Receive Descriptor 1 (RD1): Descriptions of bits 31 to 16 (RBL[15:0]) and bits 15 to 0 (RDL[15:0]) corrected
		26-93	Figure 26.23 Data Subject to Checksum Calculation and Figure 26.24 Data after Checksum Data Addition: Corrected
		29. Video Display Controller 5 (1): Overview	
		29-1	Table 29.1 Features of Video Display Controller 5: Function of operating frequency corrected
		31. Video Display Controller 5 (3): Scaler	
		31-23	Table 31.22 Control of Interrupt on Specified Image Line before Scaling-down, and Reading of Current Image line before Scaling-down: Description of the SC_RES_LINE[10:0] bits of the SC_SCL0_INT register corrected
		31-46	31.2.11 Interrupt Control Register (SC0_SCL0_INT): Description of bits 10 to 0 (SC0_RES_LINE[10:0]) corrected
		33. Video Display Controller 5 (5): Image Synthesizer	
		33-22	33.1.15 CLUT Table: Body corrected
		33-23	33.1.16 Multiplication Processing with Current Alpha at Alpha Blending in Rectangular Area: Body corrected
		35. Video Display Controller 5 (8):System Controller	
		35-1	Table 35.1 Interrupt Signals: Function of the INT_STA 9 bit of the S0_WLINE request source corrected
		35-2	Table 35.2 Interrupt Clear/Hold Settings: Description of the INT_STA9 bit of the SYSCNT_INT2 register corrected
		35-3	Table 35.3 Interrupt Output On/Off Settings: Description of the INT_OUT9_ON bit of the SYSCNT_INT5 register corrected
		35-9	35.2.2 Interrupt Control Register 2 (SYSCNT_INT2): Description of bit 4 (INT_STA9) corrected
		35-11	35.2.4 Interrupt Control Register 5 (SYSCNT_INT5): Description of bit 4 (INT_OUT9_ON) corrected
		37. SCUX	
		37-66	37.3.65 DMA Transfer Register for FFD0_n (DMATDn_CIM) (n = 0, 1, 2, 3): Body corrected
		37-67	37.3.66 DMA Transfer Register for FFU0_n (DMATUn_CIM) (n = 0, 1, 2, 3): Body corrected
		42. Power-Down Modes	
		42-21	42.2.18 CPU Status Register (CPUSTS): Description of bit 7 to 5 and bit 3 to 0 corrected
		42-24	42.2.21 Standby Acknowledge Register 1 (STBACK1): Descriptions of bits 7, 6, 4, and 1 corrected
		42-25	42.2.22 Standby Acknowledge Register 2 (STBACK2): Description of bits 4 to 0 corrected

chanhvinh.com

Rev.	Date	Description	
		Page	Summary
4.00	Mar 19. 2018	42-44	42.4.4 Notes on Using IRQ Pins as Triggers for Release from Standby when Software Standby is in Use: Newly added
		43. Debugger Interface	
		43-12	Table 43.11 Configuration of ICE Registers: Initial value of mode reset control register (ICEREGMDRSTCTL) corrected
			43.4.1 Mode Reset Control Register (ICEREGMDRSTCTL): Initial value and description of bits 15 to 13 corrected
			43.4.1 Mode Reset Control Register (ICEREGMDRSTCTL): Initial value and description of bits 11 to 9 corrected
		45. EthernetAVB	
		45-199	Figure 45.69 Data Subject to Checksum Calculation and Figure 45.70 Data after Checksum Data Addition: Corrected
		46. List of Registers	
		46-11	Table 46.1 Register Addresses: Address of CPU interface implementer identification register (ICCIIDR) corrected
		46-33	Table 46.1 Register Addresses: CAM entry table specification enable register (common) (TSU_FWSLC), CAM entry table POST 1 register (TSU_POST1), CAM entry table POST 2 register (TSU_POST2), CAM entry table POST 3 register (TSU_POST3), and CAM entry table POST 4 register (TSU_POST4) added
		46-128	Table 46.2 Register Bits: TSU_FWSLC, TSU_POST1, TSU_POST2, TSU_POST3, and TSU_POST4 registers added
		46-231	Table 46.3 Register States: SDIR*7 register of the debugger interface module and note 7 deleted
		47. Electrical Characteristics	
		47-8	Table 47.5 Clock Timing: Figures for reference of on-chip PLL circuit oscillation settling time, on-chip oscillation circuit oscillation settling time (other than above), and mode hold time corrected
			Table 47.5 Clock Timing: SSCG stabilizing time added
		47-9	Figure 47.4 SSCG Stabilizing Time: Newly added
		47-10	Figure 47.5 (1) Oscillation Stabilizing Time on Return from Standby (Return by Reset) and Figure 47.5 (2) Oscillation Stabilizing Time on Return from Standby (Return by NMI or IRQ): Figure numbers corrected
		47-11	Table 47.6 Control Signal Timing: Figures for reference of $\overline{\text{RES}}$ pulse width and $\overline{\text{TRST}}$ pulse width corrected
			Table 47.6 Control Signal Timing: Figures for reference of NMI pulse width, IRQ pulse width, and TINT pulse width corrected
		47-68	47.4.20 SD Host Interface Timing: Newly added
		Appendix	
		Appendix-1 to Appendix-3	Package Dimensions A.1, A.2, A.3: Corrected
		Appendix-4	Package Dimensions A.4: Newly added
5.00	Mar 19. 2020	1. Overview	
		1-13	Figure 1.4 Pin Assignment of the 233-pin BGA: The pin names on location C11 and C12 were corrected.
		3. Boot Mode	
		3-4	Figure 3.1 Control Signals Output to the Serial Flash Memory Through SPI Communication Conversion: Corrected
		9. Direct Memory Access Controller	
		9-55	Table 9.18 Setting for Detection of External Pin Request: LVL, HIEN, and LOEN in the header were corrected.
		9-57	Table 9.19 DACK0 Output Timing Setting: AM[2], AM[1:0], and REQD in the header were corrected.
		11. OS Timer	
		11-10	Figure 11.3 Timing Diagram of OSTM in Interval Timer Mode: Corrected
		11-13	Figure 11.6 Timing Diagram of OSTM in Free-Run Compare Mode: Corrected
		16. Renesas Serial Peripheral Interface	
		16-2	Figure 16.1 Block Diagram (for One Channel): A typo in the register name for SPCKD was corrected.
		18. I^2C Bus Interface	
		18-3	18.2.1 Functional Overview: The description for the SCL clock was corrected.
		18-22	18.3.6 RIICnFER — I^2C Bus Function Enable Register: The description for the NACKE bit was corrected.
		18-23, 18-24	18.3.7 RIICnSER — I^2C Bus Status Enable Register: SAR0E, SAR1E, SAR2E, and SARyE were corrected.
		18-28	18.3.9 RIICnSR1 — I^2C Bus Status Register 1: Corrections for SARyE
		18-35, 18-36	18.3.11 RIICnSARy — I^2C Slave Address Register y: Corrections for SARyE

chanhvinh.com

Rev.	Date	Description	
		Page	Summary
5.00	Mar 19. 2020	18-66	Figure 18.23 Block Diagram of Digital Noise Filter Circuit: Corrected
		18-67	18.9.1 Slave-Address Match Detection: Corrections for SARyE
		18-86	18.13.2 Extra SCL Clock Cycle Output Function: Body corrected
		18-86	Figure 18.40 Extra SCL Clock Cycle Output Function (CLO Bit): Corrected
		25. LIN Interface	
		25-52	25.8 LIN Self-Test Mode: Body corrected
		25-55	25.8.2 Transmission in LIN Master Self-Test Mode Settings for the baud rate, noise filter, and interrupt output related registers: The setting of the RLN3nLWBR register was corrected and *1 was deleted.
		25-55	25.8.2 Transmission in LIN Master Self-Test Mode Settings for the interrupt enable and error enable related registers: *2 was added to the setting of the RLN3nLIE register.
		25-55	25.8.2 Transmission in LIN Master Self-Test Mode: Note 1 was corrected.
		25-56	25.8.3 Reception in LIN Master Self-Test Mode Settings for the baud rate, noise filter, and interrupt output related registers: The setting of the RLN3nLWBR register was corrected and *1 was deleted.
		25-56	25.8.3 Reception in LIN Master Self-Test Mode Settings for the interrupt enable and error enable related registers: *2 was added to the setting of the RLN3nLIE register.
		25-56	25.8.3 Reception in LIN Master Self-Test Mode: Note 1 was corrected.
		29. Video Display Controller 5 (1): Overview	
		29-1	Table 29.1 Features of Video Display Controller 5 Input video image specification: The specification of the 8-bit input conforming to ITU-R BT.656 extended standard was corrected.
		29-3	Figure 29.1 Video Display Controller 5 Former Stage Block Diagram: Note (*) was added.
		29-3	Figure 29.2 Video Display Controller 5 Middle Stage Block Diagram: Note (*) was added.
		30. Video Display Controller 5 (2): Input Controller	
		30-33	Table 30.27 Register Configuration of Input Controller: The table title was corrected.
		30-33	Table 30.28 Register Configuration of Image Quality Adjustment Block: The table title was corrected.
		31. Video Display Controller 5 (3): Scaler	
		31-19	Table 31.16 Horizontal Scale Up Control: The description for the SC_SCL0_US4 register was corrected.
		31-20	Table 31.17 Vertical Scale Up Control: The description for the SC_SCL0_US4 register was corrected.
		31-56	31.2.22 Scaling-Up Control Block Input Size Register (SC0_SCL0_US4) Bit function table: The descriptions for the SC0_RES_IN_VW[10:0] and SC0_RES_IN_HW[10:0] bits were corrected.
		32. Video Display Controller 5 (4): Image Quality Improver	
		32-11, 32-12	Table 32.10 Image Quality Improver Register Configuration: The table title was corrected.
		34. Video Display Controller 5 (7): Output Controller	
		34-26 to 34-29	Table 34.23 Gamma Correction Block Register Configuration: The table title was corrected.
		34-29	Table 34.24 TCON Block Register Configuration: The table title was corrected.
		34-30	Table 34.25 Output Controller Register Configuration: The table title was corrected.
		37. SCUX	
		37-95	Table 37.12 Data Transfer Routes (1) Route SSIF→SRC→FFU: The module names from No. 5 through 8 were corrected.
		38. SD Host Interface	
		38-1	The description regarding non-disclosure agreement was deleted.
		(Rev.4.00)	38.1.4 SD Card Hardware Interface: Deleted
		38-7	38.2.3 Data STOP Register (SD_STOP) Bit function table: The description for the STP bit was corrected.
		38-20	38.2.16 SDIO Mode Control Register (SDIO_MODE) Bit function table: The description for the IOABT bit was corrected.
		38-27	Figure 38.5 The figure title was corrected to Reading from SD_BUF0.
		(Rev.4.00)	38.3.1 SD I/F, (2) Command Issue Timing within the Command Sequence: Deleted

chanhvinh.com

Rev.	Date	Description	
		Page	Summary
5.00	Mar 19. 2020	(Rev.4.00)	38.3.1 SD I/F, (3) SDIO Interrupt: Deleted
		(Rev.4.00)	38.4.1 Card Detect: Deleted
		38-34	38.4.1 Command without Data Transfer, (2) Operation Example: Deleted
		38-36	38.4.2 Single Block Read, (2) Operation Example: Deleted
		38-38	38.4.3 Single Block Write, (2) Operation Example: Deleted
		38-40	38.4.4 Multiple Block Read, (2) Operation Example: Deleted
		38-42	38.4.5 Multiple Block Write, (2) Operation Example: Deleted
		38-44	38.4.6 IO_RW_DIRECT Command (CMD52), (2) Operation Example: Deleted
		38-46	38.4.7 IO_RW_EXTENDED Command (CMD53/Multiple Block Read), (2) Operation Example: Deleted
		38-48	38.4.8 IO_RW_EXTENDED Command (CMD53/Multiple Block Write), (2) Operation Example: Deleted
		(Rev.4.00)	38.6 Electrical Characteristics: Deleted
		43. Debugger Interface	
		43-7	43.3.3 Boundary Scan Register (SDBSR): Body corrected
		43-7 to 47-10	Table 43.10 Correspondence between the Pins of the RZ/A1L and RZ/A1LC Group Products and the Boundary Scan Register Bits: The table title was corrected.
		43-11 to 43-14	Table 43.11 Correspondence between the Pins of the RZ/A1LU Group Products and the Boundary Scan Register Bits: Added
		46. List of Registers	
		46-103, 46-104	Table 58.2 Register Bits: I^2C bus interface RIIC0SER, RIIC1SER, RIIC2SER, and RIIC3SER registers: The names of bits 2 to 0 were corrected.
		47. Electrical Characteristics	
		47-2	Table 47.2 DC Characteristics (1) [Common Items]: The description of Three-state leakage current was corrected.
		47-3	Table 47.2 DC Characteristics (2) [Current Consumption]: The typical values of Current consumption in normal operation were corrected.
		47-5	Table 47.2 DC Characteristics (4) [I^2C Bus Interface Related Pins*]: Note was corrected.
6.00	Jan 29, 2021	3. Boot Mode	
		3-1	3. Boot Mode: Body corrected and note 1 added
		6. Clock Pulse Generator	
		6-15	Figure 6.5 Clock Signals for the System and Realtime Clock: Description of peripheral clocks 0C and 1C added
		9. Direct Memory Access Controller	
		9-36	Table 9.4 On-Chip Peripheral Module Requests: Setting of the TM bit in the CHCFG_n register for the A/D converter corrected
		9-38	Table 9.4 On-Chip Peripheral Module Requests: DMA transfer request signals of I^2C bus interface channels 0 to 3 corrected
		26. Ethernet Controller	
		26-64	26.4.1 Descriptors and Descriptor List, (a) Receive Descriptor 0 (RD0): Description of bit 27 (RFE) corrected
		27. A/D Converter	
		27-21	27.5 Interrupt Sources and DMA Transfer Request, (a) ADI Interrupt: Body corrected

chanhvinh.com

RZ/A1L Group, RZ/A1LU Group, RZ/A1LC Group
User's Manual: Hardware

Publication Date: Rev.1.00 Aug 30, 2013
Rev.6.00 Jan 29, 2021

Published by: Renesas Electronics Corporation

chanhvinh.com

RZ/A1L Group, RZ/A1LU Group, RZ/A1LC Group

Renesas Electronics Corporation

R01UH0437EJ0600